Byzantine-Resilient Federated Alternating Gradient Descent and Minimization for Partly-Decoupled Low Rank Matrix Learning

Ankit Pratap Singh [1] **Ahmed Ali Abbasi** [1] **Namrata Vaswani** [1]

Abstract

This work has two contributions. First, we introduce novel provably Byzantine-resilient sample- and communication-efficient alternating gradient descent (GD) and minimization based algorithms for solving the federated low rank matrix completion (LRMC) problem. This involves learning a low rank (LR) matrix from a small subset of its entries. Second, we extend our ideas to show how a simple modification of our algorithms also provably solves two other partly-decoupled vertically federated LR matrix learning problem, – LR column-wise sensing (LRCS), also referred to as multi-task linear representation learning, and its phaseless generalization, LR phase retrieval (LRPR). In all problems, we consider column-wise or vertical federation, i.e. each node observes a small subset of entries of a disjoint column sub-matrix of the entire LR matrix.

1. Introduction

Modern machine learning (ML) systems are vulnerable to various kinds of failures. One natural and powerful attack class is that of Byzantine attacks (Guerraoui, Rouault, et al., 2018) This means that the adversarial nodes have complete knowledge of the data at every node and of the exact algorithm (and all its parameters) implemented by every node, including center; and all the adversarial nodes can collude to use this information to design the worst possible attacks. In this work we develop, and analyze, secure (Byzantine resilient) algorithms for solving three different federated low rank matrix learning problems that share certain common features - Low rank matrix completion (LRMC), LR column-wise sensing (LRCS), and LR phase retrieval (LRPR). These find important applications in many different modern ML and medical imaging domains – recommender system design (Koren, Bell, & Volinsky, 2009), multi-task representation learning for few shot learning (Collins, Hassani, Mokhtari, & Shakkottai, 2021; Shome & Kar, 2021), federated sketching (Srinivasa, Lee, Junge, & Romberg, 2019; Anaraki & Hughes, 2014; Azizyan, Krishnamurthy, & Singh, 2014), accelerated dynamic MRI (Babu, Lingala, & Vaswani, 2023; Haldar & Liang, 2010; Lingala, Hu, DiBella, & Jacob, 2011; Yao, Xu, Huang, & Huang, 2018) and Fourier ptychography (Jagatap, Chen, Nayer, Hegde, & Vaswani, 2019). All these problems can be expressed as: learn an $n \times q$ rank r matrix $\boldsymbol{X}^*$ from measurements of the form $\boldsymbol{y}_k := \boldsymbol{A}_k \boldsymbol{x}_k^*$ or $|\boldsymbol{A}_k \boldsymbol{x}_k^*|$, $k \in [q]$ with the matrix $\boldsymbol{A}_k$ defined differently. For LRMC it is a 0-1 very sparse matrix, while for LRCS and LRPR it is a random Gaussian matrix. These problems can be federated horizontally or vertically. For LRMC, both settings are analogous (due to row-column symmetry) and both involves learning from data that is not identically distributed at the different nodes. This so-called *heterogeneous data setting* is known to be more difficult to design secure (Byzantine-resilient) algorithms for. For LRCS and LRPR, the horizontal setting is the easier homogeneous data setting, while the vertical one involves heterogeneous data. *We consider the vertical one in this work.* This is also the practically relevant one.

Related Work. Centralized LRMC problem has been extensively studied. Solutions consist of convex relaxation (Candes & Recht, 2008), which was very slow, Alternating Minimization (AltMin) with a spectral initialization (Netrapalli, Jain, & Sanghavi, 2013), and gradient descent (GD) based algorithms - Projected GD (ProjGD) (Cherapanamjeri, Gupta, & Jain, 2017; Jain & Netrapalli, 2015) and Factorized GD (FactGD) (Yi, Park, Chen, & Caramanis, 2016; Zheng & Lafferty, 2016). AltMin algorithms for LRPR and LRCS have been introduced and studied theoretically in the last six years (Nayer, Narayanamurthy, & Vaswani, 2019; Nayer & Vaswani, 2021). The works of (Nayer & Vaswani, 2023, on arXiv since Feb. 2021; Collins et al., 2021; Thekumparampil, Jain, Netrapalli, & Oh, 2021; Vaswani, 2024) introduced a much faster and novel GD-based algorithm called alternating GD and minimization (AltGDmin). This algorithm is also

[1]Department of Electrical and Computer Engineering, Iowa State University, Ames IA, USA. Correspondence to: Ankit Pratap Singh <sankit@iastate.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

communication-efficient for federated LRCS or LRPR. In recent work (Abbasi & Vaswani, 2024; Abbasi, Moothedath, & Vaswani, 2023), an AltGDmin based solution was introduced and analyzed for solving the federated LRMC problem. This was as fast as AltMin or FactGD, while being the most communication-efficient.

Byzantine-resilient federated learning algorithms, primarily (stochastic) GD and modifications, have been developed and studied extensively recently. Typical solutions involve replacing the mean/sum of the gradients from the different nodes by a different robust statistic, such as geometric median (GM) or GM of means (Chen, Su, & Xu, 2017), coordinate-wise median or trimmed mean (Yin, Chen, Kannan, & Bartlett, 2018), or Krum (Blanchard, El Mhamdi, Guerraoui, & Stainer, 2017). One of the first non-asymptotic results for Byzantine attacks is (Chen et al., 2017). This studied GD and used the geometric median (GM) of means to replace the regular mean/sum of the partial gradients from each node. Under standard assumptions – strong convexity, Lipschitz gradients, sub-exponential-ity of sample gradients, and an upper bound on the fraction of Byzantine nodes – it provided an exponentially decaying bound on the distance between the estimate at the t-th iteration and the unique global minimizer. In (Yin et al., 2018), the authors developed non-asymptotic guarantees for coordinate-wise median and the trimmed-mean estimators based GD for both convex and non-convex problems, albeit under *very strong assumptions - this work needed bounds on variance and skewness along each dimension, and also needed smoothness and convexity along each dimension*. Other work is (Alistarh, Allen-Zhu, & Li, 2018; Allen-Zhu, Ebrahimian, Li, & Alistarh, 2020). All the above works assumed homogeneous data distributions

More recent work has explored the more difficult heterogeneous data distribution setting by either assuming a bound on the amount of heterogeneity (Pillutla, Kakade, & Harchaoui, 2019; Data & Diggavi, 2021; Li, Xu, Chen, Giannakis, & Ling, 2019; Ghosh, Hong, Yin, & Ramchandran, 2019; Allouah et al., 2023), or by assuming the existing of a trusted dataset at the center (root dataset) and using detection methods to remove bad nodes (Regatti, Chen, & Gupta, 2022; Lu, Li, Chen, & Ma, 2022; Cao, Fang, Liu, & Gong, 2020; Cao & Lai, 2019; Xie, Koyejo, & Gupta, 2019). Recently (Singh & Vaswani, 2024a, 2024b) studied Byzantine-resiliency of LRCS using GM in the easier horizontally federated setting. But the ideas presented in these works cannot be extended for LRMC or vertically federated LRCS/MTRL which we explain next. (He, Ling, & Chen, 2019) presents experimental results for Byzantine-resilient LRMC, but with no theoretical guarantees. Other related works include (Alistarh et al., 2018; Cao et al., 2020; Allen-Zhu et al., 2020; Wu, Ling, Chen, & Giannakis, 2020; Defazio, Bach, & Lacoste-Julien, 2014; Acharya et al., 2022; Dadras, Stich, & Yurtsever, 2024).

Contributions and Novelty. This work has two contributions. (1) Our main contribution is novel provably secure and communication-efficient algorithms, Krum-AltGDmin and GM-AltGDmin, for solving the federated LRMC problem. Krum is an easy to compute estimator but needs order nrL^2 time to compute. GM can only be approximated, and the only algorithm for it that comes with a useful guarantee is way too complicated to implement (even the algorithm authors have not implemented it). But the guarantee is near-linear-time, order nrL times log factors. We also explain why use of coordinate-wise median is not useful in this setting: its required sample complexity is very high. This comparison is summarized in Table 1. (2) Second, we explain how both our novel algorithm and our proof approach can be extended directly to also solve two other federated LR problems – LRCS and LRPR. All three problems involve solving a partly decoupled optimization problem and all three also involve dealing with heterogeneity across nodes. We use the term "partly decoupled" to refer to optimization problems in which the unknown can be split into two subsets, and the optimization with respect to at least one subset of variables, keeping the other fixed, is decoupled. Heterogeneity means that the data at the different federated nodes is not identically distributed.

The work most closely related to ours is (Singh & Vaswani, 2024b), however this deals with a much easier setting of horizontally federated LRCS. In this case, the data, and hence node gradients, are homogeneous, making it easier to study. Also, LRCS only requires incoherence of right singular vectors of the true unknown matrix, and of each algorithm estimate. Since the columns of $\boldsymbol{B}$ (the right factor of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$), are updated locally at the nodes, the analysis of this step does not require any changes for the secure algorithm. Thirdly, it only provides guarantees for GM, which cannot be computed exactly and theory and practical algorithms for it are different. On the other hand, (1) **Incoherence of $\boldsymbol{U}$:** LRMC also requires assuming incoherence of $\boldsymbol{U}^*$, and ensuring it for each $\boldsymbol{U}$ at each algorithm iteration. This is tricky because update of $\boldsymbol{U}$ requires interaction between nodes and the GM or Krum output may not be incoherent. For this, we have to introduce a novel filtering step to eliminate the non-incoherent gradients. (2) **Heterogeneous gradients:** It is well known in the secure federated optimization literature that heterogeneity makes it more difficult to design secure algorithms: if the data at different nodes is very different, it is impossible to distinguish a Byzantine node output from an honest node output. Clearly, one can only handle a bounded amount of heterogeneity. All past work for this setting assumes a bound on the difference between gradients from different good nodes, at each algorithm iteration (Data & Diggavi, 2023, Assumption 2), (Allouah et al., 2023, Assumption

1). This is a confusing assumption on intermediate algorithm outputs and it is not clear how to satisfy it. The reason it is needed is the past works consider a large class of optimization problems. Our work only solves three LR problems and hence a bound on the difference between the column sub-matrices of $\boldsymbol{X}^*$ at different nodes suffices. (3) **Guarantees for Krum:** Our work provides guarantees for both Krum and GM. In fact, our result may be the first non-asymptotic guarantee for using the Krum estimator, which is a simple, intuitive, and easy to compute estimator, introduced in (Blanchard et al., 2017). By proving a result for Krum that is analogous to that for GM, we show how we can use it to replace GM. We obtain this result by carefully borrowing and modifying an argument embedded in the asymptotic proof given in (Blanchard et al., 2017). This may be of independent interest.

2. Secure Federated LRMC

Problem Setting. LRMC involves recovering a rank-r matrix $\boldsymbol{X}^* \in \Re^{n\times q}$, where $r \ll \min(n,q)$, from a subset of its entries. Entry j of column k, denoted $\boldsymbol{x}^*_{jk}$, is observed, independently of all other observations, with probability p. Let $\xi_{jk} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$ for $j \in [n], k \in [q]$. For each $k \in [q]$, define a diagonal 1-0 matrix $\boldsymbol{A}_k \in \Re^{n\times n}$ as

$$\boldsymbol{A}_k := \text{diag}(\xi_{jk},\ j \in [n])$$

Then, we can write $\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x}^*_k, k \in [q]$ and the data matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_q]$ is of the same size as $\boldsymbol{X}^*$ but with unobserved entries zeroed out.

For federated LRMC, we assume that there are a total of L nodes out of which at most L_{byz} can be Byzantine. Let $\mathcal{S}_\ell, \ell \in [L]$ be a partition of $[q] := \{1, 2, \ldots, q\}$ such that $|\mathcal{S}_\ell| \geq q/L > r$ for all ℓ. Node ℓ observes a subset of entries of $\boldsymbol{X}^*_\ell := [\boldsymbol{x}^*_k, k \in \mathcal{S}_\ell]$, i.e., it has access to $\boldsymbol{y}_k := \boldsymbol{A}_k \boldsymbol{x}^*_k$. We assume that each $\boldsymbol{X}^*_\ell$ has rank r and can be expressed as $\boldsymbol{X}^*_\ell = \boldsymbol{U}^* \boldsymbol{B}^*_\ell$ where $\boldsymbol{U}^*$ is an $n \times r$. The goal is to recover $\boldsymbol{U}^*$ and $\boldsymbol{B}^*_\ell$ for each $\ell \in [L]$, and thus recover $\boldsymbol{X}^* = [\boldsymbol{X}^*_1, \boldsymbol{X}^*_2, \ldots, \boldsymbol{X}^*_L]$.

