Communication-Efficient Federated Low-Rank Matrix Completion: Convergence Guarantees and Gradient Minimization

Chris Junchi Li$^\diamond$

Department of Electrical Engineering and Computer Sciences$^\diamond$
University of California, Berkeley

October 9, 2024

Abstract

In this paper, we address the low-rank matrix completion (LRMC) problem, a fundamental challenge in data science and machine learning. We propose a Gradient Descent-based solution, termed Alternating Gradient Minimization (AltGDMin), specifically designed for federated learning settings. The algorithm efficiently recovers an $n \times q$ rank-r matrix $X^\star$ from a subset of its entries, particularly when $r \ll \min(n, q)$, while minimizing communication between distributed nodes and a central server. Our theoretical guarantees include iteration and sample complexity bounds, positioning AltGDMin as one of the most communication-efficient and fastest methods available. Additionally, we extend the analysis to noisy LRMC, establishing robust convergence guarantees. Numerical experiments validate that AltGDMin outperforms existing centralized and federated approaches, highlighting its potential to recover missing entries by leveraging low-dimensional data structure.

Keywords: Low-rank matrix completion, Federated learning, Alternating gradient minimization, Sample complexity, Communication-efficient algorithms.

1 Introduction

The problem of low-rank matrix completion (LRMC) has become a critical task in data science and machine learning, with applications ranging from collaborative filtering to recommendation systems and compressive sensing [Achlioptas & McSherry, 2007, Keshavan et al., 2010b, Gross, 2011]. In LRMC, only a small subset of entries from a large matrix are observed, and the goal is to accurately recover the full matrix, leveraging its low-rank structure. This problem is inherently ill-posed, as the number of observed entries is significantly smaller than the total number of unknowns. Nevertheless, by exploiting the low-dimensional structure of the data, accurate recovery of missing entries is feasible [Candès & Recht, 2009, Candès & Tao, 2010].

Recent advances have highlighted the challenges of LRMC in federated learning settings, where data is distributed across multiple nodes, and direct communication with a central server is restricted to preserve privacy and reduce bandwidth usage. In such scenarios, algorithms must be designed to be both communication-efficient and privacy-preserving while maintaining high accuracy in recovering the low-rank matrix. This introduces additional complexity, as nodes can only share limited information with the central server without revealing sensitive data [Chen & Chi, 2018, Davenport & Romberg, 2016].

To address these challenges, we introduce a novel algorithm, Alternating Gradient Minimization (AltGDMin), designed specifically for LRMC in federated settings. AltGDMin iteratively minimizes a loss function by sharing only partial gradients between distributed nodes and the central server, significantly reducing communication overhead. The algorithm is not only efficient in federated

settings but also provides robust guarantees for recovering the low-rank matrix under both clean and noisy conditions. Our theoretical analysis shows that AltGDMin achieves low iteration and sample complexity, making it one of the fastest and most communication-efficient methods available for LRMC.

In this work, we develop and analyze a fast and communication-efficient solution for efficiently solving the low rank matrix completion (LRMC) problem in a federated setting. LRMC involves recovering/learning a low rank (LR) matrix from a small number of its observed entries. Our proposed algorithm, and the sample and iteration complexity guarantees that we prove for it, are also useful in centralized settings. Federation means that (i) different subsets of the data are acquired at different distributed nodes; and (ii) all nodes can only communicate with the central node or "center". Communication-efficiency is a key concern with all distributed algorithms, including federated ones. Privacy of the data is another concern in a federated setting. In this work, "privacy" means the following: the nodes' raw data (here observed matrix entries) cannot be shared with the center; and the algorithm should be such that the center cannot recover any of the columns of the LR matrix.

1.1 Problem Setup and Notation

LRMC involves recovering a rank-r matrix $\boldsymbol{X}^\star \in \mathbb{R}^{n \times q}$, where $r \ll \min(n, q)$, from a subset of its entries. Entry j of column k, denoted $\boldsymbol{X}^\star_{jk}$, is observed, independently of all other observations, with probability p. Let $\xi_{jk} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$ for $j \in [n], k \in [q]$. Then, the set of observed entries, denoted by Ω, is

$$\Omega := \{(j, k) : \xi_{jk} = 1\}$$

By setting the unobserved entries to zero, the observed data matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times q}$ can be defined as

$$\boldsymbol{Y}_{jk} := \begin{cases} \boldsymbol{X}^\star_{jk} & \text{if } (j, k) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \quad \text{or, equivalently,} \quad \boldsymbol{Y} := \boldsymbol{X}^\star_\Omega \tag{1}$$

Here and below, $\boldsymbol{M}_\Omega$ refers to the matrix $\boldsymbol{M}$ with all entries whose indices are not in the set Ω are zeroed out; while the rest of the entries remain unchanged.

Notation For any matrix $\boldsymbol{M}$, $\boldsymbol{m}_k$ denotes its k-th column while $\boldsymbol{m}^j$ denotes its j-th row transposed (so it is a column vector). $(\cdot)^\intercal$ denotes the matrix/vector transpose. We use $\boldsymbol{I}$ to denote the identity matrix and $\boldsymbol{e}_k$ to denote its k-th column (this is a 1-0 vector with 1 at the k-th location and zero everywhere else). $\|\cdot\|$ denotes either the vector ℓ_2 or the induced ℓ_2 norm of a matrix ($\|\boldsymbol{M}\| := \max_{\boldsymbol{z}:\|\boldsymbol{z}\|=1} \|\boldsymbol{M}\boldsymbol{z}\| = \sigma_{\max}(\boldsymbol{M})$), while $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We use $vec(\boldsymbol{M})$ to denote a vectorized version of matrix $\boldsymbol{M}$; thus $\|\boldsymbol{M}\|_F = \|vec(\boldsymbol{M})\|$. We use the standard notation $[q] := \{1, 2, \ldots, q\}$.

For a tall matrix $\boldsymbol{M}$, $\boldsymbol{M}^\dagger \triangleq (\boldsymbol{M}^\intercal \boldsymbol{M})^{-1} \boldsymbol{M}^\intercal$ denotes the Moore-Penrose pseudo-inverse. Also, $\text{QR}(\boldsymbol{M})$ maps the matrix $\boldsymbol{M} \in \mathbb{R}^{n \times r}$ to $\boldsymbol{Q} \in \mathbb{R}^{n \times r}$ such that $\boldsymbol{M} = \boldsymbol{Q}\boldsymbol{R}$ is the QR decomposition of $\boldsymbol{M}$ (columns of $\boldsymbol{Q}$ are orthornormal). For matrices $\boldsymbol{U}_1, \boldsymbol{U}_2$ with orthonormal columns, the two commonly used measures of subspace distance (SD) between their column spans are $\text{SD}_2(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1 \boldsymbol{U}_1^\intercal) \boldsymbol{U}_2\|$ and $\text{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1 \boldsymbol{U}_1^\intercal) \boldsymbol{U}_2\|_F$. We use the Frobenius norm SD in most of our analysis in this paper because it allows us to get a better sample complexity guarantee.

We use $\boldsymbol{M} = \text{diag}(m_k, k \in [q])$ to denote a diagonal $q \times q$ matrix with scalar entries m_k and $\boldsymbol{M} = \text{blockdiag}(\boldsymbol{M}_k, k \in [q])$ to denote a block-diagonal matrix with block diagonal entries $\boldsymbol{M}_k$.

We use $\Omega_k := \{j \in [n] \mid \xi_{jk} = 1\}$ to denote the set of indices of the observed entries in column k and we define a diagonal 1-0 matrix $\boldsymbol{S}_k \in \mathbb{R}^{n \times n}$ as

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

Thus, $\boldsymbol{y}_k = \boldsymbol{S}_k \boldsymbol{x}_k^\star$ for all $k \in [q]$. *We reuse the letters c, C to denote different numerical constants in each use with the convention that $c < 1$ and $C \geq 1$.*

Let $\boldsymbol{X}^\star \overset{\mathrm{SVD}}{=} \boldsymbol{U}^\star (\boldsymbol{\Sigma}^*) \boldsymbol{V}^* := \boldsymbol{U}^\star \boldsymbol{B}^\star$, where $\boldsymbol{U}^\star \in \mathbb{R}^{n \times r}$ and has orthonormal columns and $\boldsymbol{V}^\star \in \mathbb{R}^{r \times q}$ with orthonormal rows. We use $\kappa = \sigma_{\max}^* / \sigma_{\min}^*$ to denote the condition number of the diagonal $r \times r$ matrix $\boldsymbol{\Sigma}^*$. Here $\sigma_{\max}^*, \sigma_{\min}^*$ to denote its largest, smallest singular values. Also, we let $\boldsymbol{B}^\star := \boldsymbol{\Sigma}^* \boldsymbol{V}^*$ so that $\boldsymbol{X}^\star = \boldsymbol{U}^\star \boldsymbol{B}^\star$.

Assumption As in all past works on LRMC, e.g. [Candes & Recht, 2008, Netrapalli et al., 2013], we need the following assumption on the singular vectors of $\boldsymbol{X}^\star$; this is a way to guarantee that the rows and columns of $\boldsymbol{X}^\star$ are dense (non-sparse). This and the LR assumption help ensure that one can correctly interpolate (fill in) the missing entries even with observing only a few entries of each row or column.

Assumption 1 (μ-incoherence of singular vectors of $\boldsymbol{X}^\star$). *Assume row norm bounds on $\boldsymbol{U}^\star$: $\max_{j \in [n]} \|\boldsymbol{u}^{*j}\| \leq \mu \sqrt{r/n}$, and column norm bounds on $\boldsymbol{V}^*$: $\max_{k \in [q]} \|\boldsymbol{v}_k^*\| \leq \mu \sqrt{r/q}$. Since $\boldsymbol{B}^\star = \boldsymbol{\Sigma}^* \boldsymbol{V}^*$, this implies that $\|\boldsymbol{b}_k^\star\| \leq \mu \sqrt{r/q} \sigma_{\max}^*$.*

We should point out that our proof for AltGDmin correctness only uses the incoherence of $\boldsymbol{U}^\star$ and of $\boldsymbol{B}^\star$. This is a slightly weaker requirement than requiring incoherence of $\boldsymbol{U}^\star, \boldsymbol{V}^\star$.

Federation We assume that there are a total of γ nodes, with $\gamma \leq q$. Each node has access to a different subset of the columns of the observed data matrix $\boldsymbol{Y}$. We use $\boldsymbol{S}_\ell$ to denote the subset of columns of $\boldsymbol{Y}$ available at node ℓ. The sets $\boldsymbol{S}_\ell$ form a partition of $[q]$, i.e., they are mutually disjoint and $\cup_{\ell=1}^{\gamma} \boldsymbol{S}_\ell = [q]$. To keep notation simple, we assume q is a multiple of γ and $|\boldsymbol{S}_\ell| = q/\gamma$. All nodes can only communicate with a central node or "center". In a federated setting, the two desirable properties are communication-efficiency and "privacy". *In this work, "privacy" means the following. The nodes' raw data cannot be shared with the center and the center should not be able reconstruct the matrix $\boldsymbol{X}^\star$.*

Our discussion of complexities assumes $\gamma \ll q$ and treats γ as a numerical constant. Thus order $|\Omega|/\gamma$ is equal to order $|\Omega|$ with $|\Omega| \geq (n+q)r$ (the number of samples needs to be larger than the number of unknowns in rank r matrix).

1.2 Related Work

Starting with the seminal work of [Fazel, 2002, Candes & Recht, 2008] which introduced a nuclear norm based convex relaxation, the LRMC problem has been extensively studied in the last decade and a half [Fazel, 2002, Candes & Recht, 2008, Keshavan et al., 2010a, Netrapalli et al., 2013, Sun & Luo, 2016, Cherapanamjeri et al., 2016, Ma et al., 2018, Hardt & Wootters, 2014, Zheng & Lafferty, 2016, Yi et al., 2016]. Two classes of algorithms feature prominently in this literature - solutions to convex relaxations and direct iterative algorithms. The former [Fazel, 2002, Candes & Recht, 2008] are slow: the required number of iterations for ϵ accuracy (iteration complexity) grows as $1/\sqrt{\epsilon}$ [Netrapalli et al., 2013]. The first provably accurate iterative solution was the Alternating Minimization (AltMin) algorithm with a spectral initialization [Netrapalli et al., 2013, Keshavan et al., 2010a]. AltMin was

shown to converge geometrically (iteration complexity order $\log(1/\epsilon)$) with a sample complexity of $\kappa^4 \mu^2 nr^{4.5} \log(1/\epsilon)$ in [Netrapalli et al., 2013]. Subsequent work [Hardt, 2014] considered a modified version of AltMin and improved its sample complexity to order $\kappa^2 \mu^2 nr^{2.5} \log(1/\epsilon)$. AltMin factorizes the unknown LR matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{UB}$. After initializing $\boldsymbol{U}$, it alternately updates $\boldsymbol{B}$ and $\boldsymbol{U}$ using minimization for one, keeping the other fixed. Later works proposed two gradient descent (GD) based algorithms - Projected GD (ProjGD) [Jain & Netrapalli, 2015, Cherapanamjeri et al., 2016] and Factorized GD (FactGD) [Yi et al., 2016, Zheng & Lafferty, 2016] - that reduced the sample complexity dependence on r to r^2. ProjGD involves GD, followed by projection onto the space of rank r matrices (this is done by singular value decomposition (SVD)) after each GD iteration. FactGD factorizes the unknown LR matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{UB}$, where $\boldsymbol{U}, \boldsymbol{B}$ have r columns and rows respectively, and updates both by GD as follows. After initialization, at each iteration, it updates $\boldsymbol{U}$ and $\boldsymbol{B}$ by one GD step for the cost function $f(\boldsymbol{U}, \boldsymbol{B}) + c_1 \|\boldsymbol{U}^\mathsf{T}\boldsymbol{U} - \boldsymbol{BB}^\mathsf{T}\|_F^2$, followed by projecting each of them onto the set of matrices with incoherent rows and columns respectively. Denote the former (incoherent rows) set by $\mathcal{U}$. Initialization involves SVD on $\boldsymbol{Y}$ followed by a similar projection onto $\mathcal{U}$. The second term of the FactGD cost function is a norm balancing term that ensures the norm of $\boldsymbol{U}$ does not keep increasing with iterations while that of $\boldsymbol{B}$ decreases (or vice versa). The best ProjGD guarantee needs a sample complexity of $\mu^4 nr^2 \log^2 n \log^2 1/\epsilon$ while FactGD reduces this even further to $\kappa^4 \mu^2 nr^2 \log^2 n$. ProjGD converges geometrically with a constant GD step size, while FactGD needs the GD step size to be of order $1/r$. Consequently, its iteration complexity is worse than that of ProjGD or AltMin by a factor of r. However, in terms of per-iteration time cost, ProjGD is significantly slower. Consequently, overall in terms of total cost, it is slower. Numerically, it is much slower than all other methods, because each ProjGD iteration involves a rank r SVD step (constants per iteration are much larger). Some of this discussion treats κ, μ as numerical constants.

To our best knowledge, *there is no existing work on provably accurate federated LRMC*. Federation requires a communication-efficient and private solution, with "private" as defined above.

In tangentially related work [Mackey et al., 2015, Teflioudi et al., 2012], distributed-computing solutions to LRMC are studied. These do not consider the federated setting, instead these assume that all data is observed centrally and then is distributed to different nodes to parallelize the computation; and these develop an approximate solution to the convex relaxation which is known to be very slow. Fully decentralized LRMC has been studied in [Ling et al., 2012, Lin & Ling, 2015, Mardani et al., 2013]. These methods are much slower than GD. Other tangentially related works include [Li et al., 2021, Anelli et al., 2022, He et al., 2019], all these consider differential privacy or attack resilience.

The Alternating GD and Minimization (AltGDmin) algorithm was introduced in our past work [Nayer & Vaswani, 2023b, Nayer & Vaswani, 2023a] as a fast solution to the LR column-wise compressive sensing (LRCS) problem. LRCS involves recovering $\boldsymbol{X}^\star$ from $\boldsymbol{y}_k := \boldsymbol{A}_k \boldsymbol{x}_k^\star$ when $\boldsymbol{A}_k$'s are $m \times n$ i.i.d. random Gaussian matrices (each entry of each $\boldsymbol{A}_k$ is i.i.d. standard Gaussian), and the right singular vectors of $\boldsymbol{X}^\star$ satisfy the incoherence assumption. The overall idea of AltGDmin is as follows [Nayer & Vaswani, 2023b]. Suppose the goal is to solve $\min_{\boldsymbol{Z}} f(\boldsymbol{Z})$. We split the unknown variable $\boldsymbol{Z}$ into two parts, $\boldsymbol{Z} = \{\boldsymbol{Z}_a, \boldsymbol{Z}_b\}$. After initializing $\boldsymbol{Z}_a$, we alternatively update $\boldsymbol{Z}_b, \boldsymbol{Z}_a$ using minimization for $\boldsymbol{Z}_b$ (keeping $\boldsymbol{Z}_a$ fixed at its previous value) and GD for $\boldsymbol{Z}_a$ (keeping $\boldsymbol{Z}_b$ fixed at its previous value). We choose the split-up so that $\boldsymbol{Z}_b$ is the set of variables for which the minimization can be "decoupled", i.e., subsets of $\boldsymbol{Z}_b$ are functions of only a subsets of the data matrix $\boldsymbol{Y}$. AltGDmin is an efficient solution only for problems where this is possible to do. As will be clear later, this is what helps ensure both the communication, and computation, efficiency.