Notation. We use $\|.\|_F$ to denote the Frobenius norm and $\|.\|$ without a subscript to denote the (induced) l_2 norm; $^\top$ denotes matrix or vector transpose. We use $\boldsymbol{e}_k$ to denote the k-th canonical basis vector (k-th column of identity matrix $\boldsymbol{I}$); and $\boldsymbol{M}^\dagger = (\boldsymbol{M}^\top \boldsymbol{M})^{-1} \boldsymbol{M}^\top$. For tall matrices with orthonormal columns $\boldsymbol{U}_1, \boldsymbol{U}_2$, we use $\boldsymbol{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1 \boldsymbol{U}_1{}^\top)\boldsymbol{U}_2\|_F$ as the Subspace Distance (SD) measure between the column spans of the two matrices. For any matrix $\boldsymbol{M} = [\boldsymbol{m}_1, ..., \boldsymbol{m}_q]$, we denote its sub-matrix with $\widetilde{q} = |\mathcal{S}_\ell| = \frac{q}{L}$ columns corresponding to indices in $\mathcal{S}_\ell$ by $\boldsymbol{M}_\ell$. Thus, $\boldsymbol{M} = [\boldsymbol{M}_1, \ldots, \boldsymbol{M}_\ell, \ldots, \boldsymbol{M}_L]$.

Let $\boldsymbol{X}^* \overset{\text{SVD}}{=} \boldsymbol{U}^*(\boldsymbol{\Sigma^*})\boldsymbol{V}^* := \boldsymbol{U}^* \boldsymbol{B}^*$, where $\boldsymbol{U}^* \in \Re^{n\times r}$ and has orthonormal columns and $\boldsymbol{V}^* \in \Re^{r\times q}$ with orthonormal rows. Also, we let $\boldsymbol{B}^* := \boldsymbol{\Sigma^*}\boldsymbol{V}^*$ so that $\boldsymbol{X}^* = \boldsymbol{U}^* \boldsymbol{B}^*$. We state guarantees in terms of $\tilde{\kappa} = \frac{\sigma^*_{\max}}{\sigma^*_{\min}}$ where $\sigma^*_{\min} = \min_{\ell\in[L]} \sigma_{\min}(\boldsymbol{X}^*_\ell)$, and $\sigma^*_{\max} = \max_{\ell\in[L]} \sigma_{\max}(\boldsymbol{X}^*_\ell)$. Our guarantees actually only depend the maximum and minimum singular values over the set of good nodes (and not all nodes). *We reuse the letters c, C to denote different numerical constants in each use with the convention that $c < 1$ and $C \geq 1$.*

Definition 2.1 (Krum)**.** For a set of matrices $\{\boldsymbol{Z}_1, \boldsymbol{Z}_2, ..., \boldsymbol{Z}_L\}$, their Krum and the corresponding best index Kr are defined as follows

$$Kr = \underset{\ell\in[L]}{\text{argmin}} \left(\sum_{\ell \longrightarrow \ell'} \|\boldsymbol{Z}_\ell - \boldsymbol{Z}_{\ell'}\|_F^2 \right), \text{ Krum} = \boldsymbol{Z}_{Kr}$$

where the sum $\sum_{\ell \longrightarrow \ell'}$ runs over the $(L - L_{byz} - 2)$ matrices $\boldsymbol{Z}_{\ell'}$ which are closest to $\boldsymbol{Z}_\ell$ in Frobenius norm.

Assumptions. We need three standard assumptions. The first is incoherence of the left and right singular vectors of each $\boldsymbol{X}^*_\ell$. This is needed in all works that study LRMC solutions. The second is a bound on the fraction of Byzantine nodes, which is also always needed. For notational simplicity, we assume this to be at most 40%. But any bound that satisfies $L_{byz} < \frac{L-2}{2}$ (Blanchard et al., 2017) can be assumed. Lastly, we need a bound on the amount of heterogeneity, this is also needed in all past work that deals with heterogeneous data and does not assume existence of a trustworthy root dataset.

Assumption 1. Assume that $\frac{L_{byz}}{L} < 0.4$.

Assumption 2. Assume row norm bounds on $\boldsymbol{U}^*$: $\max_{j\in[n]} \|\boldsymbol{u}^{*j}\| \leq \mu\sqrt{r/n}$, and assume that right singular vectors' incoherence holds locally for each node, i.e., $\max_{k\in\mathcal{S}_\ell} \|\boldsymbol{b}^*_k\| \leq \mu\sqrt{r/\widetilde{q}}\sigma_{\max}(\boldsymbol{X}^*_\ell)$ for a constant $\mu \geq 1$.

Assumption 3 (Bounded heterogeneity).

$$\max_{\ell,\ell'\in[L]} \|\boldsymbol{B}^*_\ell - \boldsymbol{B}^*_{\ell'}\|_F^2 \leq G_B^2 \sigma^{*2}_{\max}$$

We bound G_B in our guarantee. This assumption in turn implies that, for all $\ell, \ell' \in [L]$,

$$\|\boldsymbol{X}^*_\ell - \boldsymbol{X}^*_{\ell'}\|_F^2 = \|\boldsymbol{U}^*\boldsymbol{B}^*_\ell - \boldsymbol{U}^*\boldsymbol{B}^*_{\ell'}\|_F^2 \leq G_B^2 \sigma^{*2}_{\max}$$

2.1. Krum-AltGDmin

The complete stepwise algorithm is provided in Algorithm 1. We explain its steps below, starting with first reviewing the basic altGDmin idea. Guarantee is provided after that.

Basic AltGDmin. We first explain the basic AltGDmin idea. This imposes the LR constraint by expressing the unknown matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$ where $\boldsymbol{U}$ is an $n \times r$ matrix and $\boldsymbol{B}$ is an $r \times q$ matrix. The goal is to minimize $f(\boldsymbol{U}, \boldsymbol{B}) := \sum_{k=1}^q \|\boldsymbol{y}_k - \boldsymbol{A}_k \boldsymbol{U} \boldsymbol{b}_k\|^2$ with $\boldsymbol{U}$ being a matrix with orthonormal columns. After a careful

Methods→	**Krum** (Blanchard et al., 2017)	**GM** (Chen et al., 2017)	**CWMed** (Yin et al., 2018)
Sample Comp for Byz-AltGDmin (lower bound on $n\widetilde{q}p$)	$r^2\widetilde{q}\log\widetilde{q}\log(\frac{1}{\epsilon})$	$r^2\widetilde{q}\log\widetilde{q}\log(\frac{1}{\epsilon})$	$r\widetilde{q}\sqrt{n\log\widetilde{q}\log nr\log(\frac{1}{\epsilon})}$
Communic Cost	$nr\log(\frac{1}{\epsilon})$	$nr\log(\frac{1}{\epsilon})$	$nr\log(\frac{1}{\epsilon})$
Approximate Algorithm	No	Yes	No
Compute Cost at Center - GD	$nr^2L^2\log(\frac{1}{\epsilon})$	$nr^2L\log^3(\frac{L}{\epsilon_{approx}})\log(\frac{1}{\epsilon})$	$nr^2L\log(L)\log(\frac{1}{\epsilon})$
Compute Cost at Node - GD Ω is set of observed entries, $\mathbb{E}[\lvert\Omega\rvert] = npq$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$

Table 1: *We compare Krum, Geometric Median (GM), and Coordinate wise median (CWMed) based modification of AltGDmin. Observe that Compute cost for CWMed is smallest but its sample complexity is unreasonably high making it useless. Krum and GM have same sample complexity. GM compute cost is slightly less than Krum but it is an approximate algorithm i.e., we can compute GM with ϵ_{approx} error.*

Algorithm 1 Byz-AltGDmin-LRMC

1: ***AltGDmin Initialization:***
2: **Nodes** $\ell = 1, ..., L$
3: $\boldsymbol{U}_{00} \leftarrow$ top r left-singular vectors of $\boldsymbol{Y}_\ell$
4: Compute $\Pi_{\mathcal{U}}(\boldsymbol{U}_{00})$: if a row of $\boldsymbol{U}_{00}$ has 2-norm more than $\mu\sqrt{r/n}$, then re-normalize the row entries so that its norm equal the this value, else do nothing.
5: $\boldsymbol{U}_0 \leftarrow QR(\Pi_{\mathcal{U}}(\boldsymbol{U}_{00}))$.
6: Push $\boldsymbol{U}_{0\ell} \leftarrow QR(\Pi_{\mathcal{U}}(\boldsymbol{U}_{00}))$ to center
7: **Central Server**
8: Define set $\mathcal{I}_0 = \{\}$
9: **for** $\ell = 1$ to L **do**
10: **if** $\|\boldsymbol{u}_{0\ell}^j\| \le 1.5\mu\sqrt{\frac{r}{n}}$ for all $j \in [n]$ **then**
11: **Add** ℓ to set $\mathcal{I}_0$
12: **end for**
13: Compute $\mathcal{P}_{\boldsymbol{U}_{0\ell}} \leftarrow \boldsymbol{U}_{0\ell}\boldsymbol{U}_{0\ell}^\top$, $\ell \in \mathcal{I}_0$
14: $[Kr, \mathcal{P}_{\boldsymbol{U}_{Kr}}] = \text{Krum}\{\mathcal{P}_{\boldsymbol{U}_{0\ell}}\}_{\ell\in\mathcal{I}_0}$
15: Push $\boldsymbol{U}_0 = \boldsymbol{U}_{Kr}$ to nodes.
16: ***AltGDmin Iterations:***
17: **for** $t = 1$ to T **do**
18: **Nodes** $\ell = 1, ..., L$
19: $\boldsymbol{b}_k \leftarrow (\boldsymbol{A}_k\boldsymbol{U}_{t-1})^\dagger\boldsymbol{y}_k \; \forall\, k \in \mathcal{S}_\ell$
20: $\nabla_\ell \leftarrow 2\sum_{k\in\mathcal{S}_\ell}(\boldsymbol{A}_k\boldsymbol{U}_{t-1}\boldsymbol{b}_k - \boldsymbol{y}_k)\boldsymbol{b}_k^\top$
21: **Central Server**
22: Define set $\mathcal{I}_t = \{\}$
23: **for** $\ell = 1$ to L **do**
24: Compute $\boldsymbol{U}_{temp} \leftarrow \boldsymbol{U}_{t-1} - \eta\nabla_\ell$
25: **if** $\|\boldsymbol{u}_{temp}^j\| \le (1 - \frac{0.4}{\kappa^2})\|\boldsymbol{u}_{t-1}^j\| + 1.4\mu\sqrt{\frac{r}{n}}$ for all $j \in [n]$ **then**
26: **Add** ℓ to set $\mathcal{I}_t$
27: **end for**
28: $\nabla_{Kr} = \text{Krum}\{\nabla_\ell\}_{\ell\in\mathcal{I}_t}$
29: Compute $\boldsymbol{U}_t \leftarrow QR(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})$
30: Push $\boldsymbol{U}_t$ to nodes.
31: **end for**
32: **Output** $\boldsymbol{U}_T$.

spectral initialization for $\boldsymbol{U}$, at each iteration, it alternatively updates $\boldsymbol{B}$ and $\boldsymbol{U}$ as follows: (1) *Minimization for $\boldsymbol{B}$:* keeping $\boldsymbol{U}$ fixed, update $\boldsymbol{B}$ by solving $\min_{\boldsymbol{B}} f(\boldsymbol{U}, \boldsymbol{B})$. Clearly, this minimization decouples across columns, making it a cheap least squares problem of recovering q different r length vectors. It is solved as $\boldsymbol{b}_k \leftarrow (\boldsymbol{A}_k\boldsymbol{U})^\dagger\boldsymbol{y}_k$ for each $k \in [q]$. (2) *GD for $\boldsymbol{U}$:* keeping $\boldsymbol{B}$ fixed, update $\boldsymbol{U}$ by a GD step, followed by orthonormalizing its columns: $\boldsymbol{U}^+ \leftarrow QR(\boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}))$. Due to the decoupling in the minimization step, its time complexity is only as much as that of computing one gradient w.r.t. $\boldsymbol{U}$. In a federated setting, AltGDmin is also communication-efficient because each node needs to only send nr scalars (gradient w.r.t $\boldsymbol{U}$) at each iteration. The updating of $\boldsymbol{b}_k$s is done locally at the node where its data is available. The initialization is computed as given in line 1 and the two lines below it in Algorithm 1. *The algorithm implementation uses special features of the LRMC problem to speed up all computations, e.g., $\boldsymbol{A}_k\boldsymbol{U}$ is computed by just sub-selecting the rows corresponding to the nonzero diagonal entries of $\boldsymbol{A}_k$.* Sample-splitting is assumed to prove the guarantees.

AltGDmin-Krum. The resilient Krum based modification proceeds as follows. At each iteration, node ℓ first updates $\boldsymbol{b}_k$s for $k \in \mathcal{S}_\ell$ locally using the $\boldsymbol{y}_k$s. This step is the same as in the basic case. It can also be analyzed similarly using (Abbasi & Vaswani, 2024, Lemma 4.2, 4.3). Next, it computes the partial gradient, denoted ∇_ℓ as given in Algorithm 1, line 20. This is sent to the center.

The center does not know which received ∇_ℓ, if any, is Byzantine. To deal with this, it processes them in two steps. First, it finds the index set of ℓs for which ∇_ℓ is such that the updated $\boldsymbol{U}$ would be sufficiently incoherent (as needed by our proof). This filtered set is computed as $\mathcal{I} := \{\ell : \max_{j\in[n]} \|[\boldsymbol{U} - \eta\nabla_\ell]^j\| \le (1 - 0.4/\kappa^2)\|\boldsymbol{U}^j\| + 1.4\mu\sqrt{r/n}\}$ See line 25. Next, we compute Krum of the set of gradients ∇_ℓ, $\ell \in \mathcal{I}$. The filtering is needed because LRMC algorithms need to ensure incoherence of the updated $\boldsymbol{U}$ at each iteration. While the gradients computed

by the good (honest) nodes will satisfy this w.h.p. (can be proved), this cannot be guaranteed for Byzantine outputs.