1.3 Contributions and Novelty

We develop and analyze the AltGDmin algorithm for solving the LRMC problem. The design of AltGDmin is motivated by a federated setting. To our best knowledge, our work provides the first theoretical guarantees for solving LRMC in a federated setting; see Theorem 2.2. The sample and iteration complexity bounds that we prove are applicable in centralized settings as well. In addition, we also prove two important corollaries. (a) We provide a guarantee for AltGDmin for solving the noisy LRMC problem. (b) We also show how our lemmas can be used to provide an improved sample complexity guarantee for AltMin which is the fastest centralized solution for LRMC.

Using our results we can argue that, in a federated setting, AltGDmin is the most communication-efficient solution. It is also one of the two fastest private solutions and has the second smallest sample complexity. Our new corollary for AltMin (and smooth AltMin) proves that it also has the same sample complexity as that of AltGDmin. This discussion treats κ, μ and γ (number of nodes) as numerical constants. We discuss these points in more detail in Sec. 3 and in Table 1.

Proof Novelty   While AltGDmin has been studied for solving the LRCS problem, there are some important differences between the two problems that necessitate new analysis approaches. LRMC samples are row and column-wise local, while LRCS samples are global for each column. Because of this, LRMC requires assuming left and right singular vectors' incoherence and proving the same for each new iterate of U and B. LRCS needed this only for B. We need a new approach to show the incoherence of U at each iteration since the update of U in AltGDmin is different from that in all existing solutions to LRMC. FactGD for LRMC [Yi et al., 2016] projects U (and B) onto the space of row (column) incoherent matrices after each GD step; this automatically ensures incoherence after each update. AltMin [Netrapalli et al., 2013] updates U by solving an LS problem (and not by GD), so the incoherence proof there is different too. ProjGD [Cherapanamjeri et al., 2016] implemented projected GD for X itself (does not factorize it). There are other important proof differences too. We explain these in Sec. 3.2.

1.4 Organization

We develop the AltGDmin algorithm and state and explain our main theoretical guarantee, Theorem 2.2, for it next in Sec. 2. We provide a detailed comparison of this guarantee with those from existing works in Sec. 3. This section also summarizes the novel proof ideas and gives a proof outline. Theorem 2.2 is proved in Sec.4. The lemmas used in this proof are proved in Sec. 5. The corollary for noisy LRMC is provided in Sec. 6. We use our lemmas to prove an improved guarantee for AltMin and Smooth AltMin in Sec. 7. We conclude in Sec. 8.

2 Alternating GD and Minimization (AltGDMin) Algorithm and Guarantees

2.1 AltGDmin Algorithm

The goal is to minimize the following squared loss cost function

$$\min_{\check{\boldsymbol{B}}, \check{\boldsymbol{U}} \,:\, \check{\boldsymbol{U}}^\intercal \check{\boldsymbol{U}} = \boldsymbol{I}} f(\check{\boldsymbol{U}}, \check{\boldsymbol{B}}), \ f(\check{\boldsymbol{U}}, \check{\boldsymbol{B}}) := \|(\boldsymbol{Y} - \check{\boldsymbol{U}}\check{\boldsymbol{B}})_\Omega\|_F^2 \tag{2}$$

We impose the orthornormal columns constraint on $\check{U}$ as one way to ensure that the norm of U does not keep increasing or decreasing continuously with algorithm iterations, while that of B decreases or increases.

AltGDmin proceeds as follows. We initialize U as explained below; this approach is adapted from that in FactGD [Yi et al., 2016]. After the initialization, different from AltMin [Netrapalli et al., 2013], which used alternating exact minimization for both U and B, and different from FactGD [Yi et al., 2016], which used GD for both U and B, AltGDmin alternates between exact minimization over B and a single GD step for U. The GD step is followed by an orthonormalization (QR) step. The above simple changes help ensure that the AltGDmin iteration complexity is better than that of FactGD and comparable to that of AltMin. The use of exact minimization for one of the variables, along with not using the projection onto incoherent matrices step (replacing it by QR), helps ensure that AltGDmin provably converges with a nearly constant step size. Because of this, the AltGDmin iteration complexity is better than that of FactGD by a factor of r.

Initialization As in most previous work [Netrapalli et al., 2013, Yi et al., 2016], the first step of our initialization is to compute the top r singular vectors of Y; denote the matrix formed by these singular vectors by $U^{(00)}$. This is followed by a step to make $U^{(00)}$ incoherent. We borrow this step from [Yi et al., 2016]. It involves projecting $U^{(00)}$ onto the space of row incoherent matrices,

$$\mathcal{U} := \{\check{U} : \|\check{u}^j\| \leq \mu\sqrt{r/n}\}$$

i.e., computing $\Pi_{\mathcal{U}}(U^{(00)}) = \min_{\check{U} \in \mathcal{U}} \|\check{U} - U^{(00)}\|_F$. It is easy to see that this projection can be obtained in closed form as follows:

$$[\Pi_{\mathcal{U}}(M)]^j = m^j \cdot \min\left(1, \frac{\mu\sqrt{r/n}}{\|m^j\|}\right), \quad \text{for all } j \in [n]$$

In words, if a row of M has ℓ_2 norm that is more than the threshold $\mu\sqrt{r/n}$, then one renormalizes the row so that its norm equals the threshold. If the norm is less than this threshold, then we do not change it. Clearly this is an order nr time operation. The last step is computing a QR decomposition of $\Pi_{\mathcal{U}}(U^{(00)})$. This needs time of order nr^2.

AltGDmin iterations We update B and U as follows for a total of T iterations. The minimization over B is a decoupled least squares (LS) problem since $f(\check{U}, \check{B})$ decouples as $f(\check{U}, \check{B}) = \sum_{k \in [q]} \|y - S_k \check{U}\check{b}_k\|^2$. We update b_k as

$$b_k = \arg\min_{\check{b}} \|y - S_k U\check{b}\|^2 = (S_k U)^\dagger y_k, \quad \text{for all } k \in [q]$$

Let $U_k := S_k U$, then $b_k := U_k^\dagger y_k$. Recall that $M^\dagger := (M^\intercal M)^{-1}M^\intercal$. We emphasise here that we write things as above for notational ease in both explaining the idea and the proof. The time complexity for computing b_k depends only on the sub-matrix of U_k with nonzero rows. This is of size $|\Omega_k| \times r$. The time cost of computing b_k is thus order $|\Omega_k|r^2$ for a given k. Hence, the total cost for all nodes is $\sum_k |\Omega_k|r^2 = |\Omega|r^2$.

We update U by one GD step followed by orthnormalization using QR, i.e.,

$$\widetilde{U}^+ = U - \eta\nabla_U f(U, B)), \text{ and } U^+ = \text{QR}(\widetilde{U}^+)$$

6

Algorithm 1 AltGDMin

Require: partial observations $\boldsymbol{Y}$, rank r, step size η, and number of iterations T

1: Partition $\boldsymbol{Y}$ into $2T+1$ subsets $\boldsymbol{Y}_{\Omega^{(0)}}, \cdots, \boldsymbol{Y}_{\Omega^{(2T)}}$
2: $\boldsymbol{U}^{(00)} \leftarrow$ top r left-singular vectors of $\boldsymbol{Y}_{\Omega^{(0)}}$
3: $\boldsymbol{M}^{(0)} \leftarrow \Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)})$
4: $\boldsymbol{U}^{(0)} \leftarrow \mathrm{QR}(\boldsymbol{M}^{(0)}$
5: **for** $t \in 1 \cdots T$ **do**
6: $\quad \boldsymbol{b}_k^{(t)} \leftarrow (\boldsymbol{U}_k^{(t-1)})^{\dagger} \boldsymbol{y}_{\Omega_k^{(t)}}$ for all $k \in [q]$
7: $\quad \widetilde{\boldsymbol{U}}^{(t)} \leftarrow \boldsymbol{U}^{(t-1)} - \eta(\boldsymbol{U}^{(t-1)}\boldsymbol{B}^{(t)} - \boldsymbol{Y})_{\Omega^{(T+t)}}\boldsymbol{B}^{(t)\mathsf{T}}$
8: $\quad \boldsymbol{U}^{(t)} \leftarrow \mathrm{QR}(\widetilde{\boldsymbol{U}}^{(t)})$, i.e. $\widetilde{\boldsymbol{U}}^{(t)} \overset{\mathrm{QR}}{=} \boldsymbol{U}^{(t)}\boldsymbol{R}(t)$
9: **end for**
10: **Return** $\boldsymbol{U}, \boldsymbol{B}$

The gradient is

$$\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}) = 2((\boldsymbol{U}\boldsymbol{B})_{\Omega} - \boldsymbol{Y})\boldsymbol{B}^{\mathsf{T}} = 2\sum_{k=1}^{q}(\boldsymbol{S}_k \boldsymbol{U}\boldsymbol{b}_k - \boldsymbol{y}_k)\boldsymbol{b}_k^{\mathsf{T}}$$

For the gradient computation, the time cost is $\sum_k |\Omega_k| r = |\Omega| r$. The QR step needs time of order nr^2.

We summarize the complete algorithm in Algorithm 1. Sample splitting (line 1) is assumed, as is common in most structured data recovery literature, e.g., [Netrapalli et al., 2013, Hardt, 2014, Jain & Netrapalli, 2015, Cherapanamjeri et al., 2016]. In fact, as we discuss in Sec. 3, sample-splitting is assumed for obtaining provable guarantees for all iterative solutions for LRMC (and those for LRCS) in which one or both of the alternating steps is a minimization step.

Federated AltGDmin In a federated setting, at each algorithm iteration, t, each node $\ell = 1, 2, \ldots, \gamma$ performs two operations i) updating $\boldsymbol{b}_k$ by the LS solution, for all $k \in \boldsymbol{S}_{\ell}$; and ii) computation of the partial gradient $\sum_{k \in \boldsymbol{S}_{\ell}}[\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B})]_k = \sum_{k \in \boldsymbol{S}_{\ell}}(\boldsymbol{U}_k \boldsymbol{b}_k - \boldsymbol{y})_{\Omega_k} \boldsymbol{b}_k^{\mathsf{T}}$. Only the $n \times r$ partial gradient needs to be sent to the center. The center sums the received partial gradients, implements the GD step, and computes the QR decomposition, and broadcasts the updated $\boldsymbol{U}^+$ to all the nodes. This is used by the nodes in the next iteration. The communication cost from nodes to center is equal to the number of nonzero entries in $\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B})$; this is $r \cdot \max(n, \sum_{k \in \boldsymbol{S}_{\ell}} |\Omega_k|) = r \cdot \min(n, (|\Omega|/\gamma)) = nr$ since we assume γ is a numerical constant and $|\Omega| \geq (n+q)r$ is a necessary lower bound for any approach to work. The center to nodes communication cost is also nr. The computation cost is as explained above with $|\Omega|$ replaced by $|\Omega|/\gamma$.

The initialization can be federated using the power method applied to $\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{T}}$. This costs order $\sum_{k \in \boldsymbol{S}_{\ell}} |\Omega_k| r$ per power method iteration. The upstream (node to center) per-node communication cost is $\min(n, \sum_{k \in \boldsymbol{S}_{\ell}} |\Omega_k|) r = nr$ and the downstream cost is also nr. The power method converges linearly and thus, for ϵ_{svd} accuracy, $\log(1/\epsilon_{svd})$ iterations are required. We prove that we need $\epsilon_{svd} = c/\kappa^2$.

2.2 AltGDmin Guarantee

We can prove the following for AltGDmin. Our theorem statements and proofs assume $n \leq q$ for simplicity. Recall from above that the per-iteration per-node computation and communication complexities of AltGDmin are $\max(nr^2, (|\Omega|/\gamma)r^2)$ and $\max(nr, \min(n, (|\Omega|/\gamma))r) = nr$. Here

per-node takes the maximum of the node or center complexities. We use this and the iteration complexity (expression for T) derived in the result below to provide an expression for its total per-node computation and communication complexities.

Let $p_{tot} := 2Tp$. Our proof of Theorem 2.2 given next obtains a lower bound on the required p in each algorithm iteration.

Theorem 2.2. *Pick an $\epsilon < 1$. Assume that Assumption 1 holds and that, entries of $\mathbf{X}^\star$ are observed independently with probability p_{tot} satisfying $nqp_{tot} > C\kappa^6\mu^2 qr^2 \log q \log(1/\epsilon)$. Set $\eta = 0.5/(p\sigma^{*2}_{\max})$ and $T = C\kappa^2 \log(1/\epsilon)$. Then, with probability (w.p.) at least $1 - 4T/n^3$,*

$$\mathrm{SD}_F(\mathbf{U}^{(T)}, \mathbf{U}^\star) \leq \epsilon \ \text{and} \ \|\mathbf{X}^{(T)} - \mathbf{X}^\star\|_F \leq \epsilon \|\mathbf{X}^\star\| \tag{3}$$

(recall that $\mathbf{X}^{(T)} = \mathbf{U}^{(T)}\mathbf{B}^{(T)}$).

The total per-node computation complexity of federated AltGDmin is $C\kappa^2 \log(1/\epsilon) \cdot \max(n, |\Omega|/\gamma)r^2$ and its total per-node communication complexity is $C\kappa^2 \log(1/\epsilon) \cdot nr$.

The proof is given in Sec. 4 and 5. Before this, we discuss this result and the novel ideas used in our proof in Sec. 3.

Observe that $nq \cdot p_{tot} = \mathbb{E}[|\Omega|]$, i.e., it is the expected value of the sample complexity. We often just use the phrase "sample complexity" when referring to it in our writing. Treating κ, μ as numerical constants, the above result says that as long as we observe order $nr^2 \log q \log(1/\epsilon)$ matrix entries, and we set the GD step size, η, and the total number of iterations, T, as stated, then with high probability (whp), we can fill in the rest of the entries accurately: the normalized Frobenius norm of the error in this estimation is at most ϵ. Also, we can estimate the column span of $\mathbf{X}^\star$ with ϵ accuracy. The number of iterations needed is order $\log(1/\epsilon)$. We can achieve any ϵ accuracy as long as we increase the sample and iteration complexity accordingly.

Remark. We state Theorem 2.2 for one value of the step size η because this makes some of our proof arguments notationally less messy. However, our proof will go through for any step size $\eta \leq 0.5/(p\sigma^{*}_{\max}{}^2)$. We can actually show the following: let $\eta = c_\eta/(p\sigma^{*}_{\max}{}^2)$ for a $c_\eta \leq 0.5$. Then, if at each iteration, $nq \cdot p > C\kappa^6\mu^2 qr^2 \log q$, then with probability (w.p.) at least $1 - 4/n^3$,

$$\mathrm{SD}_F(\mathbf{U}^+, \mathbf{U}^\star) \leq (1 - \frac{cc_\eta}{\kappa^2})\mathrm{SD}_F(\mathbf{U}, \mathbf{U}^\star)$$

3 Discussion and Proof Main Ideas/Novelty

3.1 Discussion

The average (expected value of) the required sample complexity is nqp. For this discussion, we assume $n \approx q$, $\kappa \approx \mu$, $\max(\log(1/\epsilon), r) = \log(1/\epsilon)$. Our guarantee given in Theorem 2.2 shows that the AltGDmin sample complexity is $\kappa^6\mu^2 nr^2 \log n \log(1/\epsilon)$. This is worse than that of FactGD by a factor of $\kappa^2 \log(1/\epsilon)$. However, the iteration complexity of AltGDmin is $T = \kappa^2 \log(1/\epsilon)$ while that of FactGD is $T = \kappa\mu r \log(1/\epsilon)$. Assuming $\kappa \approx \mu$, this means the AltGDmin iteration complexity is better by a factor of r. This is also demonstrated numerically in Sec. A. Per iteration, both have the same communication cost, while the FactGD computation cost is lower than that of AltGDmin by a factor of r. Consequently, the total AltGDmin communication cost is lower than that of FactGD by a factor of r, while the total computation costs are similar. We provide a summary in Table 1.