Subspace-Krum for initialization. The initialization at the nodes is done exactly as in the basic AltGDmin algorithm (Abbasi & Vaswani, 2024). This involves computing the top r singular vectors of $\boldsymbol{Y}$; projecting the resulting matrix $\boldsymbol{U}_{00}$ onto space of row incoherent matrices, i.e., computing $\Pi_{\mathcal{U}}(\boldsymbol{U}_{00}) = \min_{\tilde{\boldsymbol{U}}\in\mathcal{U}} \|\tilde{\boldsymbol{U}} - \boldsymbol{U}_{00}\|_F$, where $\mathcal{U} := \{\tilde{\boldsymbol{U}} : \|\tilde{\boldsymbol{u}}^j\| \le \mu\sqrt{r/n}\}$; and finally computing a QR decomposition of $\Pi_{\mathcal{U}}(\boldsymbol{U}_{00})$. See line 6. This is computed as given in line 4. The projection needs time nr while the QR step needs time nr^2. The nodes send their $\boldsymbol{U}_{0\ell}$ to the center. The center processes these in two steps. The first step is again a filtering step that selects only those $\boldsymbol{U}_{0\ell}$s which are incoherent. See line 10. The second step involves computing the Subspace-Krum. This is done as follows. Observe that the received $\boldsymbol{U}_{0\ell}$s are subspace estimates. Their actual entries can be quite different. Thus we cannot use Krum on them to get a meaningful aggregate. We need a different approach that applies Krum to subspace distances. To do this, we rely on the fact that, for subspace basis matrices $\boldsymbol{U}_1, \boldsymbol{U}_2$, $\|\boldsymbol{U}_1\boldsymbol{U}_1^\top - \boldsymbol{U}_2\boldsymbol{U}_2^\top\|_F$ is another measure of subspace distance.

Guarantee for Krum-AltGDmin. We can prove the following for Algorithm 1. Under the three simple assumptions stated earlier, and if G_B is small enough, then, as long as we observe roughly order $nr^2\log\tilde{q}\log(1/\epsilon)$ matrix entries at each node, w.h.p., the subspace distance between $\boldsymbol{U}^*$ and $\boldsymbol{U}_t$, and hence error between $\boldsymbol{X}_\ell^* = \boldsymbol{U}^*\boldsymbol{B}_\ell^*$ and $\boldsymbol{X}_t = \boldsymbol{U}_t\boldsymbol{B}_\ell$ decreases exponentially with iteration t until either the desired error level ϵ is reached or the heterogeneity bound G_B is reached. Convergence up to the heterogeneity bound is also what the other SGD works show (Pillutla et al., 2019; Data & Diggavi, 2021; Li et al., 2019; Ghosh et al., 2019; Allouah et al., 2023).

Theorem 2.2. *(Krum-altGDmin-LRMC) Consider Algorithm 1 with Krum, and sample-splitting. Let* $T = C\tilde{\kappa}^2\log(1/\epsilon)$*, and step-size* $\eta \le 0.5/p\sigma^{*2}_{\max}$*. Assume that Assumption 1, 2, 3 holds and* $G_B \le \frac{c}{\tilde{\kappa}^2}$*. If*

$$n\tilde{q}p \ge C\tilde{\kappa}^{10}\mu^2\tilde{q}r^2\log\tilde{q}\log(1/\epsilon),$$

then w.p. at least $1 - 3Ln^{-10} - C\tilde{\kappa}^2\log(1/\epsilon)Ln^{-10}$

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_T) \le \max\big(\epsilon, 14C_1\tilde{\kappa}^2 G_B\big)$$

and $\|\boldsymbol{X}_T - \boldsymbol{X}^*\|_F \le \epsilon\sigma_1^{*2}$*. Compute and communication cost are stated in Table 1.*

2.2. GM-AltGDmin

It is possible to obtain a different algorithm and guarantee with using GM to replace Krum. GM is theoretically, faster to compute than Krum, although its theoretical and practical algorithms are not the same (as noted earlier). Also, its guarantee holds only with constant probability (the approximate GM algorithm works only with this much probability. The entire algorithm would be the same as Algorithm 1 except for one change. After the GM step, in both initialization and GD iterations, we need to find the gradient that is closest to the GM and use that as the output. This is needed because the GM is not one of the entries being aggregated. So then there is no way to prove that $(\boldsymbol{U} - \eta\nabla_{GM})$ will satisfy the required incoherence. The proof technique of Theorem 2.2 also extends to GM.

Algorithm 2 Byz-AltGDmin-LRMC-GM

Consider Algorithm 1 with the following changes:

Replace Line 14 by $[\ell_{best}, \mathcal{P}_{\boldsymbol{U}_{\ell_{best}}}]; \ell_{best} = \operatorname{argmin}_{\ell\in\mathcal{I}_0} \|\mathrm{GM}\{\mathcal{P}_{\boldsymbol{U}_\ell}\}_{\ell\in\mathcal{I}_0} - \mathcal{P}_{\boldsymbol{U}_\ell}\|_F$

Replace Line 28 by $\nabla_{\ell_{best}}; \ell_{best} = \operatorname{argmin}_{\ell\in\mathcal{I}_t} \|\mathrm{GM}\{\nabla_\ell\}_{\ell\in\mathcal{I}_t} - \nabla_\ell\|_F$

Corollary 2.3. *(GM-altGDmin-LRMC) Consider Algorithm 2. Assume everything in Theorem 2.2. Its conclusion holds with probability at least* $1 - (L+1)n^{-10} - c_{approxGM} - C\tilde{\kappa}^2\log(1/\epsilon)\big(Ln^{-10} - c_{approxGM}\big)$

3. Proof Outline for Theorem 2.2

The theorem statement is a subset of the following main claim. Let $\delta_t = \boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)$ and $\mu_u = 8\kappa^2\mu$. Let $\mathcal{J}_{good}$ denote the set of good (honest) nodes and let ℓ_1 be *any one* good node, i.e., any one entry of $\mathcal{J}_{good}$.

Claim 3.1. w.p. at least $1 - 3Lp_3 - t(Lp_2 + 2Lp_1)$

$$\text{(i)}\quad \delta_t \le (1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})^t\delta_0 + \eta p\sigma^{*2}_{\max}C8.6G_B\sum_{w=0}^{t-1}(1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})^w$$

(ii) $\|\boldsymbol{u}_t^j\| \le (1 - \frac{0.3}{\tilde{\kappa}^2})^t\|\boldsymbol{u}_0^j\| + 2\mu\sqrt{r/n}\sum_{w=0}^{t-1}(1 - \frac{0.3}{\tilde{\kappa}^2})^w$, (iii) $\delta_t \le \delta_0$, and (iv) $\mathcal{J}_{good} \subseteq \mathcal{I}_t$. Here $p_1 = \exp(\log\tilde{q} - c\frac{pn}{\max(\tilde{\kappa}^8\mu^2, \tilde{\kappa}^6\mu_u\mu)r^2}) + \exp(\log\tilde{q} - c\frac{pn}{r^2\tilde{\kappa}^4\mu_u^2})$, $p_2 = \exp(\log\tilde{q} - c\frac{pn}{\tilde{\kappa}^4\mu^2r^2})$, and $p_3 = n^{-10}$.

3.1. Proving Claim 3.1

This claim is proved using an induction argument and the following five lemmas. Lemma 3.2 proves the base case, while the others are used to prove the induction step. Claim (iv) follows using Lemma 3.4. Once we have $\mathcal{J}_{good} \subseteq \mathcal{I}_t$ Lemma 3.6, 3.7 holds. Claim (ii) follows by construction (see Fact 3.5) and Lemma 3.7. Claim (i) follows by substituting the bounds on Err Lemma 3.6, and denominator term Lemma 3.7 in algebra Lemma 3.3. Since doing a QR step results in a denominator term $\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}$ Lemma 3.7. To obtain a bound on this term Claim (iii) is required which is a consequence of Claim (i) and $G_B \le \frac{c}{\tilde{\kappa}^2}$. Full proof is in Appendix A.

Lemma 3.2 (Initialization). *Let Assumption 1, and 2 holds. Assume* $p \geq C\tilde{\kappa}^2 r^2 \mu \log \widetilde{q}/(n\delta_0^2)$. *Let* $p_3 = n^{-10}$. *Then,*

1. *w.p. at least* $1 - Lp_3$, $\mathcal{J}_{good} \subseteq \mathcal{I}_0$
2. *w.p. at least* $1 - 3Lp_3$, $\mathrm{SD}_F(\boldsymbol{U}_0, \boldsymbol{U}^*) \leq \delta_0$.
3. *w.p. at least* $1-3Lp_3$, $\boldsymbol{U}_0$ *is* 1.5μ *row-incoherent, i.e.,* $\|\boldsymbol{u}_0^j\| \leq 1.5\mu\sqrt{r/n}$ *for all* $j \in [n]$.

Lemma 3.3 (Algebra Lemma). *Let* $\mathrm{Err} := \nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]$. *For* $[\boldsymbol{U}_{t-1} - \eta\nabla_{Kr}] \overset{QR}{=} \boldsymbol{U}_t\boldsymbol{R}^+$ *we have*

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \frac{\|\boldsymbol{I}_r - \eta p \boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top\|\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_{t-1}) + \eta\|\mathrm{Err}\|_F}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}$$

Lemma 3.4 (Good nodes contained in filtered set). *Suppose that* $\mathcal{J}_{good} \subseteq \mathcal{I}_{t-1}$. *Then, w.p. at least* $1-Lp_2$, *at iteration* t, $\mathcal{J}_{good} \subseteq \mathcal{I}_t$. *Here* $p_2 = \exp(\log \widetilde{q} - cpn/\tilde{\kappa}^4\mu^2 r^2)$.

Fact 3.5 (Incoherence of U_{temp} for any $\ell \in \mathcal{I}_t$). $\boldsymbol{U}_{temp} = \boldsymbol{U}_{t-1} - \eta\nabla_\ell$, for any $\ell \in \mathcal{I}_t$, by construction (Line 25 of Algorithm 1) $\|\boldsymbol{u}^j{}_{temp}\| \leq \big(1 - \frac{0.4}{\tilde{\kappa}^2}\big)\|\boldsymbol{u}^j{}_{t-1}\| + 1.4\mu\sqrt{\frac{r}{n}}$.

Fact says any gradient with index in $\mathcal{I}_t$ gives incoherent $\boldsymbol{U}_{temp}$. The set $\mathcal{I}_t$ contains all of $\mathcal{J}_{good}$ but also can contain some of the Byz gradients. Our construction of the filtered set $\mathcal{I}_t$ guarantees that all gradients in it give incoherent $\boldsymbol{U}_{temp}$. Thus the Krum gradient (which is one of the gradients from $\mathcal{I}_t$) also gives incoherent $\boldsymbol{U}_{temp}$.

Lemma 3.6 (Bounding Err). *Assume* $\mathcal{J}_{good} \subseteq \mathcal{I}_t$ *and Assumption 1, 2, 3 holds. Let* $p_1 = \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{\max(\tilde{\kappa}^4\mu^2, \tilde{\kappa}^2\mu_u\mu, \mu)r^2}) + \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{r^2\mu_u^2})$. *We have w.p. at least* $1 - 2Lp_1$,

$$\|\nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}]\|_F \leq Cp\sigma_{\max}^{*2}(8\epsilon\delta_{t-1} + 4.3G_B) \quad (1)$$

Lemma 3.7 (Bounding denominator). *Consider the setting of Lemma 3.6. If* η *satisfies* $\eta p\sigma_{\max}^{*2}((2.5 + C8\epsilon)\delta_{t-1} + C4.3G_B) < 1$, *then w.p. at least* $1 - 2Lp_1$, $\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})} \leq 1 + \eta p\sigma_{\max}^{*2}((5 + C16\epsilon)\delta_{t-1} + C8.6G_B)$

The algebra lemma and the denominator lemma follow using ideas similar to those in (Abbasi & Vaswani, 2024). We show next how to prove the others.

Bounding error in B and showing its incoherence. We borrow this from (Abbasi & Vaswani, 2024). It does not change because $\boldsymbol{b}_k$s are updated locally at the nodes, and thus, we do not need to worry about Byzantine resilience.

Lemma 3.8 (Lemma 4.2, 4.3 (Abbasi & Vaswani, 2024)). *For any node* $\ell \in \mathcal{J}_{good}$. *Assume* $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$. *Then, w.p. at least* $1 - \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{r^2\mu_u^2})$,

1. $\|\boldsymbol{B}_\ell - \boldsymbol{G}_\ell\|_F \leq \epsilon\delta_{t-1}\sigma_{\max}^*$ *and* $\|\boldsymbol{X}_\ell - \boldsymbol{X}_\ell^*\|_F \leq 2\delta_{t-1}\sigma_{\max}^*$
2. $\sigma_{\max}(\boldsymbol{B}_\ell) \leq (1 + \delta_{t-1})\sigma_{\max}^*$ *and* $\sigma_{\min}(\boldsymbol{B}_\ell) \geq \sqrt{1 - \delta_{t-1}^2}\sigma_{\min}^* - \delta_{t-1}\sigma_{\max}^*$. *Thus, if* $\delta_{t-1} \leq c/\tilde{\kappa}^2$, *then* $\sigma_{\min}(\boldsymbol{B}_\ell) \geq 0.9\sigma_{\min}^*$ *and* $\sigma_{\max}(\boldsymbol{B}_\ell) \leq 1.1\sigma_{\max}^*$
3. $\|\boldsymbol{b}_k\| \leq 1.1\sigma_1^*\mu\sqrt{r/q}$.

This is used in the proof of the lemmas needed for the Err bound. It is also used to prove the $\|\boldsymbol{X}_T - \boldsymbol{X}^*\|_F$ bound of the theorem.

3.2. Proof of Initialization lemma

We need to first show that each good node returns an accurate enough estimate and one that is also incoherent. Item one of Lemma 3.2 follows using second part of (Abbasi & Vaswani, 2024, Lemma 4.1) that $\boldsymbol{U}_{0\ell}$ is 1.5μ row-incoherent for all $\ell \in \mathcal{J}_{good}$ and then using union bound over $\mathcal{J}_{good}$.

Next we need to analyze the proposed Subspace Krum approach. This requires using the following Krum Lemma 3.9 which we modified from (Blanchard et al., 2017) to include probabilistic argument.

Lemma 3.9 (Krum (Blanchard et al., 2017)). *Let* $\boldsymbol{z}_\ell \subseteq \Re^n$, *for* $\ell \in [L]$ *and let* z_{Kr} *denote the vector selected by Krum operator* $Kr = \mathrm{Krum}\{\boldsymbol{z}_\ell\}_{\ell=1}^L$. *For a* $\tau < 0.4$, *suppose that, for at least* $(1-\tau)L$ $\boldsymbol{z}_\ell$*'s,*

$$\Pr\{\|\boldsymbol{z}_\ell - \tilde{\boldsymbol{z}}\| \leq \epsilon\|\tilde{\boldsymbol{z}}\|\} \geq 1 - p$$

Then, w.p. at least $1 - 2L(1-\tau)p$,

$$\|\boldsymbol{z}_{Kr} - \tilde{\boldsymbol{z}}\| \leq 10\epsilon\|\tilde{\boldsymbol{z}}\|$$

Proof. Proof is in Appendix G □

Subspace Krum is modified version of Krum for subspaces using the idea from (Singh & Vaswani, 2024c) that the Frobenius norm of the difference between two subspace projection matrices is within a constant factor of the subspace distance between their respective subspaces. Therefore Krum is calculated for $\mathcal{P}_{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{U}^\top$'s where $\boldsymbol{U}$ is any subspace with the distance metric as $\|\mathcal{P}_{\boldsymbol{U}_1} - \mathcal{P}_{\boldsymbol{U}_2}\|_F^2$. We next give Subspace Krum Lemma 3.10

Lemma 3.10 (Subspace-Krum). *Consider Algorithm 1 Lines 13-15. For a* $\tau < 0.4$, *suppose that, for at least* $(1-\tau)L$ $\boldsymbol{U}_\ell$*'s,*

$$\Pr(\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_\ell) \leq \delta) \geq 1 - p$$

then, w.p. at least $1 - 2L(1-\tau)p$,

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_{out}) \leq 10\delta.$$

Fact 3.11. For any $\ell \in \mathcal{I}_0$. By construction (Line 10 of Algorithm 1) $\|\boldsymbol{u}_{0\ell}^j\| \leq 1.5\mu\sqrt{r/n}$ for all $j \in [n]$.

Item two of Lemma 3.2 follows using first part of Lemma 3.2, (Abbasi & Vaswani, 2024, Lemma 4.1), and Lemma 3.10. From (Abbasi & Vaswani, 2024, Lemma 4.1) we have $\boldsymbol{SD}_F(\boldsymbol{U}_{0\ell}, \boldsymbol{U}^*) \leq \delta' = \delta_0/10$ w.h.p. for all $\ell \in \mathcal{J}_{good}$. From first part of Lemma 3.2 $\mathcal{J}_{good} \subseteq \mathcal{I}_0$ w.h.p. implies using Lemma 3.10 $\boldsymbol{SD}_F(\boldsymbol{U}_{Kr}, \boldsymbol{U}^*) \leq \delta_0$ w.h.p.

Lemma 3.2 item three follows from Fact 3.11 as $Kr \in \mathcal{I}_0$.

3.3. Proof of Lemmas 3.4 and 3.6

First we prove Lemma 3.4. This lemma follows using modified version of second part of (Abbasi & Vaswani, 2024, Lemma 4.6), where for $\boldsymbol{U}_{temp} = \boldsymbol{U}_{t-1} - \eta \nabla_\ell$ we have $\|\boldsymbol{u}^j_{temp}\| \leq (1 - 0.4/\tilde{\kappa}^2)\|\boldsymbol{u}^j_{t-1}\| + 1.4\mu\sqrt{r/n}$ for all $j \in [n]$ w.h.p. and then using union bound over $\mathcal{J}_{good}$ we have the required proof.

Once we have $\mathcal{J}_{good} \subseteq \mathcal{I}_t$, then Lemma 3.6 is a direct consequence of Lemma 3.9, and 3.12 which we give next.

Lemma 3.12. *Assume $\mathcal{J}_{good} \subseteq \mathcal{I}_t$, and Assumption 2, 3 holds. Then for all $\ell \in \mathcal{J}_{good}$,*

1. *w.p. at least* $1 - \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{\max(\tilde{\kappa}^4\mu^2, \tilde{\kappa}^2\mu_u\mu)r^2})$
$\|\nabla_\ell - \mathbb{E}[\nabla_\ell]\|_F \leq \epsilon p \sigma^{*2}_{\max} \delta_{t-1}$
2. *w.p. at least* $1 - \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{\max(\tilde{\kappa}^4\mu^2, \tilde{\kappa}^2\mu_u\mu)r^2}) - \exp(\log \widetilde{q} - c\frac{\epsilon^2 pn}{r^2\mu_u^2})$
$\|\nabla_\ell - \mathbb{E}[\nabla_{\ell_1}]\|_F \leq p\sigma^{*2}_{\max}(8\epsilon\delta_{t-1} + 4.3G_B)$

Proof. Item one follows directly from (Abbasi & Vaswani, 2024, Lemma 4.5) which uses matrix Bernstein inequality and Lemma 3.8 in its proof. We prove item two in Appendix B. This follows using the first item and careful algebra that allows us to express $\|\nabla_\ell - \mathbb{E}[\nabla_{\ell_1}]\|_F$ in terms of $\|\nabla_\ell - \mathbb{E}[\nabla_\ell]\|_F, \|\boldsymbol{B}_\ell - \boldsymbol{G}_\ell\|_F, \sigma_{\max}(\boldsymbol{B}_\ell)$, and $\|\boldsymbol{B}^*_\ell - \boldsymbol{B}^*_{\ell_1}\|_F$. And since $\mathcal{J}_{good} \subseteq \mathcal{I}_t$ i.e., $\boldsymbol{U}_t$ is row incoherent we can bound each term using (Abbasi & Vaswani, 2024, Lemma 4.5), Lemma 3.8, and Assumption 3 respectively. Proof is in Appendix B. □

4. Secure Federated LRCS and LRPR

In the writing below $|\boldsymbol{z}|$ and $sign(\boldsymbol{z})$ return the element-wise magnitude and sign of $\boldsymbol{z}$ respectively. LRCS and LRPR involve recovering $\boldsymbol{X}^*$ from $\boldsymbol{y}_k := \boldsymbol{A}_k\boldsymbol{x}^*_k, k \in [q]$ (LRCS) and from $\boldsymbol{z}_k := |\boldsymbol{y}_k|, k \in [q]$ (LRPR), with each $\boldsymbol{y}_k$ being an m-length observed vector with $m < n$, and the measurement/sketching matrices $\boldsymbol{A}_k \in \Re^{m\times n}$ being known dense matrices (random Gaussian for guarantees) that are mutually independent over k. We consider the vertically federated setting, i.e., distinct subsets of columns of $\boldsymbol{Y}$ are available at different nodes. These are completely different problems than federated LRMC because here the matrices $\boldsymbol{A}_k$ are dense. This means there is asymmetry, we get dense measurements of each column but not of the different rows. Consequently, we only need right singular vectors' incoherence (instead of that of both). Also, because of the use of random Gaussian matrices, the measurements are now unbounded and we need different concentration inequalities to analyze these problems. However, in spite of these differences, we show how simple modifications of our algorithm provide very similar guarantees for both these problems as well. The following assumption replaces Assumption 2.

Assumption 4 (right singular vectors' incoherence). $\max_{k\in\mathcal{S}_\ell} \|\boldsymbol{b}^*_k\| \leq \mu\sqrt{r/\widetilde{q}}\sigma_{\max}(\boldsymbol{X}^*_\ell)$ for a constant $\mu \geq 1$.

Consider Algorithm 1 with the following changes: (1) the node initialization step replaced by the LRCS initialization from (Vaswani, 2024); and (2) remove the filtering step from both initialization and GD.

Theorem 4.1. *(Krum-AltGDmin for LRCS) Consider the algorithm as described above. Assume that Assumptions 1, 3 and 4 hold, and $G_B \leq \frac{c}{\tilde{\kappa}^2}$. The conclusions of Theorem 2.2 hold if $m\widetilde{q} \geq \tilde{\kappa}^{10}\mu^2 nr^2 \log(\frac{1}{\epsilon})$, and $m \geq \tilde{\kappa}^6 \max(\log \widetilde{q}, r)\log(\frac{1}{\epsilon})$,*

For LRPR, the algorithm for LRCS needs three simple modifications. (1) The node initialization needs to be the one for LRPR introduced in (Nayer & Vaswani, 2023, on arXiv since Feb. 2021). (2) The update of $\boldsymbol{b}_k$ involves solving an r-dimensional standard phase retrieval problem, e.g., using the RWF algorithm of (Zhang, Zhou, Liang, & Chi, 2017). (2) We estimate the phase (sign in case of real measurements which is what we consider here) after updating $\boldsymbol{b}_k$ as $sign(\boldsymbol{A}_k\boldsymbol{U}\boldsymbol{b}_k)$ and use this to obtain $\hat{\boldsymbol{y}}_k = sign(\boldsymbol{A}_k\boldsymbol{U}\boldsymbol{b}_k) \odot \boldsymbol{z}_k$. This is used to replace $\boldsymbol{y}_k$ in the gradient expression. Denote this gradient as $\hat{\nabla}_\ell$

Corollary 4.2. *(Krum-AltGDmin for LRPR) Consider the algorithm as described above. The conclusions of Theorem 4.1 hold if $m\widetilde{q} \geq \max(\tilde{\kappa}^{10}\mu^2 nr^3 \log(\frac{1}{\epsilon}), \tilde{\kappa}^8\mu^2 nr^2 \log(\frac{1}{\epsilon}))$, and $m \geq \tilde{\kappa}^6 \max(\log \widetilde{q}, r)\log(\frac{1}{\epsilon})$,*

LRCS needs order $nr^2 \log(1/\epsilon)$ samples which is also what LRMC needs. LRPR needs r times more samples for its initialization since it is a more difficult problem.

Remark 4.3. Both above results also hold apply for GM.

Proof. Our proof techniques introduced for LRMC apply here with minimal changes. For LRCS, only three things change: (i) we do not need to prove incoherence of $\boldsymbol{U}$ or worry about the filtering step (which is removed from the algorithm); (ii) the concentration bound lemmas change - the bound on $\|\boldsymbol{B} - \boldsymbol{U}^\top\boldsymbol{X}^*\|_F$ is obtained using Lemma 3 of (Vaswani, 2024) and the gradient deviation bound in the first part of Lemma 3.12 given above gets replaced by Lemma 5 of (Vaswani, 2024).