The ProjGD sample complexity is worse than that of AltGDmin by a factor of $\log n \log(1/\epsilon)$. Moreover, the ProjGD communication cost per iteration is much higher than that of AltGDmin and FactGD. The same is true for its computation cost making it one of the slowest and most communication inefficient.

The original sample complexity of AltMin and Smooth AltMin is much higher than that of AltGDmin or FactGD (see Table 1). However, the corollary that we prove for AltMin later in Sec 7 shows that the AltMin sample complexity is $\kappa^4 \mu^2 n r^2 \log n \log(1/\epsilon)$ and its iteration complexity is $\log(1/\epsilon)$.

In summary, when compared with all other methods, AltGDmin has the lowest total communication cost. It also has one of the lowest total computation costs and the second lowest sample cost. Moreover, AltGDmin is private and requires just one data exchange per algorithm iteration. Federated FactGD can be made private with two data exchanges per algorithm iteration. AltMin can be made private by solving the LS problem needed to update U using GD rather than the closed form solution. This requires order $\log(1/\epsilon)$ extra data exchanges per main algorithm iteration. AltMin-Private is much slower both in numerical experiments and using theoretical complexities; see Table 1 and Sec. A.

The reason that AltGDmin needs a factor of $\log(1/\epsilon)$ in its sample complexity is because its proof uses sample-splitting. To our best knowledge, all guarantees for all solutions for LRMC or LRCS that factor X as $X = UB$, and in which one (or both) of the two alternating steps is a minimization step, use sample-splitting [Netrapalli et al., 2013, Hardt, 2014, Nayer et al., 2020, Nayer & Vaswani, 2021, Nayer & Vaswani, 2023b, Nayer & Vaswani, 2023a]. Some of this cited work provides guarantees for LR phase retrieval which is a phaseless measurements' generalization of LRCS; and hence any LRPR solution automatically solves LRCS. All three problems involve recovery from non-global measurements (no scalar measurement depends on the entire $X^\star$). For such problems, properties such as restricted isometry property cannot be proved.

The reason that AltGDmin iteration and sample complexity need an extra factor of κ^2 is due to the orthonormalization step (QR step) when updating U. For any algorithm that updates one or both of U, B by GD, such a step is one way to prevent the norm of the estimate U from increasing/decreasing over iterations (while that of B decreases/increases). The other way to ensure this is to add a norm balancing term in the cost function for the GD step for U, as done in FactGD. However this would then also require a row norm clipping step for U at each iteration, like in FactGD. Use of these two steps to replace the QR step can possibly help improve its dependence on κ. But it will also mean that we cannot borrow the overall proof approach developed for AltGDmin for LRCS in [Nayer & Vaswani, 2023a] that allows us to get a guarantee with a (nearly) constant step size, consequently, a (nearly) best iteration complexity of $\kappa^2 \log(1/\epsilon)$.

3.2 Proof Novelty

(i) The most important difference between LRCS and LRMC is that the LRMC proofs require incoherence of each iterate of U and of B. When analyzing AltGDmin for LRCS in [Nayer & Vaswani, 2023b, Nayer & Vaswani, 2023a], this was needed only for B because LRCS measurements are column-wise global (matrix A_k is dense and so each entry of y_k depends on the entire column $x_k^\star$). Our proof of incoherence of each updated U needs a different approach than that used in all past work on LRMC as well because our algorithm is different. (i) ProjGD [Cherapanamjeri et al., 2016] does not use a factorized representation for X and hence the gradient w.r.t. U does not exist for it. (ii) We do not update U by LS and hence we cannot use the approach used for AltMin or Smooth AltMin

[Netrapalli et al., 2013, Hardt, 2014]. (iii) FactGD [Yi et al., 2016] uses row norm clipping to ensure incoherence at each algorithm iteration, while AltGDmin does not use this (except for initialization). The details of our approach are explained below in the proof outlines, and in Lemma 6 (and proof) and its use in proving Theorem 2.2.

As noted earlier, not using row norm clipping at each iteration is one reason why we can borrow the overall proof technique used for LRCS [Nayer & Vaswani, 2023a] in order to prove a correctness guarantee with a constant GD step size. The other reason is that we use minimization for updating $\boldsymbol{B}$ (and not GD).

(ii) Unlike LRCS, LRMC measurements are both row-wise and column-wise local. Consequently, it is not possible to get a tight column-wise bound (bound on $\|\boldsymbol{b}_k - \boldsymbol{U}^\intercal \boldsymbol{x}_k^\star\|$, and hence on $\|\boldsymbol{x}_k - \boldsymbol{x}_k^\star\|$, for each k) under the desired sample complexity. We can only bound $\|\boldsymbol{B} - \boldsymbol{U}^\intercal \boldsymbol{X}^\star\|_F$ and $\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F$. This has implications on how we bound the gradient norm (both the gradient expected value and its deviation from it). This is one reason why the use of SD_F (instead of SD_2) in our bounding helps prove a better guarantee.

(iii) We need to use the matrix Bernstein inequality [Vershynin, 2018] for bounding all of terms, instead of the concentration bounds used in [Nayer & Vaswani, 2023a] (sub-Gausian Hoeffding or sub-exponential Bernstein inequality followed by an epsilon-net argument).

(iv) Our initialization guarantee is better than that of AltMin and Smooth Altmin [Netrapalli et al., 2013, Hardt, 2014] by a factor of r^3 and r respectively, and comparable to that of FactGD. The reason for this is that we adapt the row norm clipping idea of [Yi et al., 2016] for our initialization step. This approach is better than the entry-wise clipping used in [Netrapalli et al., 2013]. Row norm clipping can be analyzed in terms of $\mathrm{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^\star)$ and this is the other reason we use SD_F in our proofs (see Sec. 3.4).

(v) All analysis of AltGDmin for LRCS has been for the noise-free case. In this work, we also provide an extension of our result to the more practical noisy case. Our proof explains a general strategy for doing this for any iterative algorithm.

(vi) Lastly, we also show how our lemmas can be used to improve the sample complexity dependence of AltMin and Smoothed AltMin on r from $r^{4.5}$ and r^3 respectively to r^2.

3.3 Proof outline of Theorem 2.2

Let $\boldsymbol{U} = \boldsymbol{U}^{(t-1)}$, $\boldsymbol{G} := \boldsymbol{U}^\intercal \boldsymbol{X}^\star$ and $\boldsymbol{U}^+ = \boldsymbol{U}^{(t)}$. To prove Theorem 2.2, (i) we obtain a bound on $\delta^{(t)} := \mathrm{SD}_F(\boldsymbol{U}^+, \boldsymbol{U}^\star)$ in terms of $\delta^{(t-1)} = \mathrm{SD}_F(\boldsymbol{U}, \boldsymbol{U}^\star)$ that can be used to show exponential error decay, whp, under the desired sample complexity. (ii) This bound requires the initialization error $\delta^{(0)}$ to be small enough. Hence we also need to analyze the initialization step to show that this is true whp. Initialization is analyzed in Lemma 1 (proof outline given below). The overall idea for (i) is borrowed from [Nayer & Vaswani, 2023a] but there are differences because we need each new $\boldsymbol{U}$ to be incoherent. We use induction. The induction step assumes a certain bound on $\delta^{(t-1)}$ and on the row norms of $\boldsymbol{U}$ and proves the same for its updated version, $\boldsymbol{U}^+$. To do this, we first use the induction assumption (which implies $\kappa^2 \mu$ row-incoherence of $\boldsymbol{U}$) and the fact that $\mathbb{E}[\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B})] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal$ to get a deterministic bound on $\delta^{(t)}$. Next, we get high probability bounds on the terms of this bound using matrix Bernstein and appropriate linear algebra. These are obtained in the lemmas given in Sec. 4.2. The last step is to use the induction assumption and Lemma 6 to bound the row norms of $\boldsymbol{U}^+$.

Finally, we simplify the bounds in order to show that, whp, $\delta^{(t)}$ decays exponentially with t as long as η is at most $0.5/(p\sigma_{\max}^{*}{}^{2})$ and $\delta^{(0)} \leq c/\kappa^2$. The proof is in Sec. 4.3 and it relies on the six lemmas stated in Sec. 4.2.

3.4   Proof outlines of the lemmas used to prove Theorem 2.2

Initialization Lemma 1 uses the following ideas, many of which are borrowed from initialization step analysis of FactGD [Yi et al., 2016] (with filling in the missing details from there). (i) Using the results of [Chen et al., 2021] (Lemma 3.21, Theorem 3.22), we can bound $\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^{\star})$. (ii) The row norm clipping step can be interpreted as projecting its input onto a convex set, $\mathcal{U} := \{\boldsymbol{U} : \|\boldsymbol{u}^j\| \leq \mu\sqrt{r/n}\}$, with the projection being in Frobenius norm. 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, $\Pi_{\mathcal{U}}(\boldsymbol{U}^{\star}\boldsymbol{Q}) = \boldsymbol{U}^{\star}\boldsymbol{Q}$ for any $r \times r$ unitary matrix $\boldsymbol{Q}$ (since $\boldsymbol{U}^{\star}$ as well as $\boldsymbol{U}^{\star}$ times any unitary matrix belong to $\mathcal{U}$). (iii) Let $\boldsymbol{Q}_{*,00} := \arg\min_{\boldsymbol{Q}\ unitary} \|\boldsymbol{U}^{(00)} - \boldsymbol{U}^{\star}\boldsymbol{Q}\|_F$. By Lemmas 2.5 and 2.6 of [Chen et al., 2021], $\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^{\star}) \approx \|\boldsymbol{U}^{(00)} - \boldsymbol{U}^{\star}\boldsymbol{Q}_{*,00}\|_F$ (upper and lower bounded by RHS with a factor of $\sqrt{2}$ for upper bound). (iv) The above ideas help bound $\|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^{\star}\boldsymbol{Q}_{*,00}\|_F$. In the last step, we use this bound and $\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) \overset{\mathrm{QR}}{=} \boldsymbol{U}^{(0)}\boldsymbol{R}^{(0)}$ to bound $\mathrm{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^{\star})$. The proof is in Sec. 5.1.

Lemma 2 uses incoherence of $\boldsymbol{U}$ and $\delta^{(t-1)}$ to bound $\|\boldsymbol{B} - \boldsymbol{G}\|_F$ where $\boldsymbol{G} = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{X}^{\star}$. This is proved by writing $vec(\boldsymbol{B} - \boldsymbol{G}) = \boldsymbol{F}^{-1}vec(\boldsymbol{D})$, where $\boldsymbol{F} \in \mathbb{R}^{nr\times nr}$ and $\boldsymbol{D} \in \mathbb{R}^{r\times q}$ are defined below in Sec. 4.1, bounding $\|\boldsymbol{F}^{-1}\|$ and $\|\boldsymbol{D}\|$ by the matrix Bernstein inequality, and using $\|\boldsymbol{B} - \boldsymbol{G}\|_F = \|vec(\boldsymbol{B} - \boldsymbol{G})\| \leq \|\boldsymbol{F}^{-1}\| \cdot \|vec(\boldsymbol{D})\| = \|\boldsymbol{F}^{-1}\| \cdot \|\boldsymbol{D}\|_F \leq \|\boldsymbol{F}^{-1}\| \cdot \sqrt{r}\|\boldsymbol{D}\|$. The bound on $\|\boldsymbol{F}^{-1}\|$ is borrowed from [Netrapalli et al., 2013], while the rest of our bounding is different and simpler than the approach used in [Netrapalli et al., 2013]. We use matrix Bernstein which provides a much simpler proof and the resulting bound holds with a smaller sample complexity (better dependence on r). The proof is in Sec. 5.2. Lemma 3 uses the $\|\boldsymbol{B} - \boldsymbol{G}\|_F$ bound to bound $\|\boldsymbol{X} - \boldsymbol{X}\|_F$, and the minimum and maximums singular values of $\boldsymbol{B}$. The proof idea is similar to that in [Nayer & Vaswani, 2023b]. We provide the proof in Sec. 5.3 anyway since it is very short.

Lemma 4 uses incoherence of $\boldsymbol{U}$ to show incoherence of $\boldsymbol{B}$. It follows by writing $\boldsymbol{b}_k = (\boldsymbol{U}_k^{\mathsf{T}}\boldsymbol{U}_k)^{-1}(\boldsymbol{U}_k^{\mathsf{T}}\boldsymbol{U}^{\star}_k)\boldsymbol{b}_k^{\star}$, using the fact that $\mathbb{E}[\boldsymbol{U}_k^{\mathsf{T}}\boldsymbol{U}_k] = p\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U} = p\boldsymbol{I}$, $\mathbb{E}[\boldsymbol{U}_k^{\mathsf{T}}\boldsymbol{U}^{\star}_k] = p\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}^{\star}$, $\|\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}^{\star}\| \leq 1$, and using matrix Bernstein to bound deviations of both matrices from their expected values. This proof is also much simpler than the one given in [Netrapalli et al., 2013] and the result needs a smaller sample complexity (better dependence on r). The proof is in Sec. 5.4.

Lemma 5 uses the $\boldsymbol{B} - \boldsymbol{G}$ bound and its implications, $\delta^{(t-1)} < c/\kappa^2$, and incoherence of $\boldsymbol{U}$ and of $\boldsymbol{B}$ to bound the gradient norm. It relies on the following ideas. $\mathbb{E}[\mathrm{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^{\star})\boldsymbol{B}^{\mathsf{T}}$. This holds since the expectation is taken with respect to an independent set of samples at each iteration (possible because of sample splitting). We bound the deviation from the expected value using matrix Bernstein inequality and the assumed bounds in the lemma assumptions. The proof is in Sec. 5.5.

Lemma 6 uses incoherence of $\boldsymbol{B}$ to show incoherence of $\boldsymbol{U}$. (i) We first bound the deviation of the j-th row of GradU, denoted GradU^j, from its expected value in terms of $\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)$. To get a high probability bound of this form, we use matrix Bernstein inequality and the fact that, for showing incoherence, we do not need the bound to contain $\delta^{(t-1)}$. Hence, we can use a seemingly loose bound of the form $|\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^{\star}| \leq 2\max(|\boldsymbol{x}_{jk}|, |\boldsymbol{x}_{jk}^{\star}|) \leq 2.2\mu\sqrt{r/q}\sigma_{\max}^{*} \cdot \max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)$. This bound is, in fact, tighter than a bound that contains $\delta^{(t-1)}$ for the initial iterations when $\delta^{(t-1)}$ is allowed to be larger than order $\sqrt{r/q}$. (ii) Next we use $\widetilde{\boldsymbol{u}}^{j(t+1)} = \boldsymbol{u}^{j(t)} - \eta\mathrm{Grad}\boldsymbol{u}^j \pm \mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j]$,

$\mathbb{E}[\mathrm{GradU}^j] = p(\boldsymbol{u}^{j\intercal}\boldsymbol{B}\boldsymbol{B}^\intercal - \boldsymbol{u}^{*j\intercal}\boldsymbol{B}^\star\boldsymbol{B}^\intercal)$, and an upper bound on the GD step size η to show the second part of this lemma: $\|\boldsymbol{u}^{j^+}\| \le (1 - c/\kappa^2)\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\|$. The proof is provided in Sec. 5.6.

4 Proving Theorem 2.2

4.1 Definitions and Expressions

Let $\boldsymbol{U} \equiv \boldsymbol{U}^{(t-1)}$, $\boldsymbol{B} \equiv \boldsymbol{B}^{(t)}$, and $\boldsymbol{X} \equiv \boldsymbol{X}^{(t)} = \boldsymbol{U}^{(t-1)}\boldsymbol{B}^{(t)}$. Also let $\boldsymbol{U}^+ \equiv \boldsymbol{U}^{(t)}$, $\boldsymbol{B}^+ \equiv \boldsymbol{B}^{(t+1)}$. Let $\boldsymbol{G} := \boldsymbol{U}^\intercal\boldsymbol{X}^\star$ and $\boldsymbol{P}_{*,\perp} := \boldsymbol{I} - \boldsymbol{U}^\star\boldsymbol{U}^{\star\intercal}$. Finally, Let

$$\delta^{(t)} := \mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*) = \|\boldsymbol{P}_{*,\perp}\boldsymbol{U}^+\|_F$$

Thus, $\delta^{(t-1)} = \mathrm{SD}_F(\boldsymbol{U}, \boldsymbol{U}^\star)$, $\delta^{(t)} = \mathrm{SD}_F(\boldsymbol{U}^+, \boldsymbol{U}^\star) = \|\boldsymbol{P}_{*,\perp}\boldsymbol{U}^+\|_F$, and $\boldsymbol{P}_{*,\perp}\boldsymbol{X}^\star = \boldsymbol{0}$.