For LRPR, the following additional modifications are needed. (1) The $\boldsymbol{B}$ lemma gets replaced by Lemma 3.3 of (Nayer & Vaswani, 2023, on arXiv since Feb. 2021). (2) The computed ∇_ℓ uses $\hat{\boldsymbol{y}}_k$ defined above. Thus, in order to obtain a bound similar to the first item of Lemma 3.11, we need two steps: (i) We first need to bound $\|\hat{\nabla}_\ell - \nabla_\ell\|_F$ with ∇_ℓ being the gradient expression if the linear measurement $\boldsymbol{y}_k$ were available, i.e. the gradient used in case of LRCS. $\|\hat{\nabla}_\ell - \nabla_\ell\|_F$ is the same as the $ErrPh$ term that is bounded in Lemma 5.3 of (Nayer & Vaswani, 2023, on arXiv since Feb. 2021) (LRPR section). (ii) Next we use Lemma 5 of (Vaswani, 2024) and triangle inequality to finally get a bound on $\|\hat{\nabla}_\ell - \mathbb{E}[\nabla_{\ell_1}]\|_F$ to replace the first item of our 3.12. □

5. Experiments

LRMC: Figure 1a. We plot the average subspace distance $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ against Time in seconds over 100 Monte Carlo (MC) runs. The averaging is over the observed entries of $\boldsymbol{X}^*$, with each entry being observed independently with probability p at every MC run. The matrix $\boldsymbol{X}^* = \boldsymbol{U}^*\boldsymbol{B}^*$ is generated once by letting $\boldsymbol{U}^* \in \Re^{n\times r}$ be the left-singular vectors of an $n \times r$ random i.i.d. Gaussian matrix and $\boldsymbol{B}^* \in \Re^{r\times q}$ be a random Gaussian matrix. Thus, $\boldsymbol{X}^*$ has $\mu = O(1)$ incoherence. At each iteration, $L_{byz} = 8$ of the $L = 20$ gradients communicated from the nodes to the center are corrupted. We consider Reverse Gradient Attack for each experiment where we corrupt the L_{byz} gradients by setting $\nabla_{byz} = -\sum_{\ell=1}^{L} \nabla_\ell$. This forces the GD step to move in the reverse direction of the true gradient. We set the stepsize $\eta = 1/p(\sigma^{*2}_{\max})$, where $\sigma^*_{\max} = \max_{\ell\in[L]} \sigma_{\max}(\boldsymbol{X}^*_\ell)$, is estimated as $\sigma^*_{\max} \simeq \max_{\ell\in[L]} \sigma_{\max}(\boldsymbol{Y}_\ell)/p$. We compared Krum-AltGDmin, GM-AltGDmin, and CWMedian-AltGDmin. We use Weiszfeld's algorithm with 1000 iterations to approximate the Geometric Median (GM).

Observation from Figure 1a. Theoretically, GM-AltGDmin has similar sample complexity to Krum-AltGDmin, so it also converges in the experiments. CW-Median does not converge due to its large sample complexity requirement. GM-AltGDmin is slower than Krum, possibly for two reasons mentioned below:

a) We use an approximate algorithm Weiszfeld's algorithm (Weiszfeld, 1937; Beck & Sabach, 2015) to approximate GM. Weiszfeld's algorithm (Beck & Sabach, 2015, Theorem 5.1) is known to converge, but the number of iterations is not specified. In theory, GM can be faster when using (Cohen, Lee, Miller, Pachocki, & Sidford, 2016, Algorithm 1). However, that algorithm is complex and to our best knowledge has no known experimental results.

b) As shown in Table 1, Krum has a compute cost of $nr^2L^2\log(1/\epsilon)$, and GM has a compute cost of $nr^2L\log^3(\frac{L}{\epsilon_{approx}})\log(1/\epsilon)$ when using (Cohen et al., 2016, Algorithm 1). To see any speedup from GM in simulations, one would need to use (Cohen et al., 2016, Algorithm 1) with a large L. This also requires a large q to maintain accuracy. As a result, the total simulation time becomes much higher.

LRCS: Figure 1b. We plot the average subspace distance $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ vs Time in seconds over 100 Monte Carlo (MC) runs. The data generation part for $\boldsymbol{X}^*$ remains same as LRMC. For each Monte Carlo run we generated matrices $\boldsymbol{A}_k, k \in [q]$ with each entry being i.i.d. standard Gaussian and we set $\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x}^*_k, k \in [q]$. We used $\eta = 1/m(\sigma^{*2}_{\max})$, where $\sigma^*_{\max} = \max_{\ell\in[L]} \sigma_{\max}(\boldsymbol{X}^*_\ell)$. We used Weiszfeld's Algorithm with 1000 iterations to approximate Geometric Median (GM).

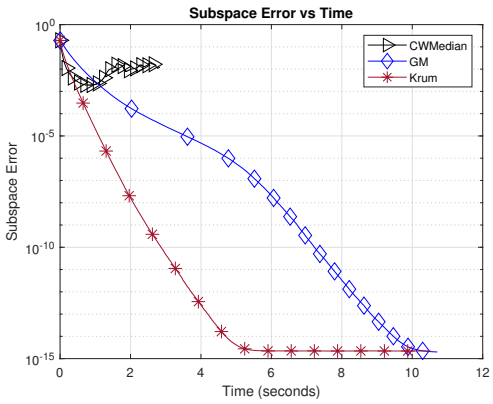
Observation from Figure 1b. Same as LRMC observation Section 5.

Real world MovieLens 1M dataset (Real World): Figure 1c. We test our proposed algorithm on the real-world MovieLens 1M dataset (Harper & Konstan, 2015), which contains $1,000,209$ ratings for about $3,900$ movies from $6,040$ users. The algorithms are compared using the relative error $\|(\boldsymbol{X}^* - \boldsymbol{X})_\Omega\|/\|\boldsymbol{X}^*_\Omega\|$, shown on the y-axis, where Ω is the set of observed entries. This is used because, in real data, unlike synthetic data, the true rank-r $\boldsymbol{U}^*$ is not known.

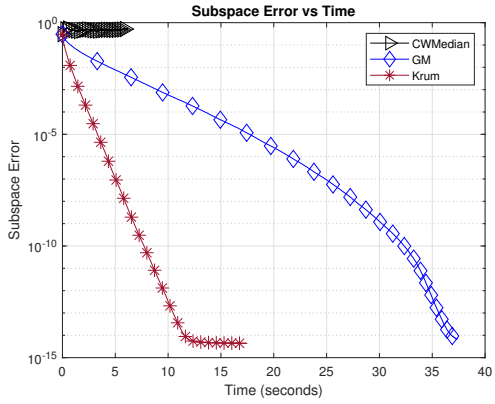
Observation from Figure 1c. We observe that our algorithm, Byz-AltGDmin-LRMC, converges on the federated MovieLens 1M dataset even with 40% Byzantine nodes ($L_{byz} = 8$, $L = 20$). We also observe that CWMedian converges in this setting due to the large number of samples ($n\widetilde{q}p$).

Heterogeneity Effect (LRMC): Figure 1d. We first generate $\boldsymbol{B}^*$ as a random Gaussian matrix. Each $\boldsymbol{B}^*_\ell$ is formed by selecting a subset of columns $k \in \mathcal{S}_\ell$ from $\boldsymbol{B}^*$. Then, with $L = 10$ total nodes, we multiply 5 randomly chosen $\boldsymbol{B}^*_\ell$ by a factor C to introduce heterogeneity across the datasets see Assumption 3 (Bounded heterogeneity). This was done once (outside Monte Carlo loop). For 25 Monte Carlo runs we did Reverse Gradient Attack and compared 'Krum-AltGDmin' for different values of $C = 1, 4, 6$.

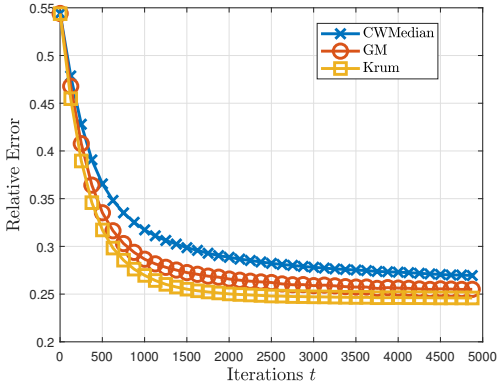
Observations from Figure 1d. It can be seen that increasing heterogeneity C means that the $\boldsymbol{SD}_F$ saturates higher, as shown in our main result Theorem 2.2.



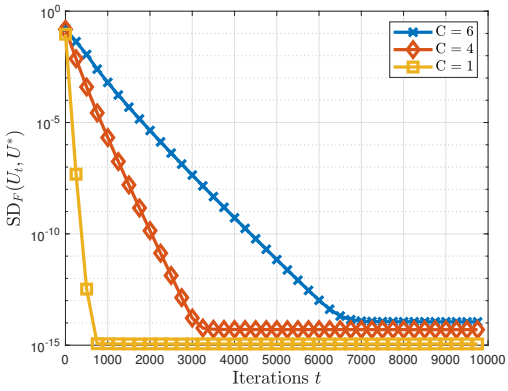
(a) **LRMC:** $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ vs Time(seconds) with $n = 1000$, $q = 500$, $r = 3$, $L = 20$, and $L_{byz} = 8$.



(b) **LRCS:** $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ vs Time(seconds) with $n = 1000$, $m = 50$, $q = 1000$, $r = 3$, $L = 20$, and $L_{byz} = 8$.



(c) **Real-world MovieLens 1M dataset**: $\|(\boldsymbol{X}^* - \boldsymbol{X})_\Omega\|/\|\boldsymbol{X}^*_\Omega\|$ vs Iteration t with $n = 6040$, $q = 3940$, $r = 3$, $p = 0.041902$, $L = 20$, and $L_{byz} = 8$.



(d) **Heterogeneity Effect (LRMC):** $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ vs Iteration t with $n = 200$, $q = 1000$, $r = 4$,$L = 10$, $L_{byz} = 2$, $p = 0.4$, and using Krum.

6. Addressing reviewer comments

We have strengthened the experiments by adding results for GM and for the LRCS problem. We have now also evaluated our algorithm on the MovieLens dataset. We have included and compared the related works as mentioned by the reviewer. We have also highlighted the technical differences between this work and the related ones.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abbasi, A. A., Moothedath, S., & Vaswani, N. (2023). Fast federated low rank matrix completion. In *2023 59th annual allerton conference on communication, control, and computing (allerton)* (pp. 1–6).

Abbasi, A. A., & Vaswani, N. (2024). Efficient federated low rank matrix completion. *arXiv preprint arXiv:2405.06569*.

Acharya, A., Hashemi, A., Jain, P., Sanghavi, S., Dhillon, I. S., & Topcu, U. (2022). Robust training in high dimensions via block coordinate geometric median descent. In *International Conference on Artificial Intelligence and Statistics* (pp. 11145–11168).

Alistarh, D., Allen-Zhu, Z., & Li, J. (2018). Byzantine stochastic gradient descent. *Advances in Neural Information Processing Systems*, *31*.

Allen-Zhu, Z., Ebrahimian, F., Li, J., & Alistarh, D. (2020). Byzantine-resilient non-convex stochastic gradient descent. *arXiv preprint arXiv:2012.14368*.

Allouah, Y., Farhadkhani, S., Guerraoui, R., Gupta, N., Pinot, R., & Stephan, J. (2023). Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In *International conference on artificial intelligence and statistics* (pp. 1232–1300).

Anaraki, F. P., & Hughes, S. (2014). Memory and computation efficient pca via very sparse random projections. In *International conference on machine learning* (pp. 1341–1349).

Azizyan, M., Krishnamurthy, A., & Singh, A. (2014). Subspace learning from extremely compressed measurements. In *2014 48th asilomar conference on signals, systems and computers* (pp. 311–315).

Babu, S., Lingala, S. G., & Vaswani, N. (2023). Fast low rank compressive sensing for accelerated dynamic MRI. *IEEE Trans. Comput. Imag.*.

Beck, A., & Sabach, S. (2015). Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, *164*(1), 1–40.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, *30*.

Candes, E. J., & Recht, B. (2008). Exact matrix completion via convex optimization. *Found. of Comput. Math*(9), 717-772.

Cao, X., Fang, M., Liu, J., & Gong, N. Z. (2020). Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*.

Cao, X., & Lai, L. (2019). Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Transactions on Signal Processing*, *67*(22), 5850–5864.

Chen, Y., Chi, Y., Fan, J., Ma, C., et al. (2021). Spectral methods for data science: A statistical perspective. *Foundations and Trends*
textregistered in Machine Learning, *14*(5), 566–806.

Chen, Y., Su, L., & Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *1*(2), 1–25.

Cherapanamjeri, Y., Gupta, K., & Jain, P. (2017). Nearly optimal robust matrix completion. In *International conference on machine learning* (pp. 797–805).

Cohen, M. B., Lee, Y. T., Miller, G., Pachocki, J., & Sidford, A. (2016). Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing* (pp. 9–21).

Collins, L., Hassani, H., Mokhtari, A., & Shakkottai, S. (2021). Exploiting shared representations for personalized federated learning. In *International conference on machine learning* (pp. 2089–2099).

Dadras, A., Stich, S. U., & Yurtsever, A. (2024). Personalized federated learning via low-rank matrix factorization. In *Opt 2024: Optimization for machine learning.*

Data, D., & Diggavi, S. (2021). Byzantine-resilient SGD in high dimensions on heterogeneous data. In *2021 IEEE International Symposium on Information Theory (ISIT)* (pp. 2310–2315).

Data, D., & Diggavi, S. N. (2023). Byzantine-resilient high-dimensional federated learning. *IEEE Transactions on Information Theory*, *69*(10), 6639–6670.

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, *27*.

Ghosh, A., Hong, J., Yin, D., & Ramchandran, K. (2019). Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*.

Guerraoui, R., Rouault, S., et al. (2018). The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning* (pp. 3521–3530).

Haldar, J. P., & Liang, Z.-P. (2010). Spatiotemporal imaging with partially separable functions: A matrix recovery approach. In *2010 ieee international symposium on biomedical imaging: From nano to macro* (pp. 716–719).

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, *5*(4), 1–19.

He, X., Ling, Q., & Chen, T. (2019). Byzantine-robust

stochastic gradient descent for distributed low-rank matrix completion. In *2019 ieee data science workshop (dsw)* (pp. 322–326).