Let $\mu_u := 20\kappa^2\mu$. All our proofs use c, C to denote different numerical constants in each use. The numerical values in our intermediate steps are often loose bounds to make the analysis simpler.

Recall that $\boldsymbol{U}_k := \boldsymbol{S}_k\boldsymbol{U}$ where $\boldsymbol{S}_k = \mathrm{diag}(\xi_{jk}, j \in [n])$. Similarly, let $\boldsymbol{U}^\star{}_k := \boldsymbol{S}_k\boldsymbol{U}^\star$. Thus

$$\boldsymbol{S}_k^\intercal\boldsymbol{S}_k = \boldsymbol{S}_k\boldsymbol{S}_k = \boldsymbol{S}_k$$

$$\boldsymbol{U}_k = \boldsymbol{S}_k\boldsymbol{U} = \sum_{j=1}^n \xi_{jk}\boldsymbol{e}_j\boldsymbol{u}^{j\intercal}, \ \boldsymbol{U}^\star{}_k = \sum_{j=1}^n \xi_{jk}\boldsymbol{e}_j\boldsymbol{u}^{*j\intercal}$$

and consequently,

$$\boldsymbol{U}_k^\intercal\boldsymbol{U}^\star{}_k = \boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U}^\star = \sum_{j=1}^n \xi_{jk}\boldsymbol{u}^j\boldsymbol{u}^{*j\intercal}$$

Using this,

$$\boldsymbol{b}_k = \boldsymbol{U}_k^\dagger\boldsymbol{y}_k = \boldsymbol{U}_k^\dagger\boldsymbol{S}_k\boldsymbol{x}_k^\star = (\boldsymbol{U}_k^\intercal\boldsymbol{U}_k)^{-1}\boldsymbol{U}_k^\intercal\boldsymbol{U}^\star{}_k\boldsymbol{b}_k^\star = (\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U})^{-1}\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U}^\star\boldsymbol{b}_k^\star$$

Thus, $\boldsymbol{b}_k - \boldsymbol{g}_k = \boldsymbol{b}_k - \boldsymbol{U}^\intercal\boldsymbol{U}^\star\boldsymbol{b}_k^\star$ simplifies to

$$\boldsymbol{b}_k - \boldsymbol{g}_k = (\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U})^{-1}[\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U}^\star\boldsymbol{b}_k^\star - (\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U})\boldsymbol{U}^\intercal\boldsymbol{U}^\star\boldsymbol{b}_k^\star] = (\underbrace{\boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U}}_{\boldsymbol{F}_k})^{-1}\underbrace{\boldsymbol{U}^\intercal\boldsymbol{S}_k[\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal]\boldsymbol{U}^\star\boldsymbol{b}_k^\star}_{\boldsymbol{d}_k} \quad (4)$$

with

$$\boldsymbol{d}_k := \boldsymbol{U}^\intercal\boldsymbol{S}_k[\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal]\boldsymbol{U}^\star\boldsymbol{b}_k^\star, \ \boldsymbol{F}_k := \boldsymbol{U}^\intercal\boldsymbol{S}_k\boldsymbol{U}$$

Thus, letting $\boldsymbol{F} := \mathrm{blockdiag}(\boldsymbol{F}_k, k \in [q])$,

$$vec(\boldsymbol{B} - \boldsymbol{G}) = \boldsymbol{F}^{-1}vec(\boldsymbol{D}), \ \text{where } \boldsymbol{D} := [\boldsymbol{d}_1, \boldsymbol{d}_2, \dots \boldsymbol{d}_q] = \sum_k\sum_j \xi_{jk}\boldsymbol{u}^j[[\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal]\boldsymbol{U}^\star]^{j\intercal}\boldsymbol{b}_k^\star\boldsymbol{e}_k^\intercal$$

In the above $\boldsymbol{F} \in \mathbb{R}^{rq \times rq}$ and $\boldsymbol{D} \in \mathbb{R}^{r \times q}$. Observe that

$$\mathbb{E}[\boldsymbol{S}_k] = p\boldsymbol{I}, \ \mathbb{E}[\boldsymbol{D}] = \boldsymbol{0}$$

Finally,

$$\mathrm{GradU} := \nabla_{\boldsymbol{U}}f(\boldsymbol{U}, \boldsymbol{B}) = 2(\boldsymbol{X}_\Omega - \boldsymbol{Y})\boldsymbol{B}^\intercal = 2\sum_{k=1}^q \boldsymbol{S}_k(\boldsymbol{x}_k - \boldsymbol{x}_k^\star)\boldsymbol{b}_k^\intercal = 2\sum_{k=1}^q\sum_{j=1}^n \xi_{jk}\boldsymbol{e}_j(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star)\boldsymbol{b}_k^\intercal \quad (5)$$

12

where $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, $\boldsymbol{X}^\star = \boldsymbol{U}^\star\boldsymbol{B}^\star$, $\boldsymbol{x}_{jk} = \boldsymbol{u}^{j\intercal}\boldsymbol{b}_k$ and $\boldsymbol{x}_{jk}^\star = \boldsymbol{u}^{*j\intercal}\boldsymbol{b}_k^\star$. Also,

$$\text{GradU}^j := \boldsymbol{e}_j^\intercal \nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}) = 2\sum_{k=1}^q \xi_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star)\boldsymbol{b}_k^\intercal$$

Observe that
$$\mathbb{E}[\text{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal \text{ and } \mathbb{E}[\text{GradU}^j] = p\boldsymbol{e}_j^\intercal(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal$$

4.2 Lemmas for proving Theorem 2.2

All lemmas below assume Assumption 1 (singular vectors' incoherence) holds. Also, everywhere $\epsilon < 1$.

Lemma 1 (Initialization). *Pick a $\delta \leq 0.2$. Assume $p \geq C\kappa^2 r^2 \mu \log q/(n\delta^2)$. Then, w.p. at least $1 - n^{-10}$.*

   (i) $\text{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^*) \leq \delta$.

   (ii) $\boldsymbol{U}^{(0)}$ is 1.5μ row-incoherent, i.e., $\|\boldsymbol{u}^{j(0)}\| \leq 1.5\mu\sqrt{r/n}$ for all $j \in [n]$.

Lemma 2 (LS step analysis: error bound for $\boldsymbol{B}$). *Assume $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$. Then, w.p. at least $1 - \exp(\log q - c\frac{\epsilon^2 pn}{r^2\mu_u^2})$,*

$$\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \epsilon\delta^{(t-1)}\sigma_{\max}^*$$

Lemma 3 (Implications of error bound for $\boldsymbol{B}$). *Assume $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t-1)}\sigma_{\max}^*$. Then,*

   (i) $\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F \leq 2\delta^{(t-1)}\sigma_{\max}^*$.

   (ii) $\sigma_{\max}(\boldsymbol{B}) \leq (1 + \delta^{(t-1)})\sigma_{\max}^*$ and $\sigma_{\min}(\boldsymbol{B}) \geq \sqrt{1 - \delta^{(t-1)2}}\sigma_{\min}^* - \delta^{(t-1)}\sigma_{\max}^*$.

   (iii) Thus, if $\delta^{(t-1)} \leq c/\kappa$, then $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma_{\min}^*$ and $\sigma_{\max}(\boldsymbol{B}) \leq 1.1\sigma_{\max}^*$.

Lemma 4 (Incoherence of $\boldsymbol{B}$). *Assume that $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$. Then, w.p. greater than $1 - \exp(\log q - c\frac{np}{\mu_u^2 r})$,*

$$\|\boldsymbol{b}_k\| \leq 1.1\sigma_{\max}^*\mu\sqrt{r/q} \text{ for all } k \in [q].$$

Lemma 5 (Gradient expression and bounds). *Assume that $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t-1)}\sigma_{\max}^*$ with $\delta^{(t-1)} \leq c/\kappa$, $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$, $\|\boldsymbol{b}_k\| \leq \sigma_{\max}^*\mu\sqrt{r/q}$. Then,*

   (i) $\|\mathbb{E}[\text{GradU}]\|_F \leq 2.5p\delta^{(t-1)}\sigma_{\max}^{*2}$.

   (ii) $\|\text{GradU} - \mathbb{E}[\text{GradU}]\|_F \leq \epsilon p\delta^{(t-1)}\sigma_{\min}^{*2}$, w.p. at least $1 - \exp(\log q - c\frac{\epsilon^2 pn}{\max(\kappa^4\mu^2, \kappa^2\mu_u\mu)r^2})$

Lemma 6 (Incoherence of $\boldsymbol{U}$). *Assume that $\|\boldsymbol{b}_k\| \leq 1.1\sigma_{\max}^*\mu\sqrt{r/q}$. Then, w.p. greater than $1 - \exp(\log q - c\epsilon^2 pn/\kappa^4\mu^2 r^2)$*

   (i) $\|\text{Grad}\boldsymbol{u}^j - \mathbb{E}[\text{Grad}\boldsymbol{u}^j]\| \leq \epsilon p \max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\sigma_{\min}^{*2}$

   (ii) Further, if $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t-1)}\sigma_{\max}^*$ with $\delta^{(t-1)} < c/\kappa$, and GD step size $\eta \leq 0.5/(p\sigma_{\max}^{*2})$, then

$$\|\boldsymbol{u}^{j+}\| \leq (1 - 0.15/\kappa^2)\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\|$$

Proof. All these lemmas are proved in Sec 5. $\qquad\square$

13

4.3 Proof of Theorem 2.2

We prove the following result, Claim 3, using an induction argument and the lemmas from Sec. 4.2. Theorem 2.2 follows from the first claim of this result along with using the following:

- To ensure $\delta^{(T)} \leq \epsilon$, we need $(1 - c/\kappa^2)^T \cdot (c/\kappa^2) \leq \epsilon$ or that $T = C\kappa^2 \log(1/\epsilon)$.

- Since we use sample splitting, the total number of samples needed for all T iterations is $2T \cdot np \geq C\kappa^4 \mu^2 r^2 \log q \cdot \kappa^2 \log(1/\epsilon)$. If all samples are generated first and then subsets used across different iterations (as specified in our stepwise algorithm, Algorithm 1), we need to assume that an entry is observed w.p. $p_{tot} = 2T \cdot p$ and $np_{tot} \geq C\kappa^4 \mu^2 r^2 \log q \cdot \kappa^2 \log(1/\epsilon)$.

Claim 3. *Assume everything stated in Theorem 2.2 and $np \geq C\kappa^4 \mu^2 r^2 \log q$. For all times $\tau \geq 0$, the following hold w.p. at least $1 - \tau/n^3$: (i) $\delta^{(\tau)} \leq (1 - c/\kappa^2)^\tau \cdot (c/\kappa^2)$; and (ii) $\|u^{j(\tau)}\| \leq (1 - \frac{0.15}{\kappa^2})^\tau 1.5\mu\sqrt{r/n} + [1 + (1 - \frac{0.15}{\kappa^2}) + \cdots + (1 - \frac{0.15}{\kappa^2})^{\tau-1}]2\|u^{*j}\|$*

Base case: Lemma 1 shows that $\delta^{(0)} \leq \delta = 0.1/\kappa^2$ and $\|u^{j(0)}\| \leq 1.5\mu\sqrt{r/n}$. This proves (i) and (ii) for $\tau = 0$.

Induction assumption: Assume that the claim holds for $\tau = t - 1$.

Induction step: Consider $\tau = t$. Recall from Algorithm that $\widetilde{U}^+ = U - \eta\mathrm{GradU}$, and $U^+ = \widetilde{U}^+ R^{+-1}$ where $\widetilde{U}^+ \overset{\mathrm{QR}}{=} U^+ R^+$. Recall from Sec. 4.1 that $P_{*,\perp} X^\star = 0$, $\mathbb{E}[\mathrm{GradU}] = p(UB - X^\star)B^\intercal$, $\|P_{*,\perp}U\|_F = \delta^{(t-1)}$. Also, using Weyl's inequality, $\sigma_{\min}(R^+) = \sigma_{\min}(\widetilde{U}^+) \geq \sigma_{\min}(U) - \eta\|\mathrm{GradU}\| = 1 - \eta\|\mathrm{GradU}\|$. Using these,

$$
\begin{aligned}
\delta^{(t)} &= \mathrm{SD}_F(U^+, U^\star) = \|P_{*,\perp}U^+\|_F \\
&\leq \|P_{*,\perp}\widetilde{U}^+\|_F \cdot \|(R^+)^{-1}\| = \|P_{*,\perp}\widetilde{U}^+\|_F/\sigma_{\min}(\widetilde{U}^+) \\
&\leq \frac{\|P_{*,\perp}(U - \eta\mathbb{E}[\mathrm{GradU}] + \eta\mathbb{E}[\mathrm{GradU}] - \eta\mathrm{GradU})\|_F}{(1 - \eta\|\mathrm{GradU}\|)} \\
&\leq \frac{\|P_{*,\perp}(U - \eta p(UB - X^\star)B^\intercal)\|_F + \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU})\|_F}{(1 - \eta\|\mathbb{E}[\mathrm{GradU}]\| - \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU}\|)} \\
&\leq \frac{\delta^{(t-1)} \cdot \|I - \eta p BB^\intercal\| + \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU})\|_F}{(1 - \eta\|\mathbb{E}[\mathrm{GradU}]\| - \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU}\|)}
\end{aligned}
\tag{6}
$$

By the induction assumption, $\delta^{(t-1)} \leq (1 - c/\kappa^2)^{t-1} \cdot (c/\kappa^2) \leq (c/\kappa^2)$ and

$$
\|u^j\| = \|u^{j(t-1)}\| \leq (1 - \frac{0.15}{\kappa^2})^{t-1}\|u^{j(0)}\| + [1 + (1 - \frac{0.15}{\kappa^2}) + \cdots + (1 - \frac{0.15}{\kappa^2})^{t-2}]2\|u^{*j}\|
$$

$$
\leq \|u^{j(0)}\| + \frac{\kappa^2}{0.15}2\|u^{*j}\| \leq (1.5\mu + 14\kappa^2\mu)\sqrt{r/n} \leq \mu_u\sqrt{r/n}
\tag{7}
$$

The last inequality above used the infinite geometric series bound. This shows that U is μ_u-row-incoherent.

Using (7) and $\delta^{(t-1)} \leq (c/\kappa^2)$, Lemmas 2 and 4 hold, i.e. the bound on $\|B - G\|_F$ and $\|b_k\|$ holds. This then implies that all claims of Lemma 3 hold too. This then implies that Lemmas 5 and 6 hold. All the lemmas hold with probability at least $1 - 1/n^3$ if p is such that all the probabilities stated in all the lemmas are lower bounded by $1 - 0.1/n^3$. Using $\mu_u = 8\kappa^2\mu$, clearly, this is true if

$$
np \geq C\kappa^4\mu^2 r^2 \log q
$$

14

By Lemmas 2 and 3, $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma_{\min}^*$ and $\|\boldsymbol{B}\| \leq 1.1\sigma_{\max}^*$. Using these, if $\eta \leq 0.5/(p\sigma_{\max}^{*2})$, then $\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\intercal$ is positive semi-definite (psd) and hence $\|\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\intercal\| = \lambda_{\max}(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\intercal) \leq 1 - 0.8\eta p\sigma_{\min}^{*2}$. Using Lemma 4, $\boldsymbol{B}$ is 1.1μ-column-incoherent. Using Lemma 5, $\|\mathbb{E}[\text{GradU}]\| \leq 2.5p\delta^{(t-1)}\sigma_{\max}^{*2}$ and $\|\text{GradU} - \mathbb{E}[\text{GradU}]\|_F \leq \epsilon p\delta^{(t-1)}\sigma_{\min}^{*2}$. Substituting these into (6) with $\epsilon = 0.01$,

$$\delta^{(t)} \leq \frac{\delta^{(t-1)}\left(1 - 0.8\eta p\sigma_{\min}^{*2} + 0.01\eta p\sigma_{\min}^{*2}\right)}{1 - \delta^{(t-1)}\left(2.55\eta p\sigma_{\max}^{*2}\right)}.$$

$$\leq \delta^{(t-1)}(1 - 0.79\eta p\sigma_{\min}^{*2})(1 + 2\delta^{(t-1)}(2.55\eta p\sigma_{\max}^{*2}))$$

$$\leq \delta^{(t-1)}(1 - (0.79 - \delta^{(t-1)}5.1\kappa^2)\eta p\sigma_{\min}^{*2}) \leq \delta^{(t-1)}(1 - \eta p\sigma_{\min}^{*2}(0.79 - 0.051)) \leq \delta^{(t-1)}(1 - 0.3\eta p\sigma_{\min}^{*2})$$

In the above, for the denominator term we used $1/(1-x) \leq 1 + 2x$ for $x < 0.5$ and $\delta^{(t-1)} \leq 0.01/\kappa^2$. Setting $\eta = 0.5/(p\sigma_{\max}^{*2})$ in the final expression given above, we can conclude that

$$\delta^{(t)} \leq (1 - 0.15/\kappa^2)\delta^{(t-1)} \leq (1 - 0.15/\kappa^2)^t \cdot (0.1/\kappa^2)$$

Thus claim (i) holds for $\tau = t$. Next we prove claim (ii) for $\tau = t$. By Lemma 6 and the induction assumption,

$$\|\boldsymbol{u}^{j(t)}\| = \|\boldsymbol{u}^{j+}\| \leq (1 - 0.15/\kappa^2)\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\|$$

$$\leq (1 - 0.15/\kappa^2)^t 1.5\mu\sqrt{r/n} + (1 - 0.15/\kappa^2)[1 + (1 - \frac{0.15}{\kappa^2}) + \cdots + (1 - \frac{0.15}{\kappa^2})^{t-2}]2\|\boldsymbol{u}^{*j}\| + 2\|\boldsymbol{u}^{*j}\|$$

$$= (1 - 0.15/\kappa^2)^t 1.5\mu\sqrt{r/n} + [1 + (1 - \frac{0.15}{\kappa^2}) + \cdots + (1 - \frac{0.15}{\kappa^2})^{t-1}]2\|\boldsymbol{u}^{*j}\|$$

5 Proofs of the Lemmas

5.1 Proof of Lemma 1

By Lemmas 2.5 and 2.6 of [Chen et al., 2021], for two $n \times r$ matrices with orthonormal columns, $\boldsymbol{U}_1, \boldsymbol{U}_2$,

$$\text{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2) \leq \min_{\boldsymbol{Q} \ unitary} \|\boldsymbol{U}_1 - \boldsymbol{U}_2\boldsymbol{Q}\|_F \leq \sqrt{2}\text{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2) \tag{8}$$

and a similar bound holds for SD_2 as well. We use this fact frequently below.