Jagatap, G., Chen, Z., Nayer, S., Hegde, C., & Vaswani, N. (2019). Sample efficient fourier ptychography for structured data. *IEEE Transactions on Computational Imaging*, *6*, 344–357.

Jain, P., & Netrapalli, P. (2015). Fast exact matrix completion with finite samples. In *Conference on learning theory* (pp. 1007–1034).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Li, L., Xu, W., Chen, T., Giannakis, G. B., & Ling, Q. (2019). RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 1544–1551).

Lingala, S. G., Hu, Y., DiBella, E., & Jacob, M. (2011). Accelerated dynamic mri exploiting sparsity and low-rank structure: kt slr. *IEEE transactions on medical imaging*, *30*(5), 1042–1054.

Lu, S., Li, R., Chen, X., & Ma, Y. (2022). Defense against local model poisoning attacks to byzantine-robust federated learning. *Frontiers of Computer Science*, *16*(6), 166337.

Nashed, M. (1968). A decomposition relative to convex sets. *Proceedings of the American Mathematical Society*, *19*(4), 782–786.

Nayer, S., Narayanamurthy, P., & Vaswani, N. (2019). Phaseless pca: Low-rank matrix recovery from column-wise phaseless measurements. In *International conference on machine learning* (pp. 4762–4770).

Nayer, S., & Vaswani, N. (2021). Sample-efficient low rank phase retrieval. *IEEE Transactions on Information Theory*, *67*(12), 8190–8206.

Nayer, S., & Vaswani, N. (2023, on arXiv since Feb. 2021, Feb.). Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections. *IEEE Trans. Info. Th.*.

Netrapalli, P., Jain, P., & Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization..

Pillutla, K., Kakade, S. M., & Harchaoui, Z. (2019). Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*.

Regatti, J., Chen, H., & Gupta, A. (2022). Byzantine Resilience With Reputation Scores. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 1–8).

Shome, D., & Kar, T. (2021). Fedaffect: Few-shot federated learning for facial expression recognition. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 4168–4175).

Singh, A. P., & Vaswani, N. (2024a). Byzantine resilient and fast federated few-shot learning. In *Forty-first international conference on machine learning.*

Singh, A. P., & Vaswani, N. (2024b). Byzantine-resilient federated pca and low rank column-wise sensing. *IEEE Transactions on Information Theory*.

Singh, A. P., & Vaswani, N. (2024c). Byzantine-resilient federated principal subspace estimation. In *2024 ieee international symposium on information theory (isit)* (pp. 2514–2519).

Srinivasa, R. S., Lee, K., Junge, M., & Romberg, J. (2019). Decentralized sketching of low rank matrices. In (pp. 10101–10110).

Thekumparampil, K. K., Jain, P., Netrapalli, P., & Oh, S. (2021). Statistically and computationally efficient linear meta-representation learning. *Advances in Neural Information Processing Systems*, *34*, 18487–18500.

Vaswani, N. (2024). Efficient federated low rank matrix recovery via alternating gd and minimization: A simple proof. *IEEE Trans. Info. Th.*.

Weiszfeld, E. (1937). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, *43*, 355–386.

Wu, Z., Ling, Q., Chen, T., & Giannakis, G. B. (2020). Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, *68*, 4583–4596.

Xie, C., Koyejo, S., & Gupta, I. (2019). Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning* (pp. 6893–6901).

Yao, J., Xu, Z., Huang, X., & Huang, J. (2018). An efficient algorithm for dynamic mri using low-rank and total variation regularizations. *Medical image analysis*, *44*, 14–27.

Yi, X., Park, D., Chen, Y., & Caramanis, C. (2016). Fast algorithms for robust pca via gradient descent. *Advances in neural information processing systems*, *29*.

Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning* (pp. 5650–5659).

Zhang, H., Zhou, Y., Liang, Y., & Chi, Y. (2017). A nonconvex approach for phase retrieval: Reshaped wirtinger flow and incremental algorithms. *The Journal of Machine Learning Research*, *18*(1), 5164–5198.

Zheng, Q., & Lafferty, J. (2016). Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*.

A. Proof of Claim 3.1

This claim is proved using an induction argument similar to that of (Abbasi & Vaswani, 2024).

Proof. Base case: From Lemma 3.2 (i), (ii), (iii) and (iv) holds for $t \equiv 0$ w.p. at least $1 - 3Lp_3$.

Induction assumption: Assume that the Claim 3.1 holds for $t \equiv t-1$.

Induction proof: From Lemma 3.4 w.p. at least $1 - Lp_2$, $\mathcal{J}_{good} \subseteq \mathcal{I}_t$. Hence Claim 3.1 (iv) holds for $t \equiv t$. Also $|\mathcal{J}_{good}| = L - L_{byz} > 1$ implies set $\mathcal{I}_t \neq \emptyset$. Now since $Kr \in \mathcal{I}_t$ implies using Fact 3.5 $\boldsymbol{U}_{temp} = \boldsymbol{U}_{t-1} - \eta \nabla_{Kr}$ satisfies

$$\|\boldsymbol{u}^j{}_{temp}\| \leq \left(1 - \frac{0.4}{\tilde{\kappa}^2}\right)\|\boldsymbol{u}^j{}_{t-1}\| + 1.4\mu\sqrt{\frac{r}{n}}$$

We bound $\|\boldsymbol{u}_t^j\| \leq \|\boldsymbol{u}_{temp}^j\|\|(\boldsymbol{R}^+)^{-1}\|$, where $\boldsymbol{U}_{temp} = \boldsymbol{U}_{t-1} - \eta\nabla_{Kr} \overset{\text{QR}}{=} \boldsymbol{U}_t\boldsymbol{R}^+$ (Line 29 of Algorithm 1).

$$\|(\boldsymbol{R}^+)^{-1}\| = \frac{1}{\sigma_{\min}(\boldsymbol{U}_{temp})} = \frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}$$

Since $\mathcal{J}_{good} \subseteq \mathcal{I}_t$, we can use Lemma 3.7. With $\epsilon = \frac{0.1}{C16\tilde{\kappa}^2}$, $\delta_{t-1} \leq \delta_0 = \frac{0.1}{5.1\tilde{\kappa}^2}$, and $G_B \leq \frac{0.2\cdot 0.1}{5.1\cdot 8.6C\tilde{\kappa}^2}$ we have w.p. at least $1 - 2Lp_1$,

$$\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})} \leq 1 + \frac{0.2\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2} \leq 1 + \frac{0.1}{\tilde{\kappa}^2} \tag{2}$$

Above we have used $\eta \leq \frac{0.5}{p\sigma^{*2}_{\max}}$. Hence we get

$$\begin{aligned}
\|\boldsymbol{u}_t^j\| &\leq \left(1 - \frac{0.4}{\tilde{\kappa}^2}\right)\|\boldsymbol{u}^j{}_{t-1}\|\|(\boldsymbol{R}^+)^{-1}\| + 1.4\mu\sqrt{\frac{r}{n}}\|(\boldsymbol{R}^+)^{-1}\| \\
&\leq \left(1 - \frac{0.4}{\tilde{\kappa}^2}\right)\left(1 + \frac{0.1}{\tilde{\kappa}^2}\right)\|\boldsymbol{u}^j{}_{t-1}\| + 1.4\mu\sqrt{\frac{r}{n}}\left(1 + \frac{0.1}{\tilde{\kappa}^2}\right) \\
&\leq \left(1 - \frac{0.3}{\tilde{\kappa}^2}\right)\|\boldsymbol{u}_{t-1}^j\| + 2\mu\sqrt{\frac{r}{n}} \\
&\leq (1 - \frac{0.3}{\tilde{\kappa}^2})^t\|\boldsymbol{u}_0^j\| \quad + [1 + (1 - \frac{0.3}{\tilde{\kappa}^2})^2 + \cdots + (1 - \frac{0.3}{\tilde{\kappa}^2})^{t-1}]2\mu\sqrt{\frac{r}{n}} \\
&\leq \|\boldsymbol{u}_0^j\| + \frac{\tilde{\kappa}^2}{0.3}2\mu\sqrt{\frac{r}{n}} \leq (1.5\mu + 7\tilde{\kappa}^2\mu)\sqrt{\frac{r}{n}} \\
&\leq \mu_u\sqrt{\frac{r}{n}}
\end{aligned}$$

The last inequality above used the infinite geometric series bound. This shows that $\boldsymbol{U}$ is μ_u-row-incoherent i.e., (ii) holds for $t \equiv t$. Now using Lemma 3.3 we get

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \frac{\|\boldsymbol{I}_r - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top\|\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_{t-1}) + \eta\|\text{Err}\|_F}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}. \tag{3}$$

Here $\text{Err} = \nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]$.

Again since $\mathcal{J}_{good} \subseteq \mathcal{I}_t$, we can use Lemma 3.6 which implies w.p. at least $1 - 2Lp_1$,

$$\|\text{Err}\|_F \leq Cp\sigma^{*2}_{\max}(8\epsilon\delta_{t-1}\sigma_1^* + 4.3G_B) \tag{4}$$

and from (2)

$$\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})} \leq 1 + \frac{0.2\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2} \tag{5}$$

By Lemma 3.8, $\sigma_{\min}(\boldsymbol{B}_{\ell_1}) \geq 0.9\sigma^*_{\min}$ and $\sigma_{\max}(\boldsymbol{B}_{\ell_1}) \leq 1.1\sigma^*_1$. Thus, if $\eta \leq 0.5/p\sigma^{*2}_{\max}$ then, $\boldsymbol{I} - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}^\intercal_{\ell_1}$ is positive semi-definite (psd) and so

$$\begin{aligned}\|\boldsymbol{I} - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}^\intercal_{\ell_1}\| &= \lambda_{\max}(\boldsymbol{I} - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}^\intercal_{\ell_1}) = 1 - \eta p\sigma^2_{\min}(\boldsymbol{B}_{\ell_1}) \\ &\leq 1 - 0.9\eta p\sigma^{*2}_{\min} = 1 - \frac{0.9\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2}\end{aligned} \tag{6}$$

Using (6), (4), and (5) in (3) we get

$$\begin{aligned}\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_t) &\leq (1 - \frac{0.9\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})(1 + \frac{0.2\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})\delta_{t-1} \\ &\quad + \eta p \cdot C\sigma^{*2}_{\max}(8\epsilon\delta_{t-1} + 4.3G_B)(1 + \frac{0.2\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2}) \\ &\text{Since } \epsilon = \frac{0.1}{C16\tilde{\kappa}^2}, (1 + \frac{0.2\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2}) \leq 2 \text{ we have} \\ &\leq (1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})\delta_{t-1} + \eta p\sigma^{*2}_{\max}C8.6G_B \qquad (7) \\ &\text{Since } G_B \leq \frac{0.2 \cdot 0.1}{5.1 \cdot 8.6C\tilde{\kappa}^2}, \delta_0 = \frac{0.1}{5.1\tilde{\kappa}^2}, \text{ we have} \\ &\leq (1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})\delta_0 + 0.1\delta_0 \\ &\leq 0.775\delta_0 \qquad (8)\end{aligned}$$

Above we have used $\eta \leq \frac{0.5}{p\sigma^{*2}_{\max}}$. From (8) we have $\delta_t \leq \delta_0$ implies Claim 3.1 (iii) holds for $t \equiv t$, and from (7) we have $\delta_t \leq (1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})^t\delta_0 + \eta p\sigma^{*2}_{\max}C8.6G_B\sum_{w=0}^{t-1}(1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})^w$ implies Claim 3.1 (i) holds for $t \equiv t$. Using union bound Claim 3.1 holds for $t \equiv t$ w.p. at least $1 - 3Lp_3 - t(Lp_2 + 2Lp_1)$. By principle of mathematical induction Claim 3.1 is true.

For $t \equiv T$ and using infinite geometric series bound

$$\delta_T \leq (1 - \frac{0.65\eta p\sigma^{*2}_{\max}}{\tilde{\kappa}^2})^T\delta_0 + 14C\tilde{\kappa}^2G_B$$

w.p. at least $1 - 3Lp_3 - T(Lp_2 + 2Lp_1)$.

Setting $T = C\tilde{\kappa}^2\log(1/\epsilon)$, $\delta_0 = \frac{c}{\tilde{\kappa}^2}$, and if $n\widetilde{q}p \geq C\tilde{\kappa}^{10}\mu^2\widetilde{q}r^2\log\widetilde{q}\log(1/\epsilon)$ then w.p. at least $1 - 3Lp_3 - T(Lp_2 + 2Lp_1) \geq 1 - 3Ln^{-10} - C\tilde{\kappa}^2\log(1/\epsilon)Ln^{-10}$

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_T) \leq \max\big(\epsilon, 14C\tilde{\kappa}^2G_B\big).$$

□

B. Proof of Grad Concentration Lemma 3.12

Proof. Item one follows directly from (Abbasi & Vaswani, 2024, Lemma 4.5).

Bounding $\|\nabla_\ell - \mathbb{E}[\nabla_{\ell_1}]\|_F$

$$\begin{aligned}\|\nabla_\ell - \mathbb{E}[\nabla_{\ell_1}]\|_F &\leq \|\nabla_\ell - \mathbb{E}[\nabla_\ell]\|_F + \|\mathbb{E}[\nabla_{\ell_1}] - \mathbb{E}[\nabla_\ell]\|_F \\ &\leq \epsilon p\delta_{t-1}\sigma^{*2}_{\max} \qquad \text{from Lemma 3.12 item 1} \\ &\quad + \|p(\boldsymbol{X}_{\ell_1} - \boldsymbol{X}^*_{\ell_1})\boldsymbol{B}^\top_{\ell_1} - p(\boldsymbol{X}_\ell - \boldsymbol{X}^*_\ell)\boldsymbol{B}^\top_\ell\|_F \qquad (9)\end{aligned}$$

w.p. at least $1-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{\max(\tilde{\kappa}^4\mu^2,\tilde{\kappa}^2\mu_u,\mu)r^2})$

Bounding $\|p(\boldsymbol{X}_{\ell_1}-\boldsymbol{X}^*_{\ell_1})\boldsymbol{B}_{\ell_1}^\top-p(\boldsymbol{X}_\ell-\boldsymbol{X}^*_\ell)\boldsymbol{B}_\ell^\top\|_F$

$$
\begin{aligned}
&\|p(\boldsymbol{X}_{\ell_1}-\boldsymbol{X}^*_{\ell_1})\boldsymbol{B}_{\ell_1}^\top-p(\boldsymbol{X}_\ell-\boldsymbol{X}^*_\ell)\boldsymbol{B}_\ell^\top\|_F\\
&=p\|\boldsymbol{U}(\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top)-\boldsymbol{X}^*_{\ell_1}\boldsymbol{B}_{\ell_1}^\top+\boldsymbol{X}^*_\ell\boldsymbol{B}_\ell^\top\|_F\\
&=p\|\boldsymbol{U}(\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top)-\boldsymbol{U}^*(\boldsymbol{B}^*_{\ell_1}\boldsymbol{B}_{\ell_1}^\top+\boldsymbol{B}^*_\ell\boldsymbol{B}_\ell^\top)\|_F\\
&=p\|\boldsymbol{U}(\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top)-\boldsymbol{U}^*(\boldsymbol{B}^*_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}^*_\ell\boldsymbol{B}_\ell^\top\pm\boldsymbol{B}^*_{\ell_1}\boldsymbol{B}_\ell^\top)\|_F\\
&=p\|\boldsymbol{U}(\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top)-\boldsymbol{U}^*(\boldsymbol{B}^*_{\ell_1}(\boldsymbol{B}_{\ell_1}-\boldsymbol{B}_\ell)^\top-(\boldsymbol{B}^*_\ell-\boldsymbol{B}^*_{\ell_1})\boldsymbol{B}_\ell^\top)\|_F\\
&=p\|\boldsymbol{U}(\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top-\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top\pm\boldsymbol{B}_{\ell_1}\boldsymbol{B}_\ell^\top)-\boldsymbol{U}^*(\boldsymbol{B}^*_{\ell_1}(\boldsymbol{B}_{\ell_1}-\boldsymbol{B}_\ell)^\top-(\boldsymbol{B}^*_\ell-\boldsymbol{B}^*_{\ell_1})\boldsymbol{B}_\ell^\top)\|_F\\
&=p\|\boldsymbol{U}\boldsymbol{B}_{\ell_1}(\boldsymbol{B}_{\ell_1}-\boldsymbol{B}_\ell)^\top+\boldsymbol{U}(\boldsymbol{B}_{\ell_1}-\boldsymbol{B}_\ell)\boldsymbol{B}_\ell^\top-\boldsymbol{U}^*\boldsymbol{B}^*_{\ell_1}(\boldsymbol{B}_{\ell_1}-\boldsymbol{B}_\ell)^\top+\boldsymbol{U}^*(\boldsymbol{B}^*_\ell-\boldsymbol{B}^*_{\ell_1})\boldsymbol{B}_\ell^\top\|_F\\
&\le p\big(\|\boldsymbol{U}\|\|\boldsymbol{B}_{\ell_1}\|+\|\boldsymbol{U}\|\|\boldsymbol{B}_\ell^\top\|+\|\boldsymbol{U}^*\|\|\boldsymbol{B}^*_{\ell_1}\|\big)\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\|_F+G_B\sigma^*_1\|\boldsymbol{U}^*\|\|\boldsymbol{B}_\ell^\top\|\big) && \text{from Assumption 3}\\
&\le p\big((1.1\sigma^*_1+1.1\sigma^*_1+\sigma^*_1)\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\|_F+1.1G_B\sigma^{*2}_{\max}\big) && \text{from Lemma 3.8}
\end{aligned}
$$

Now Bounding $\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\|_F$

$$
\begin{aligned}
\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\|_F&=\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\pm\boldsymbol{G}_{\ell_1}\|_F\\
&=\|\boldsymbol{G}_{\ell_1}-\boldsymbol{B}_{\ell_1}+\boldsymbol{B}_\ell-\boldsymbol{G}_{\ell_1}\|_F\\
&=\|\boldsymbol{G}_{\ell_1}-\boldsymbol{B}_{\ell_1}+\boldsymbol{B}_\ell-\boldsymbol{U}^\top\boldsymbol{X}^*_{\ell_1}\pm\boldsymbol{U}^\top\boldsymbol{X}^*_\ell\|_F\\
&=\|\boldsymbol{G}_{\ell_1}-\boldsymbol{B}_{\ell_1}+(\boldsymbol{B}_\ell-\boldsymbol{G}_\ell)-\boldsymbol{U}^\top(\boldsymbol{X}^*_{\ell_1}-\boldsymbol{X}^*_\ell)\|_F\\
&\le\|\boldsymbol{G}_{\ell_1}-\boldsymbol{B}_{\ell_1}\|_F+\|\boldsymbol{B}_\ell-\boldsymbol{G}_\ell\|_F+G_B\sigma^*_{\max}\\
&\le 2\epsilon\delta_{t-1}\sigma^*_1+G_B\sigma^*_{\max} && \text{from Lemma 3.8}
\end{aligned}
$$

w.p. at least $1-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{r^2\mu_u^2})$

Using this we get

$$
\begin{aligned}
&\|p(\boldsymbol{X}_{\ell_1}-\boldsymbol{X}^*_{\ell_1})\boldsymbol{B}_{\ell_1}^\top-p(\boldsymbol{X}_\ell-\boldsymbol{X}^*_\ell)\boldsymbol{B}_\ell^\top\|_F\\
&\le p\big((1.1\sigma^*_1+1.1\sigma^*_1+\sigma^*_1)\|\boldsymbol{B}_\ell-\boldsymbol{B}_{\ell_1}\|+1.1G_B\sigma^{*2}_{\max}\big)\\
&\le p\big((3.2\sigma^*_1)(2\epsilon\delta_{t-1}\sigma^*_1+G_B\sigma^*_{\max})+1.1G_B\sigma^{*2}_{\max}\big)\\
&=p(7\epsilon\delta_{t-1}\sigma^{*2}_{\max}+4.3\sigma^{*2}_{\max}G_B)
\end{aligned}
$$

w.p. at least $1-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{r^2\mu_u^2})$

This then implies w.p. at least $1-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{\max(\tilde{\kappa}^4\mu^2,\tilde{\kappa}^2\mu_u,\mu)r^2})-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{r^2\mu_u^2})$

$$
\begin{aligned}
\|\nabla_\ell-\mathbb{E}[\nabla_{\ell_1}]\|_F&\le\epsilon p\delta_{t-1}\sigma^{*2}_{\max}+\|p(\boldsymbol{X}_{\ell_1}-\boldsymbol{X}^*_{\ell_1})\boldsymbol{B}_{\ell_1}^\top-p(\boldsymbol{X}_\ell-\boldsymbol{X}^*_\ell)\boldsymbol{B}_\ell^\top\|_F\\
&\le\epsilon p\delta_{t-1}\sigma^{*2}_{\max}+p(7\epsilon\delta_{t-1}\sigma^{*2}_{\max}+4.3\sigma^{*2}_{\max}G_B)\\
&=p\sigma^{*2}_{\max}(8\epsilon\delta_{t-1}+4.3G_B)
\end{aligned}
$$

□

C. Proof of Lemma 3.6

Proof. From Lemma 3.12 item two for all $\ell\in\mathcal{J}_{good}$, w.p. at least $1-\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{\max(\tilde{\kappa}^4\mu^2,\tilde{\kappa}^2\mu_u,\mu)r^2})+\exp(\log\widetilde{q}-c\frac{\epsilon^2pn}{r^2\mu_u^2})$

$$\|\nabla_\ell-\mathbb{E}[\nabla_{\ell_1}]\|_F\le p\sigma^{*2}_{\max}(8\epsilon\delta_{t-1}+4.3G_B) \tag{10}$$

Using Lemma 3.9 and the fact $\|M\|_F = \|\text{vec}(M)\|_2$ for any matrix M we get w.p. at least $1 - 2Lp_1$

$$\|\nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}]\|_F \leq 10p\sigma^{*2}_{\max}(8\epsilon\delta_{t-1} + 4.3G_B)$$

□

D. Proof of Lemma 3.7

Proof.

$$\begin{aligned}
&\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})} \\
&= \frac{1}{\sigma_{min}(\boldsymbol{U}_{t-1} - \eta(\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})] + (\nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})])))} \\
&\leq \frac{1}{1 - \eta\|\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}, \boldsymbol{B}_{\ell_1})]\| - \eta\|\text{Err}\|} \qquad \text{Err} = \nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]
\end{aligned}$$

Using Lemma 3.6 w.p. at least $1 - 2Lp_1$

$$\|\text{Err}\|_F \leq Cp\sigma^{*2}_{\max}(8\epsilon\delta_{t-1} + 4.3G_B) \tag{11}$$

Using the bound on $\|\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}, \boldsymbol{B}_{\ell_1})]\|_F$ from (Abbasi & Vaswani, 2024, Lemma 4.5) and $\|\text{Err}\|_F$ from (11). We get

$$\eta(\|\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}, \boldsymbol{B}_{\ell_1})]\| + \|\text{Err}\|) \leq \eta p\sigma^{*2}_{\max}((2.5 + C8\epsilon)\delta_{t-1} + C4.3G_B).$$

Above we have used the fact that $\|\boldsymbol{M}\| \leq \|\boldsymbol{M}\|_F$.

Implies

$$\begin{aligned}
\frac{1}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})} &\leq \frac{1}{1 - \eta p\sigma^{*2}_{\max}((2.5 + C8\epsilon)\delta_{t-1} + C4.3G_B)} \\
&\leq 1 + \eta p\sigma^{*2}_{\max}((5 + C16\epsilon)\delta_{t-1} + C8.6G_B)
\end{aligned}$$

Above we have used for $0 < x < 1, 1/(1-x) \leq 1 + 2x$ assuming $\eta p\sigma^{*2}_{\max}((2.5 + C8\epsilon)\delta_{t-1} + C4.3G_B) < 1$

□

E. Proof of Lemma 3.3

Proof. $\boldsymbol{U}_{temp} = \boldsymbol{U}_{t-1} - \eta\nabla_{Kr}$.

Adding and subtracting $\eta\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]$, Note $\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})] = p(\boldsymbol{X}_{\ell_1} - \boldsymbol{X}^*_{\ell_1})\boldsymbol{B}^\top_{\ell_1}$

$$\begin{aligned}
\boldsymbol{U}_{temp} &= \boldsymbol{U}_{t-1} - \eta\mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})] - \eta(\nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]) \\
&= \boldsymbol{U}_{t-1} - \eta p(\boldsymbol{X}_{\ell_1} - \boldsymbol{X}^*_{\ell_1})\boldsymbol{B}^\top_{\ell_1} - \eta\text{Err}
\end{aligned}$$

Denote $\text{Err} = \nabla_{Kr} - \mathbb{E}[\nabla_{\ell_1}(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{\ell_1})]$. Multiplying both sides by $\mathcal{P} := \boldsymbol{I} - \boldsymbol{U}^*\boldsymbol{U}^{*\top}$, we get

$$\begin{aligned}
\mathcal{P}\boldsymbol{U}_{temp} &= \mathcal{P}\boldsymbol{U}_{t-1} - \eta p\mathcal{P}(\boldsymbol{X}_{\ell_1} - \boldsymbol{X}^*_{\ell_1})\boldsymbol{B}^\top_{\ell_1} - \eta\mathcal{P}\text{Err} \\
&= \mathcal{P}\boldsymbol{U}_{t-1} - \eta p\mathcal{P}\boldsymbol{U}\boldsymbol{B}_{\ell_1}\boldsymbol{B}^\top_{\ell_1} - \eta\mathcal{P}\text{Err} \\
&= \mathcal{P}\boldsymbol{U}_{t-1}(\boldsymbol{I}_r - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}^\top_{\ell_1}) - \eta\mathcal{P}\text{Err}
\end{aligned}$$

Taking Frobenius norm and using $\|M_1M_2\|_F \leq \|M_1\|_F\|M_2\|$ we get

$$\|\mathcal{P}\boldsymbol{U}_{temp}\|_F \leq \|\mathcal{P}\boldsymbol{U}_{t-1}\|_F\|I_r - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top\| + \eta\|\mathcal{P}\mathrm{Err}\|_F \tag{12}$$

Now $\boldsymbol{U}_{temp} \overset{\text{QR}}{=} \boldsymbol{U}_t\boldsymbol{R}^+$ and since $\|M_1M_2\|_F \leq \|M_1\|_F\|M_2\|$, this means that $\boldsymbol{SD}(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \|(I - \boldsymbol{U}^*\boldsymbol{U}^{*T})\boldsymbol{U}_{temp}\|_F\|(R^+)^{-1}\|$. Since $\|(R^+)^{-1}\| = 1/\sigma_{min}(R^+) = 1/\sigma_{min}(\boldsymbol{U}_{temp})$,

$$\|(R^+)^{-1}\| = \frac{1}{\sigma_{min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}$$

Combining the last two bounds gives.