Recall from the Algorithm that $\boldsymbol{Y} \overset{\text{SVD}}{=} \boldsymbol{U}^{(00)}\Sigma^{(00)}\boldsymbol{V}^{(00)}$, and $\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) \overset{\text{QR}}{=} \boldsymbol{U}^{(0)}\boldsymbol{R}^{(0)}$. By Theorem 3.22 of [Chen et al., 2021] along with using (8) and $\text{SD}_2(\boldsymbol{U}_1, \boldsymbol{U}_2) \leq \sqrt{r}\text{SD}_F(\boldsymbol{U}_1, \boldsymbol{U}_2)$, we have: with probability at least $1 - Cq^{-10}$,

$$\text{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^\star) \leq C\sqrt{\frac{\kappa^2\mu r^2 \log q}{np}}$$

Let $\boldsymbol{Q}_{*,00} = \arg\min_{\boldsymbol{Q} \ unitary} \|\boldsymbol{U}^{(00)} - \boldsymbol{U}^\star\boldsymbol{Q}\|_F$. By (8),

$$\|\boldsymbol{U}^{(00)} - \boldsymbol{U}^\star\boldsymbol{Q}_{*,00}\|_F \leq \sqrt{2}\text{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^\star)$$

Next we use the above two bounds and the fact that $\mathcal{U}$ is a convex set[1] to bound $\|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^\star\boldsymbol{Q}_{*,00}\|_F$ as follows.

$$\|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^\star\boldsymbol{Q}_{*,00}\|_F = \|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \Pi_{\mathcal{U}}(\boldsymbol{U}^\star\boldsymbol{Q}_{*,00})\|_F$$

---

[1]To see that $\mathcal{U}$ is a convex set, let matrices $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathcal{U}$. Let $\boldsymbol{m}_1^j$ and $\boldsymbol{m}_2^j$ denote the rows of $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, respectively. For any $0 < \theta < 1$, $\|\theta\boldsymbol{m}_1^j + (1-\theta)\boldsymbol{m}_2^j\| \leq \theta\|\boldsymbol{m}_1^j\| + (1-\theta)\|\boldsymbol{m}_2^j\| \leq \theta\mu\sqrt{r/n} + (1-\theta)\mu\sqrt{r/n} = \mu\sqrt{r/n}$.

$$\leq \|\boldsymbol{U}^{(00)} - \boldsymbol{U}^\star \boldsymbol{Q}_{*,00}\|_F \leq \sqrt{2}\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^\star) \leq C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}}$$

with probability at least $1 - Cq^{-10}$. The first equality above uses the fact that, for any unitary $\boldsymbol{Q}$, $\boldsymbol{U}^\star \boldsymbol{Q}$ belongs to $\mathcal{U}$. This is a direct consequence of our incoherence assumption Assumption 1. The second one uses the facts that projection onto $\mathcal{U}$, which is a convex set, is non-expansive [Nashed, 1968, eq. (9),(10)], [Zarantonello, 1971, eq. (1.5)]). The projection $\Pi_{\mathcal{U}}(\boldsymbol{M}) := \min_{\boldsymbol{U} \in \mathcal{U}} \|\boldsymbol{U} - \boldsymbol{M}\|_F$ is non-expansive means that $\|\Pi_{\mathcal{U}}(\boldsymbol{M}) - \Pi_{\mathcal{U}}(\boldsymbol{M}_2)\|_F \leq \|\boldsymbol{M} - \boldsymbol{M}_2\|_F$.

Finally, we use the above to bound $\mathrm{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^\star) = \|\boldsymbol{P}_{*,\perp}\boldsymbol{U}^{(0)}\|_F = \|\boldsymbol{P}_{*,\perp}\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)})\boldsymbol{R}^{(0)-1}\|_F$. Using $\boldsymbol{P}_{*,\perp}\boldsymbol{U}^\star \boldsymbol{Q}_{*,00} = \boldsymbol{0}$, and $x/(1-x) < 1.5x$ for any $0 \leq x \leq 0.3$, we have: if $C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}} < 0.3$, then

$$\begin{aligned}
\mathrm{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^\star) = \|\boldsymbol{P}_{*,\perp}\boldsymbol{U}^{(0)}\|_F &\leq \|\boldsymbol{P}_{*,\perp}\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)})\|_F \|\boldsymbol{R}^{(0)-1}\| \\
&\leq \frac{\|\boldsymbol{P}_{*,\perp}\boldsymbol{U}^\star \boldsymbol{Q}_{*,00}\|_F + \|\boldsymbol{P}_{*,\perp}(\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^\star \boldsymbol{Q}_{*,00})\|_F}{\sigma_{\min}(\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}))} \\
&\leq \frac{\|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^\star \boldsymbol{Q}_{*,00}\|_F}{1 - \|\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}) - \boldsymbol{U}^\star \boldsymbol{Q}_{*,00}\|_F} \leq \frac{C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}}}{1 - C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}}} \leq 1.5C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}}
\end{aligned}$$

The denominator bound follows by Weyl's inequality, $\sigma_{\min}(\boldsymbol{U}^\star \boldsymbol{Q}_{*,00}) = 1$, and $x/(1-x) < 1.5x$ for $0 < x < 0.3$. Thus, for a $\delta < 0.3$, w.p. at least $1 - q^{-10}$, $\mathrm{SD}_F(\boldsymbol{U}^{(0)}, \boldsymbol{U}^\star) \leq \delta$ if $p \geq C\kappa^2 \mu r^2 \log q/n\delta^2$.

Proof of μ-incoherence We have $\boldsymbol{u}^{j(0)} = (\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}))^j \boldsymbol{R}^{(0)-1}$. Thus, using the denominator bound from above,

$$\|\boldsymbol{u}^{j(0)}\| \leq \|(\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}))^j\| \|\boldsymbol{R}^{(0)-1}\| = \frac{\|(\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}))^j\|}{\sigma_{\min}(\Pi_{\mathcal{U}}(\boldsymbol{U}^{(00)}))} \leq \frac{\mu\sqrt{r/n}}{1 - C\sqrt{\frac{\kappa^2 \mu r^2 \log q}{np}}}$$

Our assumed bound on p implies that the denominator is at least $1 - \delta$ for a $\delta < 0.3$. This then implies the above is upper bounded by $1.5\mu\sqrt{r/n}$.

5.2 Proof of Lemma 2

Let $\delta = \delta^{(t-1)}$. Recall from Sec. 4.1 that $vec(\boldsymbol{B} - G) = \boldsymbol{F}^{-1} vec(\boldsymbol{D})$ with $\boldsymbol{F}, \boldsymbol{D}$ defined there. Here $\boldsymbol{F} \in \mathbb{R}^{rq \times rq}$ and $\boldsymbol{D} \in \mathbb{R}^{r \times q}$. Thus,

$$\|\boldsymbol{B} - G\|_F = \|vec(\boldsymbol{B} - G)\| = \|\boldsymbol{F}^{-1} vec(\boldsymbol{D})\| \leq \|\boldsymbol{F}^{-1}\| \cdot \|vec(\boldsymbol{D})\| = \|\boldsymbol{F}^{-1}\| \cdot \|\boldsymbol{D}\|_F \leq \|\boldsymbol{F}^{-1}\| \cdot \sqrt{r}\|\boldsymbol{D}\| \tag{9}$$

By the matrix-Bernstein inequality, $\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k - p\boldsymbol{I}\| \leq \epsilon p$, w.p. greater than $1 - \exp(\log 2r - c\frac{\epsilon^2 pn}{\mu_u^2 r})$. Applying the union bound over all q diagonal blocks, and using the fact that $\sigma_{\min}(\boldsymbol{F}) = \min_k \sigma_{\min}(\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k)$ (since $\boldsymbol{F}$ is block diagonal),

$$\|\boldsymbol{F}^{-1}\| \leq \frac{1}{(1-\epsilon)p}, \quad \text{w.p. greater than } 1 - \exp(\log q + \log 2r - c\frac{\epsilon^2 pn}{\mu_u^2 r}) \tag{10}$$

(the above also follows by Lemma C.7 of [Netrapalli et al., 2013]).

16

Using the above bound and the bound on $\|\boldsymbol{D}\|$ that we derive below

$$\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \sqrt{r}\|\boldsymbol{F}^{-1}\|\|\boldsymbol{D}\| \leq 2\epsilon\sqrt{r}\delta\sigma^*_{\max}, \;\; \text{w.p. greater than } 1 - \exp(\log q - \frac{\epsilon^2 pn}{r\mu_u^2}) \tag{11}$$

Let $\epsilon = \epsilon_1/2\sqrt{r}$. Then,

$$\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \|\boldsymbol{F}^{-1}\| \cdot \sqrt{r}\|\boldsymbol{D}\| \leq \epsilon_1\delta\sigma^*_{\max}, \;\; \text{w.p. greater than } 1 - \exp(\log q - \frac{\epsilon^2 pn}{r^2\mu_u^2}) \tag{12}$$

$\|D\|$ **bound** Let
$$\boldsymbol{M} := [(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{U}^\star]$$

Recall from Sec. 4.1 that

$$\boldsymbol{D} = \sum_k \boldsymbol{d}_k \boldsymbol{e}_k^\mathsf{T} = \boldsymbol{U}^\mathsf{T}\boldsymbol{S}_k \boldsymbol{M}\boldsymbol{b}_k^\star \boldsymbol{e}_k^\mathsf{T} = \sum_k \sum_j \underbrace{\xi_{jk}\boldsymbol{u}^j \boldsymbol{M}^{j\mathsf{T}}\boldsymbol{b}_k^\star \boldsymbol{e}_k^\mathsf{T}}_{\boldsymbol{Z}_{jk}} \text{ and } \mathbb{E}[\boldsymbol{D}] = \boldsymbol{0} \tag{13}$$

We bound $\|\boldsymbol{D}\| = \|\boldsymbol{D} - \mathbb{E}[\boldsymbol{D}]\|$ using matrix Bernstein inequality. Using $n \leq q$,

$$L = \max_{jk}\|\xi_{jk}\boldsymbol{u}_j \boldsymbol{M}^{j\mathsf{T}}\boldsymbol{b}_k^\star \boldsymbol{e}_k^\mathsf{T}\| \leq \max_{jk}\|\boldsymbol{M}^j\|\|\boldsymbol{u}_j\|\|\boldsymbol{b}_k^\star\| \leq \delta\sigma^*_{\max}\mu\mu_u r/n$$

where we used $\|\boldsymbol{M}^j\| \leq \|\boldsymbol{M}\| \leq \delta$. Also, using $\mathbb{E}[\xi_{jk}] = \mathbb{E}[\xi_{jk}^2] = p$,

$$\sigma_1^2 = \|\mathbb{E}[\sum_{jk}\boldsymbol{Z}_{jk}^\mathsf{T}\boldsymbol{Z}_{jk}]\| = p\|\sum_{jk}(\boldsymbol{M}^{j\mathsf{T}}\boldsymbol{b}_k^\star)^2\|\boldsymbol{u}_j\|^2 \boldsymbol{e}_k \boldsymbol{e}_k^\mathsf{T}\|$$

$$\leq p\mu_u^2\frac{r}{n}\|\boldsymbol{M}\boldsymbol{B}^*\|_F^2 \leq 2p\mu_u^2\frac{r}{n}\|\boldsymbol{M}\|_F^2\|\boldsymbol{B}^*\|^2 \leq 2p\mu_u^2\frac{r}{n}\delta^2\sigma^{*2}_{\max}$$

and, proceeding in a very similar fashion,

$$\sigma_2^2 = \|\mathbb{E}[\sum_{jk}\boldsymbol{Z}_{jk}\boldsymbol{Z}_{jk}^\mathsf{T}]\| = p\|\sum_{jk}(\boldsymbol{M}^{j\mathsf{T}}\boldsymbol{b}_k^\star)^2\boldsymbol{u}_j\boldsymbol{u}_j^\mathsf{T}\|$$

$$\leq p\mu_u^2\frac{r}{n}\|\boldsymbol{M}\boldsymbol{B}^*\|_F^2 \leq 2p\mu_u^2\frac{r}{n}\|\boldsymbol{M}\|_F^2\|\boldsymbol{B}^*\|^2 \leq 2p\mu_u^2\frac{r}{n}\delta^2\sigma^{*2}_{\max}$$

Thus, $\sigma^2 = \max(\sigma_1^2, \sigma_2^2) = \sigma_2^2$, for $r < q$. Setting $t = \epsilon p\delta\sigma^*_{\max}$, we have

$$\frac{t}{L} = \frac{\epsilon pn}{r\mu\mu_u}, \;\; \frac{t^2}{\sigma^2} = \frac{\epsilon^2 pn}{2r\mu_u^2} \tag{14}$$

Thus, by the Matrix-Bernstein inequality, and using $\mu \leq \mu_u$,

$$\|\boldsymbol{D}\| = \|\boldsymbol{D} - \mathbb{E}[\boldsymbol{D}]\| \leq \epsilon p\delta\sigma^*_{\max}, \text{w.p. greater than } 1 - \exp(\log q - c\frac{\epsilon^2 pn}{r\mu_u^2}) \tag{15}$$

5.3 Proof of Lemma 3

Let $\delta = \delta^{(t-1)}$. Writing $\boldsymbol{X}^\star = \boldsymbol{U}\boldsymbol{G} + (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{X}^\star$, and $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, we have $\|\boldsymbol{X}^\star - \boldsymbol{X}\|_F \leq \|\boldsymbol{B} - \boldsymbol{G}\|_F + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{U}^*\boldsymbol{B}^*\|_F \leq \|\boldsymbol{B} - \boldsymbol{G}\|_F + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{U}^*\|_F\|\boldsymbol{B}^*\| \leq c\delta\sigma^*_{\max} + \delta\sigma^*_{\max}$. For ii), using the bound on $\|\boldsymbol{B} - \boldsymbol{G}\|_F$, $\|\boldsymbol{B}\| = \|\boldsymbol{B} - \boldsymbol{G} + \boldsymbol{G}\| \leq \|\boldsymbol{B} - \boldsymbol{G}\| + \|\boldsymbol{G}\| \leq \delta\sigma^*_{\max} + \sigma^*_{\max}$. For iii), $\sigma_{\min}(\boldsymbol{B}) \geq \sigma_{\min}(\boldsymbol{G}) - \sigma_{\max}(\boldsymbol{B} - \boldsymbol{G}) \geq \sqrt{1 - \delta^2}\sigma^*_{\min} - \delta\sqrt{r}\sigma^*_{\max}$. Here we used $\sigma_{\min}(\boldsymbol{G}) = \sigma_{\min}(\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^\star\boldsymbol{B}^\star) \geq \sigma_{\min}(\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^*)\sigma^*_{\min} \geq \sqrt{1 - \delta^2}\sigma^*_{\min}$.

5.4 Proof of Lemma 4

Since $\boldsymbol{b}_k = (\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k)^{-1}\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k\boldsymbol{b}_k^\star$,

$$\|\boldsymbol{b}_k\| \leq \|(\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k)^{-1}\| \cdot \|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k\| \cdot \|\boldsymbol{b}_k^\star\|$$

By Lemma C.6 of [Netrapalli et al., 2013]:

$$\|(\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k)^{-1}\| \leq \frac{1}{(1-\epsilon)p} \quad \text{w.p. at least } 1 - \exp(\log r - \epsilon^2 pn/\mu_u^2 r) \tag{16}$$

To bound $\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k\|$, first note that

$$\mathbb{E}[\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k] = p\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^\star$$

Next we bound $\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k - p\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^\star\|$ using matrix Bernstein inequality. Recall the expression for $\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k$ from Sec. 4.1. Let $\boldsymbol{Z}_j = (\xi_{jk} - p)\boldsymbol{u}^j\boldsymbol{u}^{*j\mathsf{T}}$. As done in earlier proofs, we can show that

$$L = \max_j \|\boldsymbol{Z}_j\| \leq \mu\mu_u\sqrt{r/n}, \quad \text{and}$$

$$\sigma^2 = \max(\|\sum_j \mathbb{E}[\boldsymbol{Z}_j\boldsymbol{Z}_j^\mathsf{T}]\|, \|\sum_j \mathbb{E}[\boldsymbol{Z}_j^\mathsf{T}\boldsymbol{Z}_j]\|) \leq 2p\mu_u^2\frac{r}{n}.$$

For this, we used the fact that $\mathbb{E}[(\xi_{jk}-p)^2] = 2p(1-p) \leq 2p$, $\|\sum_j \|\boldsymbol{u}^*{}_j\|^2\boldsymbol{u}_j\boldsymbol{u}_j^\mathsf{T}\| \leq \mu^2(r/n)\|\boldsymbol{U}\| = \mu^2(r/n)$ and $\|\sum_j \|\boldsymbol{u}^j\|^2\boldsymbol{u}^*{}_j\boldsymbol{u}^*{}_j^\mathsf{T}\| \leq \mu_u^2(r/n)\|\boldsymbol{U}^\star\| = \mu_u^2(r/n)$ and $\mu \leq \mu_u$.

Thus, by matrix Bernstein, $\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k - p\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^*\| \leq \epsilon p$ w.p. at least $1 - \exp(\log r - \epsilon^2 pn/\mu_u^2 r)$. Hence, with this probability,

$$\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k\| \leq \|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}^\star{}_k - \boldsymbol{U}^\mathsf{T}\boldsymbol{U}^\star\| + p\|\boldsymbol{U}^\mathsf{T}\boldsymbol{U}^\star\| \leq \epsilon p + p = (1+\epsilon)p \tag{17}$$

Thus, letting $\epsilon = 0.1$, w.p. at least $1 - \exp(\log r - cpn/\mu_u^2 r)$, $\|\boldsymbol{b}_k\| \leq 1.1\|\boldsymbol{b}_k^\star\| \leq 1.1\mu\sqrt{r/q}\sigma_{\max}^*$. Here we obtained a bound on $\|\boldsymbol{b}_k\|$ for a given k. By union bound, the above bound holds for all $k \in [q]$ w.p. at least $1 - q\exp(\log r - cpn/\mu_u^2 r) = 1 - \exp(\log q + \log r - cpn/\mu_u^2 r)$.

5.5 Proof of Lemma 5

Let $\delta = \delta^{(t-1)}$. Recall that $\text{GradU} = \sum_{jk}\xi_{jk}\boldsymbol{e}_j(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^*)\boldsymbol{b}_k^\mathsf{T}$. Using the lemma assumption, $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta\sigma_{\max}^*$. Using this and Lemma 3, $\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F \leq 2\delta\sigma_{\max}^*$, $\|\boldsymbol{B}\| \leq 1.1\sigma_{\max}^*$, We use these bounds in the proof below.

Observe that

$$\mathbb{E}[\text{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\mathsf{T}$$

Thus, using the assumed $\boldsymbol{B} - \boldsymbol{G}$ bound,

$$\|\mathbb{E}[\text{GradU}]\|_F \leq p\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F\|\boldsymbol{B}\| \leq 2.2p\delta\sigma_{\max}^{*2}$$

Next we bound the deviation using matrix Bernstein inequality. Writing $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, $\boldsymbol{X}^\star = \boldsymbol{U}\boldsymbol{G} + (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{X}^\star$, and using $n \leq q$

$$\max_{jk}|\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^*| = |\boldsymbol{e}_j^\mathsf{T}(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{e}_k|$$

18

$$\leq \|\boldsymbol{e}_j^\mathsf{T}\boldsymbol{U}\|\|\boldsymbol{B}-\boldsymbol{G}\| + \|(\boldsymbol{I}-\boldsymbol{U}\boldsymbol{U}^\mathsf{T})\boldsymbol{U}^\star\|\|\boldsymbol{B}^\star\boldsymbol{e}_k\|$$
$$\leq \mu_u\sqrt{r/n}\delta\sigma_{\max}^* + \mu\sqrt{r/q}\delta\sigma_{\max}^* \leq 2\mu_u\sqrt{r/n}\delta\sigma_{\max}^*$$

Using this,

$$L = \max_{jk}\|(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)\boldsymbol{b}_k^\mathsf{T}\| \leq \max_{jk}\|(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)\|\max_k\|\boldsymbol{b}_k\| \leq 2\mu_u\sqrt{r/n}\delta\sigma_{\max}^*\cdot\mu\sqrt{r/q}\sigma_{\max}^* \leq 2\mu_u\mu(r/n)\delta\sigma_{\max}^{*2}$$

To bound $\sigma_1^2 = \|\sum_{jk}\mathbb{E}[\boldsymbol{Z}_{jk}\boldsymbol{Z}_{jk}^\mathsf{T}]\|$ and $\sigma_2^2 = \|\sum_{jk}\mathbb{E}[\boldsymbol{Z}_{jk}^\mathsf{T}\boldsymbol{Z}_{jk}]\|$, where $\boldsymbol{Z}_{jk} = \xi_{jk}\boldsymbol{e}_j(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^*)\boldsymbol{b}_k^\mathsf{T}$, using $\mathbb{E}[(\xi_{jk}-p)^2] = 2p(1-p) \leq 2p$ we have

$$\sigma_1^2 = 2p\|\sum_{jk}(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)^2\boldsymbol{e}_j\boldsymbol{b}_k^\mathsf{T}\boldsymbol{b}_k\boldsymbol{e}_j^\mathsf{T}\| \leq 2p\|\boldsymbol{b}_k\|^2\|\boldsymbol{X}-\boldsymbol{X}^\star\|_F^2 \leq 2p\mu^2(r/q)\sigma_{\max}^{*2}\cdot(\delta\sigma_{\max}^*)^2 = 2p\mu^2(r/q)\delta^2\sigma_{\max}^{*4}$$

$$\sigma_2^2 = 2p\|\sum_{jk}(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)^2\boldsymbol{e}_j^\mathsf{T}\boldsymbol{e}_j\boldsymbol{b}_k\boldsymbol{b}_k^\mathsf{T}\| \leq 2p\|\boldsymbol{b}_k\|^2\|\boldsymbol{X}-\boldsymbol{X}^\star\|_F^2 = \sigma_1^2$$

Setting $t = \epsilon p\delta\sigma_{\min}^{*2}$, we have

$$\frac{t^2}{\sigma^2} = c\frac{\epsilon^2 p^2\delta^2\sigma_{\min}^{*}{}^4}{p\mu^2(r/q)\delta^2\sigma_{\max}^{*4}} = \frac{\epsilon^2 pq}{\kappa^4\mu^2 r}, \quad \frac{t}{L} = c\frac{\epsilon p\delta\sigma_{\min}^{*2}}{\mu_u\mu(r/n)\delta\sigma_{\max}^{*2}} = \frac{\epsilon pn}{\kappa^2\mu_u\mu r}$$

Thus,

$$\min\left(\frac{t^2}{\sigma^2}, \frac{t}{L}\right) = c\frac{\epsilon^2 pn}{\max(\kappa^4\mu^2, \kappa^2\mu_u\mu)r}$$

and so, by matrix Bernstein, w.p. at least $1-\exp(\log q - c\frac{\epsilon^2 pn}{\max(\kappa^4\mu^2, \kappa^2\mu_u\mu)r})$, $\|\mathrm{GradU}-\mathbb{E}[\mathrm{GradU}]\| \leq \epsilon p\delta\sigma_{\min}^{*2}$.

By setting $\epsilon = \epsilon_1/\sqrt{r}$, $\|\mathrm{GradU} - \mathbb{E}[\mathrm{GradU}]\|_F \leq \sqrt{r}\|\mathrm{GradU} - \mathbb{E}[\mathrm{GradU}]\| \leq \epsilon_1 p\delta\sigma_{\min}^{*2}$ w.p. at least $1 - \exp(\log q - \frac{\epsilon_1^2 pn}{\max(\kappa^4\mu^2, \kappa^2\mu_u\mu)r^2})$.

5.6 Proof of Lemma 6

Let $\delta = \delta^{(t-1)}$. This is used only for proving the second part.

Recall from Sec. 4.1 that

$$\mathrm{Grad}\boldsymbol{u}^j = \sum_k \xi_{jk}(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)\boldsymbol{b}_k^\mathsf{T}$$

To apply matrix Bernstein, we need to bound $\sigma_1^2 \equiv \|\mathbb{E}[\sum_k \xi_{jk}^2(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^*)^2\boldsymbol{b}_k^\mathsf{T}\boldsymbol{b}_k]\|$, $\sigma_2^2 \equiv \|\mathbb{E}[\sum_k \xi_{jk}^2(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^*)^2\boldsymbol{b}_k\boldsymbol{b}_k^\mathsf{T}]\|$, and $L = \max_k\|(\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star)\boldsymbol{b}_k\|$. Using

$$|\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star| \leq 2\max(|\boldsymbol{x}_{jk}|, |\boldsymbol{x}_{jk}^\star|) \leq 2\max(\|\boldsymbol{u}^j\|\|\boldsymbol{b}_k\|, \|\boldsymbol{u}^{*j}\|\|\boldsymbol{b}_k^\star\|) \leq 2\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\max(\|\boldsymbol{b}_k\|, \|\boldsymbol{b}_k^\star\|)$$

and $\max(\|\boldsymbol{b}_k^\star\|, \|\boldsymbol{b}_k\|) \leq 1.1\mu\sqrt{r/q}\sigma_{\max}^*$,

$$L \leq \max_k|\boldsymbol{x}_{jk}-\boldsymbol{x}_{jk}^\star|\max_k\|\boldsymbol{b}_k\| \leq 2\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\max_k\max(\|\boldsymbol{b}_k\|, \|\boldsymbol{b}_k^\star\|)\max_k\|\boldsymbol{b}_k\|$$
$$\leq 2\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\mu^2(r/q)\sigma_{\max}^{*2} \tag{18}$$

We use the above seemingly loose bound because we do not need $\delta^{(t-1)}$ in this bound. Instead we need to the bound to be of the form $\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^*{}_j\|)$ times a factor of $\sqrt{r/q}\sigma_{\max}^*$

19

The variance $\sigma_1^2 = \|\mathbb{E}[\sum_k \xi_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^*)^2 \boldsymbol{b}_k^\mathsf{T} \boldsymbol{b}_k]\|$ can be bounded using similar ideas as follows.

$$\sigma_1^2 = \| \sum_k p(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star)^2 \boldsymbol{b}_k^\mathsf{T} \boldsymbol{b}_k \| \leq 2p\max(\|\boldsymbol{u}^j\|_2, \|\boldsymbol{u}^{*j}\|_2)^2 \max_k(\|\boldsymbol{b}_k\|, \|\boldsymbol{b}_k^*\|)^2 \sum_k \|\boldsymbol{b}_k\|_2^2$$

$$\leq 2p\max(\|\boldsymbol{u}^j\|_2, \|\boldsymbol{u}^{*j}\|_2)^2 \mu^2 (r^2/q)\sigma_{\max}^{*4} \tag{19}$$

Similarly, $\sigma_2^2 \leq 2p\max(\|\boldsymbol{u}^j\|_2, \|\boldsymbol{u}^{*j}\|_2)^2 \mu^2 (r^2/q)\sigma_{\max}^{*4}$. By the matrix Bernstein inequality with $t = \epsilon p \max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\sigma_{\min}^{*2}$, and noting that $\sigma^2 = \max(\sigma_1^2, \sigma_2^2) = \sigma_1^2$, we have w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu^2 \kappa^4 r^2)$,

$$\|\mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j] - \mathrm{Grad}\boldsymbol{u}^j\| \leq \epsilon p \max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\sigma_{\min}^{*2} \tag{20}$$

This completes the proof for the first part of the lemma.

Proof of second part In line 7 of Algorithm 1, adding/subtracting $\mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j] = p(\boldsymbol{u}^j \boldsymbol{B}\boldsymbol{B}^\mathsf{T} - \boldsymbol{u}^{*j}\boldsymbol{B}^\star \boldsymbol{B}^\mathsf{T})$,

$$\widetilde{\boldsymbol{u}}^{j+} = \boldsymbol{u}^j - \eta\mathrm{Grad}\boldsymbol{u}^j = \boldsymbol{u}^j(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\mathsf{T}) + \eta p\boldsymbol{u}^{*j}\boldsymbol{B}^\star \boldsymbol{B}^\mathsf{T} + \eta(\mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j] - \mathrm{Grad}\boldsymbol{u}^j) \tag{21}$$

Since we assumed $\delta \leq c/\kappa$, by Lemma 3, $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma_{\min}^*$ and $\sigma_{\max}(\boldsymbol{B}) \leq 1.1\sigma_{\max}^*$. Thus, if $\eta < 0.5/p\sigma_{\max}^{*2}$ then, $\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\mathsf{T}$ is positive semi-definite (psd) and so $\|\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\mathsf{T}\| = \lambda_{\max}(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\mathsf{T}) = 1 - \eta p\sigma_{\min}^2(\boldsymbol{B}) \leq 1 - 0.9\eta p\sigma_{\min}^{*\,2}$.

Thus, using the above bound on $\|\mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j] - \mathrm{Grad}\boldsymbol{u}^j\|$, if $\eta < 0.5/p\sigma_{\max}^{*2}$,

$$\|\widetilde{\boldsymbol{u}}^{j+}\| \leq \|\boldsymbol{u}^j\|(1 - 0.9\eta p\sigma_{\min}^{*2}) + \eta p\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2} + \epsilon\eta p\sigma_{\min}^{*2}\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)$$

$$\leq (1 - (0.9 - \epsilon)\eta p\sigma_{\min}^{*2})\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|) + \eta p\sigma_{\max}^{*2}\|\boldsymbol{u}^{*j}\| \tag{22}$$

w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu^2 \kappa^4 r^2)$. We bound $\|\boldsymbol{u}^{j(t+1)}\| \leq \|(\boldsymbol{R}^+)^{-1}\| \cdot \|\widetilde{\boldsymbol{u}}^{j(t+1)}\|$, where $\widetilde{\boldsymbol{U}}^+ \overset{\mathrm{QR}}{=} \boldsymbol{U}^+ \boldsymbol{R}^+$ next. Using Lemma 5,

$$\|(\boldsymbol{R}^+)^{-1}\| = \frac{1}{\sigma_{\min}(\boldsymbol{U} - \eta\|\mathrm{GradU}\|)} \leq \frac{1}{1 - \eta 2.1 p\delta\sigma_{\max}^{*2}} \leq \frac{1}{1 - 0.25\eta p\sigma_{\max}^{*2}} \leq 1 + 0.5\eta p\sigma_{\max}^{*2} \tag{23}$$

w.p. given in Lemma 5. Thus, using (22) and (23) in $\|\boldsymbol{u}^{j+}\| \leq \|(\boldsymbol{R}^+)^{-1}\| \cdot \|\widetilde{\boldsymbol{u}}^{j(t+1)}\|$,

$$\|\boldsymbol{u}^{j+}\| \leq (1 + 0.5\eta p\sigma_{\max}^{*2})(1 - (0.9 - \epsilon)\eta p\sigma_{\min}^{*2})\max(\|\boldsymbol{u}^j\|\|\boldsymbol{u}^{*j}\|) + (1 + 0.5\eta p\sigma_{\min}^{*2})\eta p\sigma_{\max}^{*2}\|\boldsymbol{u}^{*j}\|$$

$$\leq (1 - (0.4 - \epsilon)\eta p\sigma_{\min}^{*2})\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|) + (1 + 0.5\eta p\sigma_{\max}^{*2})\eta p\sigma_{\max}^{*2}\|\boldsymbol{u}^{*j}\|$$

$$\leq (1 - (0.4 - \epsilon)\eta p\sigma_{\min}^{*2})\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|) + (1 + 0.25/\kappa^2)0.5\|\boldsymbol{u}^{*j}\|$$

$$\leq (1 - \frac{0.15}{\kappa^2})\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|) + 0.7\|\boldsymbol{u}^{*j}\|$$

where the last bound follows by setting $\eta = 0.5/p\sigma_{\max}^{*2}$ and setting $\epsilon = 0.1$ in (20). Thus, we have shown that if $\eta = 0.5/p\sigma_{\max}^{*2}$, and if $\delta \leq c/\kappa^2$, with probability exceeding $1 - 4/n^3$,

$$\|\boldsymbol{u}^{j+}\| \leq (1 - \frac{0.15}{\kappa^2})\max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|) + 0.7\|\boldsymbol{u}^{*j}\|$$

$$\leq (1 - \frac{0.15}{\kappa^2})\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\|$$

20

6 Guarantee for Noisy LRMC

Consider the noisy LRMC problem defined as follows. We observe

$$\boldsymbol{Y} := \boldsymbol{X}^\star_\Omega + \boldsymbol{W}_\Omega$$

We do not make any assumption on the noise $\boldsymbol{W}$ (deterministic $\boldsymbol{W}$). Unlike the standard Gaussian assumption, this assumption is weaker: it allows the noise entries to be anything, all entries could be positive too for example.