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \frac{\|\boldsymbol{I}_r - \eta p\boldsymbol{B}_{\ell_1}\boldsymbol{B}_{\ell_1}^\top\|\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_{t-1}) + \eta\|\mathrm{Err}\|_F}{\sigma_{\min}(\boldsymbol{U}_{t-1} - \eta\nabla_{Kr})}$$

□

F. Why we cannot use projection step to guarantee incoherence

A natural question would be to use projection $\Pi_{\mathcal{U}}$ at center after each GD step to make rows of $\boldsymbol{U}_t$ incoherent. We cannot get a very useful bound on $\boldsymbol{SD}_F(\boldsymbol{U}^*, \Pi_{\mathcal{U}}(\boldsymbol{U}_t))$ which we explain next. By Lemmas 2.5 and 2.6 of (Chen, Chi, Fan, Ma, et al., 2021), for two $n\times r$ matrices with orthonormal columns, $\boldsymbol{U}_1, \boldsymbol{U}_2$, $\boldsymbol{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2) \leq \mathrm{argmin}_{\boldsymbol{Q}\ unitary}\|\boldsymbol{U}_1 - \boldsymbol{U}_2Q\|_F \leq \sqrt{2}\boldsymbol{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2)$.

Let us say you have a bound δ_t before projection step i.e., $\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \delta_t$. After the projection from you have $\boldsymbol{SD}_F(\Pi_{\mathcal{U}}(\boldsymbol{U}_t), \boldsymbol{U}^*) \leq \|\Pi_{\mathcal{U}}(\boldsymbol{U}_t) - \boldsymbol{U}^*Q\|_F$ for any $\boldsymbol{Q}$ $unitary$. Now the row norm clipping step can be interpreted as projecting its input onto a convex set, $\mathcal{U} := \{\tilde{\boldsymbol{U}} : \|\tilde{\boldsymbol{u}}^j\| \leq (1 - \frac{0.4}{\tilde{\kappa}^2})\|\boldsymbol{u}_{t-1}^j\| + 1.4\mu\sqrt{\frac{r}{n}}\}$, with the projection being in Frobenius norm. And projection onto convex sets is non-expansive, i.e., $\|\Pi_{\mathcal{U}}(\boldsymbol{U}_1) - \Pi_{\mathcal{U}}(\boldsymbol{U}_2)\|_F \leq \|\boldsymbol{U}_1 - \boldsymbol{U}_2\|_F$ (Nashed, 1968, eq (9),(10)),(Yi et al., 2016). Also, $\Pi_{\mathcal{U}}(\boldsymbol{U}^*\boldsymbol{Q}) = \boldsymbol{U}^*\boldsymbol{Q}$ for any $r\times r$ unitary matrix $\boldsymbol{Q}$ (since $\boldsymbol{U}^*$ as well as $\boldsymbol{U}^*$ times any unitary matrix belong to $\mathcal{U}$). Let $\boldsymbol{Q}_{*,t} := \mathrm{argmin}_{\boldsymbol{Q}\ unitary}\|\boldsymbol{U}_t - \boldsymbol{U}^*Q\|_F$ this then implies $\boldsymbol{SD}_F(\Pi_{\mathcal{U}}(\boldsymbol{U}_t), \boldsymbol{U}^*) \leq \|\Pi_{\mathcal{U}}(\boldsymbol{U}_t) - \boldsymbol{U}^*\boldsymbol{Q}_{*,t}\|_F = \|\Pi_{\mathcal{U}}(\boldsymbol{U}_t) - \Pi_{\mathcal{U}}(\boldsymbol{U}^*\boldsymbol{Q}_{*,t})\|_F \leq \|\boldsymbol{U}_t - \boldsymbol{U}^*\boldsymbol{Q}_{*,t}\|_F \leq \sqrt{2}\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*) \leq \sqrt{2}\delta_t$. So it introduces a factor of $\sqrt{2}$ after each projection, and hence for large T this will grow exponentially making sample complexity worse. Thats why we use filtering step.

G. Proof of Lemma 3.9

Proof. Denote the L_{byz} Byzantine vectors as $\{\boldsymbol{z}_k^B\}_{k=1}^{L_{byz}}$ and $L - L_{byz}$ *good* vectors as $\{\boldsymbol{z}_\ell\}_{\ell=1}^{L-L_{byz}}$. By $\ell \longrightarrow j$ we mean vector j is the neighbor of vector ℓ. For each index (*good* or Byzantine) i, we denote by $\delta_g(i)$ (resp. $\delta_b(i)$) the number of *good* (resp. Byzantine) indices j such that $i \longrightarrow j$. We have

$$\begin{aligned}\delta_g(i) + \delta_b(i) &= L - L_{byz} - 2\\ L - 2L_{byz} - 2 \leq \delta_g(i) &\leq L - L_{byz} - 2\\ \delta_b(i) &\leq L_{byz}.\end{aligned}$$

Let Kr is the index selected by Krum. Let ℓ_1 be the index of any *good* vector. Now we can write

$$\begin{aligned}\|\boldsymbol{z}_{Kr} - \tilde{\boldsymbol{z}}\| &= \|\boldsymbol{z}_{Kr} - \tilde{\boldsymbol{z}} \pm \frac{1}{\delta_g(Kr)}\sum_{Kr\longrightarrow \text{good } j}\boldsymbol{z}_j\|\\ &\leq \|\boldsymbol{z}_{Kr} - \frac{1}{\delta_g(Kr)}\sum_{Kr\longrightarrow \text{good } j}\boldsymbol{z}_j\| + \|\tilde{\boldsymbol{z}} - \frac{1}{\delta_g(Kr)}\sum_{Kr\longrightarrow \text{good } j}\boldsymbol{z}_j\|\end{aligned}$$

$$\leq \frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{ good } j} \|\boldsymbol{z}_{Kr} - \boldsymbol{z}_j\| + \frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{ good } j} \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_j\|$$
$$\leq \frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{ good } j} \|\boldsymbol{z}_{Kr} - \boldsymbol{z}_j\| + \max_{Kr \longrightarrow \text{ good } j} \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_j\| \tag{13}$$

Analyzing the first term. There are two possibilities 1) $Kr \in \mathcal{J}_{good}$, or 2) $Kr \in \mathcal{J}_{good}^{\complement}$ i.e.,

$$\frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{ good } j} \|\boldsymbol{z}_{Kr} - \boldsymbol{z}_j\| = \frac{1}{\delta_g(\ell)} \sum_{\ell \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell} - \boldsymbol{z}_j\| \mathbb{1}_{Kr=\ell \in \mathcal{J}_{good}} + \frac{1}{\delta_g(k)} \sum_{k \longrightarrow \text{ good } j} \|\boldsymbol{z}_k^B - \boldsymbol{z}_j\| \mathbb{1}_{Kr=k \in \mathcal{J}_{good}^{\complement}}$$

We will first analyze the case when $Kr = k \in \mathcal{J}_{good}^{\complement}$. Since Kr minimizes the score therefore there exist a *good* gradient ℓ' such that

$$\sum_{k \longrightarrow \text{ good } j} \|\boldsymbol{z}_k^B - \boldsymbol{z}_j\| + \sum_{k \longrightarrow \text{ byz } i} \|\boldsymbol{z}_k^B - \boldsymbol{z}_i^B\| \leq \sum_{\ell' \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_j\| + \sum_{\ell' \longrightarrow \text{ byz } i} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_i^B\|$$

Each ℓ' has $L - L_{byz} - 2$ neighbors, and $L_{byz} + 1$ non-neighbors. Thus there exists a *good* gradient $\zeta(\ell')$ which is farther from ℓ' than any of the neighbors of ℓ'. In particular, for each Byzantine index i such that $\ell' \longrightarrow$ byz i, $\|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_i^B\|^2 \leq \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_{\zeta(\ell')}\|^2$. Implies

$$\begin{aligned}
\sum_{k \longrightarrow \text{ good } j} \|\boldsymbol{z}_k^B - \boldsymbol{z}_j\| &\leq \sum_{k \longrightarrow \text{ good } j} \|\boldsymbol{z}_k^B - \boldsymbol{z}_j\| + \sum_{k \longrightarrow \text{ byz } i} \|\boldsymbol{z}_k^B - \boldsymbol{z}_i^B\| \\
&\leq \sum_{\ell' \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_j\| + \sum_{\ell' \longrightarrow \text{ byz } i} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_i^B\| \\
&\leq \sum_{\ell' \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_j\| + \delta_b(\ell') \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_{\zeta(\ell')}\| \\
&\leq \delta_g(\ell') \max_{\ell' \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_j\| + \delta_b(\ell') \|\boldsymbol{z}_{\ell'} - \boldsymbol{z}_{\zeta(\ell')}\|
\end{aligned}$$

Now bounding $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|$ for any $i, j \in \mathcal{J}_{good}$.

$$\begin{aligned}
\|\boldsymbol{z}_i - \boldsymbol{z}_j\| &= \|\boldsymbol{z}_i - \boldsymbol{z}_j \pm \tilde{\boldsymbol{z}}\| \\
&\leq \|\boldsymbol{z}_i - \tilde{\boldsymbol{z}}\| + \|\boldsymbol{z}_j - \tilde{\boldsymbol{z}}\| \\
&\leq 2\epsilon \|\tilde{\boldsymbol{z}}\|
\end{aligned}$$

w.p. at least $1 - 2p$.

Using union bound w.p. at least $1 - 2(L - L_{byz} - 2)p$

$$\begin{aligned}
\sum_{k \longrightarrow \text{ good } j} \|\boldsymbol{z}_k^B - \boldsymbol{z}_j\| &\leq (\delta_g(\ell') + \delta_b(\ell')) 2\epsilon \|\tilde{\boldsymbol{z}}\| \\
&\leq (L - L_{byz} - 2) 2\epsilon \|\tilde{\boldsymbol{z}}\|.
\end{aligned}$$

For $Kr = \ell \in \mathcal{J}_{good}$

$$\begin{aligned}
\frac{1}{\delta_g(\ell)} \sum_{\ell \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell} - \boldsymbol{z}_j\| \mathbb{1}_{Kr=\ell \in \mathcal{J}_{good}} &\leq \max_{\ell \longrightarrow \text{ good } j} \|\boldsymbol{z}_{\ell} - \boldsymbol{z}_j\| \\
&\leq 2\epsilon \|\tilde{\boldsymbol{z}}\|
\end{aligned}$$

Combining $Kr \in \mathcal{J}_{good}$ and $Kr \in \mathcal{J}_{good}^{\complement}$ we get

$$\begin{aligned}\frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{good } j} \|\boldsymbol{z}_{Kr} - \boldsymbol{z}_j\| &\leq \max\left(2\epsilon\|\tilde{\boldsymbol{z}}\|, \frac{L - L_{byz} - 2}{\delta_g(k)} 2\epsilon\|\tilde{\boldsymbol{z}}\|\right) \\ &\leq \max\left(1, \frac{L - L_{byz} - 2}{L - 2L_{byz} - 2}\right) 2\epsilon\|\tilde{\boldsymbol{z}}\|\end{aligned}$$

This then implies

$$\begin{aligned}\|\boldsymbol{z}_{Kr} - \tilde{\boldsymbol{z}}\| &\leq \frac{1}{\delta_g(Kr)} \sum_{Kr \longrightarrow \text{good } j} \|\boldsymbol{z}_{Kr} - \boldsymbol{z}_j\| + \max_{Kr \longrightarrow \text{good } j} \|\tilde{\boldsymbol{z}} - \boldsymbol{z}_j\| \\ &\leq \max\left(1, \frac{L - L_{byz} - 2}{L - 2L_{byz} - 2}\right) 2\epsilon\|\tilde{\boldsymbol{z}}\| + \epsilon\|\tilde{\boldsymbol{z}}\| \\ &= \max\left(2, 1 + \frac{L - L_{byz} - 2}{L - 2L_{byz} - 2}\right) 2\epsilon\|\tilde{\boldsymbol{z}}\| \\ &\leq \left(2 + \frac{L - L_{byz} - 2}{L - 2L_{byz} - 2}\right) 2\epsilon\|\tilde{\boldsymbol{z}}\| \\ &\leq 10\epsilon\|\tilde{\boldsymbol{z}}\| \end{aligned} \tag{14}$$

w.p. at least $1 - 2(L - L_{byz})p = 1 - 2L(1 - \tau)p$. For $\tau = \frac{L_{byz}}{L} < 0.4$, $2 + \frac{L - L_{byz} - 2}{L - 2L_{byz} - 2} \lesssim 5$.

□