    We can show the following by borrowing the overall proof approach for modifying a noise-free case guarantee for an iterative algorithm from past works [Chen & Candes, 2015, Nayer & Vaswani, 2021] (the latter used a similar approach for analyzing the AltMin algorithm for LR phase retrieval).

Corollary 1. *Let*

$$\epsilon_{noise} := \max_{jk} \frac{|w_{jk}|}{|x^*_{jk}|}$$

In the setting of Theorem 2.2, if $\epsilon_{noise} \leq \frac{c}{\sqrt{r}\kappa^3}$, then, by setting $T = C\kappa^2 \log(1/\epsilon_{noise})$, we can guarantee that

$$\mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^\star) \leq \kappa^2 \sqrt{r}\epsilon_{noise}$$

and $\|\boldsymbol{X}^{(T)} - \boldsymbol{X}^\star\|_F \leq \mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^\star)\sigma^*_{\max}$. In general, for any ϵ, by setting $T = C\kappa^2 \log(1/\epsilon)$, we can ensure that

$$\mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^\star) \leq \max(\epsilon, \kappa^2 \sqrt{r}\epsilon_{noise})$$

and $\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F \leq \mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^\star)\sigma^*_{\max}$.

Proof. Corollary 1 extends our noise-free case proof to the noisy case using the following simple ideas. First we assume that the noise level is small enough so that accurate initialization is possible; this requires noise level, $\epsilon_{noise}\sqrt{r}$, to be of order $c\delta^{(0)}\sigma^*_{\min}$ for our required value of $\delta^{(0)} = c/\kappa^2$. This helps ensure that the initialization error is bounded by $\delta^{(0)}$.

    Next, at each iteration, we attempt to bound terms and prove incoherence for the updated $\boldsymbol{U}, \boldsymbol{B}$, in order to guarantee error decay similar to the noise-free case. For this, we need the noise level to be such that (i) the error in recovering $\boldsymbol{B}^\star$ is of the same order as in the noise-free case; and (ii) the same is true for the bounds on the gradient norm. Both of these are ensured if $\epsilon_{noise}\kappa^2\sqrt{r} \leq c\delta^{(t-1)}$ for a $c < 1$, e.g., $c = 0.1$. Details are given in Appendix B. $\qquad\square$

7 Improved Guarantees for AltMin and Smooth-AltMin

Using the same lemmas used to prove the AltGDmin guarantee, we are also able to improve the guarantee for AltMin and Smooth-AltMin [Hardt, 2014], as long as both are initialized as given in Algorithm 1: the initial estimate of $\boldsymbol{U}$ is clipped using row-wise clipping. We borrow this clipping idea from [Yi et al., 2016].

7.1 Improving the result for AltMin

Corollary 2 (Improved AltMin [Netrapalli et al., 2013] Guarantee)**.** *Consider the AltMin algorithm initialized using our initialization (lines 2-4) of Algorithm 1. Assume that Assumption 1*

holds and that, at each iteration t, entries of $\boldsymbol{X}^\star$ are observed independently with probability p satisfying

$$np > C\kappa^4\mu^2r^2\log q \tag{24}$$

Then, w.p. at least $1-3/n^3$, the iterates $\boldsymbol{U}^{(t)}$ of AltMin [Netrapalli et al., 2013] satisfy $\mathrm{SD}_F(\boldsymbol{U}^{(t+1)}, \boldsymbol{U}^\star) \leq 0.25\mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*)$. Consequently, if $p_{tot} > \frac{C\kappa^4\mu^2r^2\log q}{n}\log(1/\epsilon)$, then, after $T = C\log(1/\epsilon)$ iterations, $\mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^\star) \leq \epsilon$

We prove this in Appendix C. The proof is an easy corollary of Lemmas 1, 2, 3 and 4. Comparing (24) to Theorem 2.5 of [Netrapalli et al., 2013], we observe that the sample complexity for AltMinComplete has reduced in r from $r^{4.5}\log r$ to r^2.

7.2 Improved Guarantee for Smooth AltMin [Hardt, 2014]

We will reduce the overall sample complexity by decreasing p_{init} to $O(r^2)$ below. Formally,

Corollary 3. Let $r, \epsilon > 0$. Let $\boldsymbol{X}^\star$ be a symmetric $n \times n$ matrix. Then, for $T = O(\log(n/\epsilon))$, the output of SAltLS satisfies $\mathrm{SD}_F(\boldsymbol{U}, \boldsymbol{U}^*) \leq \epsilon$ with probability $9/10$, provided that sampling probability $p \geq p_{init} + p_{LS}$, where

$$p_{\mathrm{init}} \geq C\kappa^2r^2\mu\log n/n, \quad p_{\mathrm{LS}} \geq C\mu^2\kappa^2r^2\log(n/\epsilon)\log^2 n/n \tag{25}$$

This is a direct corollary of Theorem 6.1 of [Hardt, 2014] combined with our initialization lemma, Lemma 1.

8 Conclusions

In this paper, we developed and analyzed the Alternating Gradient Minimization (AltGDMin) algorithm for solving the low-rank matrix completion (LRMC) problem, with a specific focus on federated learning settings. AltGDMin is designed to efficiently recover an $n \times q$ rank-r matrix $X^\star$ from a subset of its entries, particularly when $r \ll \min(n,q)$, while minimizing communication between distributed nodes and a central server. Our theoretical results include iteration and sample complexity bounds, demonstrating that AltGDMin is among the most communication-efficient and fastest methods for federated LRMC.

Moreover, we extended the analysis to noisy LRMC, providing robust convergence guarantees. Numerical experiments validated the superior performance of AltGDMin over both centralized and federated approaches, reinforcing its ability to leverage the low-dimensional structure of the data to accurately recover missing entries. Additionally, we used our theoretical framework to improve the sample complexity guarantee for AltMin, which remains the fastest centralized solution for LRMC.

Competing Interest Declaration

I declare no conflict of interests.

References

[Abbasi et al., 2023] 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).: IEEE.

[Achlioptas & McSherry, 2007] Achlioptas, D. & McSherry, F. (2007). Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2), 9–es.

[Anelli et al., 2022] Anelli, V. W., Deldjoo, Y., Di Noia, T., Ferrara, A., & Narducci, F. (2022). User-controlled federated matrix factorization for recommender systems. *Journal of Intelligent Information Systems*, 58(2), 287–309.

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

[Candès & Recht, 2009] Candès, E. J. & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 6(9), 717–772.

[Candès & Tao, 2010] Candès, E. J. & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE transactions on information theory*, 56(5), 2053–2080.

[Chen & Candes, 2015] Chen, Y. & Candes, E. (2015). Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Neur. Info. Proc. Sys. (NeurIPS)* (pp. 739–747).

[Chen & Chi, 2018] Chen, Y. & Chi, Y. (2018). Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. *IEEE Signal Processing Magazine*, 35(4), 14–31.

[Chen et al., 2020] Chen, Y., Chi, Y., Fan, J., & Ma, C. (2020). Spectral methods for data science: A statistical perspective. *arXiv preprint arXiv:2012.08496*.

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

[Cherapanamjeri et al., 2016] Cherapanamjeri, Y., Gupta, K., & Jain, P. (2016). Nearly-optimal robust matrix completion. *ICML*.

[Davenport & Romberg, 2016] Davenport, M. A. & Romberg, J. (2016). An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4), 608–622.

[Fazel, 2002] Fazel, M. (2002). Matrix rank minimization with applications. *PhD thesis, Stanford Univ.*

[Gross, 2011] Gross, D. (2011). Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3), 1548–1566.

[Hardt, 2014] Hardt, M. (2014). Understanding alternating minimization for matrix completion. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science* (pp. 651–660).: IEEE.

[Hardt & Wootters, 2014] Hardt, M. & Wootters, M. (2014). Fast matrix completion without the condition number. In *Conf. on Learning Theory*.

[He et al., 2019] 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).: IEEE.

[Jain & Netrapalli, 2015] Jain, P. & Netrapalli, P. (2015). Fast exact matrix completion with finite samples. In *Conf. on Learning Theory* (pp. 1007–1034).

[Keshavan et al., 2010a] Keshavan, R., Montanari, A., & Oh, S. (2010a). Matrix completion from a few entries. *IEEE Trans. Info. Th.*, 56(6), 2980–2998.

[Keshavan et al., 2010b] Keshavan, R. H., Montanari, A., & Oh, S. (2010b). Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6), 2980–2998.

[Li et al., 2021] Li, Z., Ding, B., Zhang, C., Li, N., & Zhou, J. (2021). Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment*, 15(4).

[Lin & Ling, 2015] Lin, A.-Y. & Ling, Q. (2015). Decentralized and privacy-preserving low-rank matrix completion. *Journal of the Operations Research Society of China*, 3(2), 189–205.

[Ling et al., 2012] Ling, Q., Xu, Y., Yin, W., & Wen, Z. (2012). Decentralized low-rank matrix completion. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2925–2928).

[Ma et al., 2018] Ma, C., Wang, K., Chi, Y., & Chen, Y. (2018). Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution. In *Intl. Conf. Machine Learning (ICML)*.

[Mackey et al., 2015] Mackey, L. W., Talwalkar, A., & Jordan, M. I. (2015). Distributed matrix completion and robust factorization. *J. Mach. Learn. Res.*, 16(1), 913–960.

[Mardani et al., 2013] Mardani, M., Mateos, G., & Giannakis, G. (2013). Decentralized sparsity-regularized rank minimization: Algorithms and applications. *IEEE Trans. Sig. Proc.*

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

[Nayer et al., 2020] Nayer, S., Narayanamurthy, P., & Vaswani, N. (2020). Provable low rank phase retrieval. *IEEE Trans. Info. Th.*

[Nayer & Vaswani, 2021] Nayer, S. & Vaswani, N. (2021). Sample-efficient low rank phase retrieval. *IEEE Trans. Info. Th.*

[Nayer & Vaswani, 2023a] Nayer, S. & Vaswani, N. (2023a). Efficient federated low rank matrix recovery via alternating gd and minimization: A simple proof. *IEEE Trans. Info. Th.*

[Nayer & Vaswani, 2023b] Nayer, S. & Vaswani, N. (2023b). Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections. *IEEE Trans. Info. Th.*

[Netrapalli et al., 2013] Netrapalli, P., Jain, P., & Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Annual ACM Symp. on Th. of Comp. (STOC)*.

[Sun & Luo, 2016] Sun, R. & Luo, Z.-Q. (2016). Guaranteed matrix completion via non-convex factorization. *IEEE Trans. Info. Th.*, 62(11), 6535–6579.

[Teflioudi et al., 2012] Teflioudi, C., Makari, F., & Gemulla, R. (2012). Distributed matrix completion. In *2012 ieee 12th international conference on data mining* (pp. 655–664).: IEEE.

[Vershynin, 2018] Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press.

[Yi et al., 2016] Yi, X., Park, D., Chen, Y., & Caramanis, C. (2016). Fast algorithms for robust pca via gradient descent. In *Neur. Info. Proc. Sys. (NeurIPS)*.

[Zarantonello, 1971] Zarantonello, E. H. (1971). Projections on convex sets in hilbert space and spectral theory: Part i. projections on convex sets: Part ii. spectral theory. In *Contributions to nonlinear functional analysis* (pp. 237–424). Elsevier.

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

A Simulation Results

Experimental Setup   We plot averaged subspace distance at the iteration t against the average time taken until iteration t, with the averages being computed over 100 Monte Carlo runs. The averaging is over the observed entries which are generated uniformly at random. The matrix $\boldsymbol{X}^\star = \boldsymbol{U}^\star \boldsymbol{B}^\star$ was generated once: we let $\boldsymbol{U}^\star$ be an orthonormalized $n \times r$ random Gaussian matrix and $\boldsymbol{B}^\star$ an $r \times q$ random Gaussian matrix. We used the 'parfor' loop in MATLAB to distribute the computation across 10 workers, with each worker being an individual core of a multi-core CPU (13-th Gen. Intel Core i7 with 32GB RAM and 16 cores). To compute the left-singular vectors needed for the initialization of federated algorithms, we used the federated power method and performed 15 power iterations. Also, for both AltMin and AltGDMin, we do not sample-split, i.e., we run each iteration of the algorithms on all observed entries.

Step-size and other parameters   For FactGD (Centralized), we used the code provided by the authors of that work [Yi et al., 2016]. The step size was set $\eta_{\mathrm{FactGD}} = pc/\|\boldsymbol{Y}\|$, as also done in the authors' own implementation of their algorithm. Setting $c = 0.75$ showed the fastest convergence for our simulations (see Fig. 1a). For AltMin (Fed./NotPrvt.) and AltMin(Fed./Prvt.), we wrote our own MATLAB implementation; the latter uses GD to solve the LS problem for updating $\boldsymbol{U}$. We set the number of GD iterations for solving each LS problem to 10 with step size $\eta = p/\|\boldsymbol{Y}\|^2$, which showed the fastest convergence. Note that the $\boldsymbol{U}$-update least-squares problem is convex, and the chosen step-size is an upper bound on the Lipschitz constant of the expected gradient of the objective function at $t = 0$ [Abbasi et al., 2023]. AltGDMin is also implemented with step-size $\eta = p/\|\boldsymbol{Y}\|^2$, which approximates the step size choice suggested by Theorem 2.2, since $\|\mathbb{E}[\boldsymbol{Y}]\|^2 = p^2 \sigma_{\max}^{*2}$. AltMin and AltGDMin were both initialized by lines 2-4 of Algorithm 1. The incoherence parameter μ was estimated as $\widehat{\mu} = \arg\min_\mu \|\boldsymbol{u}^{(0)j}\| \leq \mu\sqrt{r/n}$ for all $j \in [n]$.

Observations   Federated algorithms are faster than their centralized counterparts, except AltMin (Fed. Prvt.). The proposed algorithm AltGDMin (Fed.) is faster than all benchmark methods, especially at the higher value of $r = 20$. Specifically, *AltGDMin (Fed.) converges to $\boldsymbol{U}^\star$ in approximately 12 seconds, compared to nearly 20 seconds for the second fastest AltMin (Fed. NotPrvt.).* We interpret/explain these observations and the benchmark methods below.

AltGDMin   AltGDMin (Fed.) has faster convergence because of low communication complexity and low time complexity at the center. The upstream communication complexity, $\min(\sum_{k \in \boldsymbol{S}_\ell} \nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{b}_k), nr) = nr$, is low because the nodes sum the column-wise gradients

FactGD $\nabla_{\boldsymbol{U}} f = \nabla_{\boldsymbol{X}} \boldsymbol{B}^\intercal \in \mathbb{R}^{n \times r}$ and $\boldsymbol{B}\boldsymbol{B}^\intercal \in \mathbb{R}^{r \times r}$ are computed at the nodes and transmitted upstream. The total upstream communication cost is $nr + r^2$, higher than the nr communication cost of AltGDMin. $\boldsymbol{B}$ is updated locally (GD iteration and normalizing) at the nodes but this requires two data exchanges with the center. This is because the gradient of $\boldsymbol{B}$, $(\boldsymbol{B}\boldsymbol{B}^\intercal - \boldsymbol{U}^\intercal\boldsymbol{U})\boldsymbol{B}$, with respect to the norm balancing term, $\|\boldsymbol{U}^\intercal\boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^\intercal\|_F^2$, cannot be computed at the nodes. The nodes compute and transmit $\sum_{k \in \mathcal{S}_\ell} \boldsymbol{b}_k \boldsymbol{b}_k^\intercal$, which are summed at the center to form $\boldsymbol{B}\boldsymbol{B}^\intercal$, and transmitted back to the nodes. The center also computes and transmits $\boldsymbol{U}^\intercal\boldsymbol{U}$. Then, at the nodes, the partial gradient $(\boldsymbol{B}\boldsymbol{B}^\intercal - \boldsymbol{U}^\intercal\boldsymbol{U})\sum_{k \in \mathcal{S}_\ell} \boldsymbol{b}_k \boldsymbol{b}_k^\intercal$ is computed, followed by column-normalizing. The federated and centralized implementations of FactGD do not use 'for' loops at all, either at the nodes or at the center. But, FactGD is slower than AltMin and AltGDMin because of its higher

communication complexity, $O(nr + r^2)$ and 2 data exchanges, compared to 1 data exchange and $O(nr)$, $O(qr)$ communication complexity for AltGDMin and AltMin (Fed.NotPrvt.), respectively.

AltMin AltMin (Fed./NotPrvt.) is slower than AltGDMin (Fed.) because the n $\boldsymbol{U}$-update LS problems are solved sequentially at the center with complexity $|\Omega|r^2$, compared to the nr^2 complexity of computing the QR decomposition for AltGDMin. Also, AltMin (Fed./NotPrvt.) is not private because the nodes communicate the updated $\boldsymbol{b}_k^{(t+1)}$ to the center. For AltMin(Fed./Prvt.), the $\boldsymbol{U}$-update LS problems are solved by multiple gradient descent iterations at the node. While private, the GD version of AltMin is slow because of the communication overhead of transmitting/receiving the gradients several times ($\log(1/\epsilon)$ times for ϵ-accuracy) in each iteration. AltMin (Cntrl.) is slower than federated methods because both $\boldsymbol{U}$, $\boldsymbol{B}$ LS problems are solved sequentially by the closed form solution at the center, but it is still the fastest centralized algorithm overall, possibly because of $\log(1/\epsilon)$ iteration complexity, which is lower than that of other methods.

B Proof of the Corollary 1 (noisy LRMC)

We can prove the following by directly modifying the noise-free case proof.

Claim 4. *Let the observed matrix $\boldsymbol{Y} = \boldsymbol{X}_\Omega^* + \boldsymbol{W}_\Omega$ be corrupted by additive noise $\boldsymbol{W}$ such that $|\boldsymbol{W}_{jk}| \leq \epsilon_{noise}|\boldsymbol{X}_{jk}^*|$. Assuming also that $p \geq C\kappa^6 r^2 \mu \log q \log(1/\epsilon)/n$ and $\epsilon_{noise} \leq 1/(\sqrt{r}\kappa^3)$, then, after $T = \kappa^2 \log(1/\epsilon_{noise})$ iterations, $\mathrm{SD}_F(\boldsymbol{U}^{(T)}, \boldsymbol{U}^*) \leq \kappa^2 \epsilon_{noise}\sqrt{r}$.*

B.1 Proof

Restricting $\epsilon_{noise} \leq 1/(\sqrt{r}\kappa^3)$ (as done in the statement of Theorem 4) ensures that the initialization bound in Lemma 1 differs only by a constant factor (see Lemma 7).

With noise $\boldsymbol{W}_\Omega$, the Least-Squares update (4) changes to

$$\boldsymbol{b}_k - \boldsymbol{g}_k = \boldsymbol{F}_k^{-1}(\boldsymbol{d}_k + \boldsymbol{U}_k^\mathsf{T}\boldsymbol{w}_k), \text{ for all } k \in [q]$$

where $\boldsymbol{w}_k$ is the k-th column of $\boldsymbol{W}_\Omega$. Consequently, the bound in (9) changes to $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \|\boldsymbol{F}^{-1}\|(\|\boldsymbol{D}\|_F + \|\boldsymbol{W}_B\|_F)$, where $\boldsymbol{W}_B = \sum_k \boldsymbol{U}_k^\mathsf{T}\boldsymbol{w}_k\boldsymbol{e}_k^\mathsf{T}$. In Lemma 8, we bound $\|\boldsymbol{W}_B\|_F \leq 2\sqrt{r}\epsilon_{noise}p\sigma_{\max}^*$, ensuring that $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \max(\delta^{(t-1)}, 2\sqrt{r}\epsilon_{noise})\sigma_{\max}^*$. Assuming $2\sqrt{r}\epsilon_{noise} \leq \delta^{(t-1)}$, Lemmas 2 and 3 continue to hold. Because $\|\boldsymbol{U}_k^\mathsf{T}\boldsymbol{w}_k\| \leq 2\epsilon_{noise}p\|\boldsymbol{b}_k^*\|_2$ (Lemma 8), $\|(\boldsymbol{U}_k^\mathsf{T}\boldsymbol{U}_k)^{-1}\boldsymbol{U}_k^\mathsf{T}\boldsymbol{w}_k\|_2 \leq 2\epsilon_{noise}\|\boldsymbol{b}_k^*\|_2$, which, together with $\|\boldsymbol{b}_k\|_2 \leq 1.1\sigma_{\max}^*\mu\sqrt{r/q}$ (Lemma 4), bounds the noisy least-squares update $\|\widetilde{\boldsymbol{b}}_k\|_2 \leq (1.1 + 2\epsilon_{noise})\mu\sigma_{\max}^*\sqrt{r/q}$, thereby proving μ-incoherence in the noisy setting.

With noise, the gradient is $\widetilde{\mathrm{GradU}} = \mathrm{GradU} - \boldsymbol{W}_{\boldsymbol{U}}$, where $\boldsymbol{W}_{\boldsymbol{U}} = \boldsymbol{W}_\Omega\boldsymbol{B}^\mathsf{T}$. We bound $\|\boldsymbol{W}_{\boldsymbol{U}}\|_F \leq 2\epsilon_{noise}\sqrt{r}p\sigma_{\max}^{*2}$ in Lemma 8, and assume $\epsilon_{noise} \leq \delta^{(t-1)}/\sqrt{r}$ so that the bound in Lemma 5 continues to hold. To prove row-incoherence in the noisy setting, the proof of Lemma 6 needs to be only slightly modified. (20) still applies and (21) has the additional row-vector term $\eta p\boldsymbol{w}^j\boldsymbol{B}^{*\mathsf{T}}$, which can be bounded $\|\boldsymbol{w}^j\boldsymbol{B}^{*\mathsf{T}}\|_2 \leq \|\boldsymbol{w}^j\|_2\|\boldsymbol{B}^*\| \leq \epsilon_{noise}\|\boldsymbol{x}^{*j}\|_2\sigma_{\max}^* \leq \epsilon_{noise}\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2}$. This term contributes $\epsilon_{noise}\|\boldsymbol{u}^{*j}\|$ in the final bound, that is, $\|\widetilde{\boldsymbol{u}}^{+j}\|_2 \leq (1 - 0.15/\kappa^2)\|\boldsymbol{u}^j\|_2 + 2\|\boldsymbol{u}^{*j}\|_2 + \epsilon_{noise}\|\boldsymbol{u}^{*j}\|$, which ensures $\kappa^2\mu$ incoherence for the noisy gradient update $\widetilde{\boldsymbol{u}}^{+j}$, since $\epsilon_{noise} \leq 1$.

The proof details are the same as those for proving Theorem 2.2 given earlier in Sec. 4.3.

B.2 Noisy case lemmas

Our assumption $|\boldsymbol{W}_{jk}| \leq \epsilon_{noise}|\boldsymbol{X}^*_{jk}|$ also implies that

$$\max\left(\frac{\|\boldsymbol{w}_k\|}{\|\boldsymbol{x}^*_k\|}, \frac{\|\boldsymbol{w}^j\|}{\|\boldsymbol{x}^{*j}\|}\right) \leq \epsilon_{noise}, \text{ and } |\boldsymbol{W}_{jk}| \leq \epsilon_{noise}\mu^2\sqrt{r/q}\sqrt{r/n}\sigma^*_{\max} \qquad (26)$$

This fact is used in proving both lemmas below.

Lemma 7. *Let the observed matrix $\boldsymbol{Y} = \boldsymbol{X}^*_\Omega + \boldsymbol{W}_\Omega$ be corrupted by additive noise $\boldsymbol{W}$ such that $|\boldsymbol{W}_{jk}| \leq \epsilon_{noise}|\boldsymbol{X}^*_{jk}|$. Assuming also that $p \geq C\kappa^6 r^2\mu\log q/n$ and $\epsilon_{noise} \leq 1/\sqrt{r}\kappa^3$, then, $\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^*) \leq c/\kappa^2$, where $\boldsymbol{U}^{(00)} \in \mathbb{R}^{n\times r}$ are the left-singular vectors of $\boldsymbol{Y}$*

Proof. By Wedin's sin theta theorem (Frob norm version),

$$\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^\star) \leq C\frac{\|\boldsymbol{Y} - p\boldsymbol{X}^\star\|_F}{\sigma^*_{\min}} \leq C\frac{\|\boldsymbol{X}^\star_\Omega - p\boldsymbol{X}^\star\|_F + \|\boldsymbol{W}_\Omega\|_F}{p\sigma^*_{\min}} \leq C\sqrt{\frac{\kappa^2\mu r^2\log n}{p}} + \frac{\|\boldsymbol{W}_\Omega\|_F}{p\sigma^*_{\min}} \qquad (27)$$

where we have used Lemma 3.21 of [Chen et al., 2020], with $p \geq C\kappa^6 r^2\mu\log q/n$, to bound $(\|\boldsymbol{X}^*_\Omega - p\boldsymbol{X}^*\|_F/p\sigma^*_{\min}) \leq c/\kappa^2$. By using matrix-Bernstein inequality, we can show that $\|\boldsymbol{W}_\Omega\|_F \leq 2\epsilon_{noise}\sqrt{r}p\sigma^*_{\max}$ whp. By the assumed upper bound on ϵ_{noise}, we can then argue that $\mathrm{SD}_F(\boldsymbol{U}^{(00)}, \boldsymbol{U}^*) \leq c/\kappa^2$. The rest of the proof is exactly the same as in the noiseless case (Lemma 1). $\square$

Lemma 8. *Assume that the observed matrix $\boldsymbol{Y} = \boldsymbol{X}^*_\Omega + \boldsymbol{W}_\Omega$ is corrupted by additive noise $\boldsymbol{W}$ such that $|\boldsymbol{W}_{jk}| \leq \epsilon_{noise}|\boldsymbol{X}^*_{jk}|$. Assuming also $\|\boldsymbol{u}^j\| \leq \mu_{\boldsymbol{u}}\sqrt{r/n}$, where $\mu_u = \kappa^2\mu$, $\|\boldsymbol{b}_k\| \leq \sigma^*_{\max}\mu\sqrt{r/q}$ and $\sigma_{\max}(\boldsymbol{B}) \leq C\sigma^*_{\max}$. For $\epsilon_{noise} \leq 1/(\sqrt{r}\kappa^3)$,*

(i) $\|\boldsymbol{W}_B - \mathbb{E}[\boldsymbol{W}_B]\|_F \leq \epsilon_{noise}\sqrt{r}p\sigma^*_{\max}$, *w.p. greater than* $1 - \exp(\log 2q - np/(\mu_{\boldsymbol{u}}^2 r^2))$

(ii) $\|\boldsymbol{W}_B\|_F \leq 2\sqrt{r}\epsilon_{noise}p\sigma^*_{\max}$, *w.p. same as 1).*

(iii) $\|\boldsymbol{U}^\mathsf{T}_k\boldsymbol{w}_k\|_2 \leq 2\epsilon_{noise}p\|\boldsymbol{b}^*_k\|_2$, *w.p. same as 1)*

(iv) $\|\boldsymbol{W}_U - \mathbb{E}[\boldsymbol{W}_U]\|_F \leq \epsilon_{noise}\sqrt{r}p\sigma^{*2}_{\max}$, *w.p. greater than* $1 - \exp(\log 2q - np/(\mu^2 r^2))$

(v) $\|\boldsymbol{W}_U\|_F \leq 2\epsilon_{noise}\sqrt{r}p\sigma^{*2}_{\max}$, *w.p. same as 4).*

Proof. All terms are bounded using (26) and the matrix Bernstein inequality. $\square$

C Proof of Corollary 2 (Improved guarantee for AltMin for LRMC)

In section 4.2 of [Netrapalli et al., 2013], it is shown that the least-squares updates for $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r\times q}$ are equivalent to the following QR-based updates:

$$\boldsymbol{B}^{(t)} = \arg\min_{\boldsymbol{B}}\|\boldsymbol{Y} - \boldsymbol{U}^{(t)}\boldsymbol{B}\|^2_F, \quad \boldsymbol{B}^{(t)\mathsf{T}} \overset{\mathrm{QR}}{=} \boldsymbol{V}^{(t)}\boldsymbol{R_B} \qquad (28)$$

$$\widetilde{\boldsymbol{U}}^{(t+1)} = \arg\min_{\widetilde{\boldsymbol{U}}}\|\boldsymbol{Y}^\mathsf{T} - \boldsymbol{B}^{(t)}\widetilde{\boldsymbol{U}}\|^2_F, \quad \widetilde{\boldsymbol{U}}^{(t+1)\mathsf{T}} \overset{\mathrm{QR}}{=} \boldsymbol{U}^{(t+1)}\boldsymbol{R_U} \qquad (29)$$

B-update (28) / $\mathrm{SD}_F(\mathbf{V}^{(t)}, \mathbf{V}^*)$ **Bound**   Note that the $\boldsymbol{B}$-update in (28) is exactly the same as the $\boldsymbol{B}$-update for AltGDMin. Therefore, we use Lemma 2 to bound $\mathrm{SD}_F(\boldsymbol{V}^{(t)}, \boldsymbol{V}^*)$. Let $\boldsymbol{P}_{*,\perp} :=$ $\boldsymbol{I} - \boldsymbol{V}^*\boldsymbol{V}^{*\mathsf{T}}$. Then,

$$\mathrm{SD}_F(\boldsymbol{V}^{(t)}, \boldsymbol{V}^*) = \|\boldsymbol{P}_{*,\perp}\boldsymbol{V}^{(t)}\|_F = \|\boldsymbol{P}_{*,\perp}\boldsymbol{B}^\mathsf{T}\boldsymbol{R}_B^{-1}\|_F \le \|\boldsymbol{P}_{*,\perp}\boldsymbol{B}^\mathsf{T}\|_F\|\boldsymbol{R}_B^{-1}\| \le \|(\boldsymbol{B}-\boldsymbol{G})^\mathsf{T}\|_F/\sigma_{\min}(\boldsymbol{B}) \tag{30}$$

where $\boldsymbol{P}_{*,\perp}\boldsymbol{B}^\mathsf{T} = \boldsymbol{P}_{*,\perp}(\boldsymbol{G}^\mathsf{T}+(\boldsymbol{B}-\boldsymbol{G})^\mathsf{T}) = \boldsymbol{P}_{*,\perp}(\boldsymbol{B}-\boldsymbol{G})^\mathsf{T}$, because $\boldsymbol{G}^\mathsf{T} = \boldsymbol{X}^{*\mathsf{T}}\boldsymbol{U}^\mathsf{T}$ and the column-span of $\boldsymbol{X}^{*\mathsf{T}}$ is orthogonal to the span of $\boldsymbol{P}_{*,\perp}$. By Lemma 2, $\|\boldsymbol{B}-\boldsymbol{G}\|_F \le \sigma_{\min}^*\delta^{(t-1)}/4$. By Lemma 3, assuming $\delta^{(t-1)} \le 1/(4\kappa)$, $\sigma_{\min}(\boldsymbol{B}) \ge \sigma_{\min}^*/2$. Substituting these bounds in (30) and recalling that $\delta^{(t-1)} := \mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*)$, w.p. greater than $1 - \exp(\log q - pn/(r^2\mu^2\kappa^4))$,

$$\mathrm{SD}_F(\boldsymbol{V}^{(t)}, \boldsymbol{V}^*) \le \frac{1}{2}\mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*) \tag{31}$$

Incoherence of $\mathbf{V}^{(t)}$ Recall that $\widetilde{\boldsymbol{B}}^\mathsf{T} \overset{\mathrm{QR}}{=} \boldsymbol{V}\boldsymbol{R}$. Then, the k-th row, $\|\boldsymbol{v}^k\|_2 \le \|\boldsymbol{b}_k\|_2/\sigma_{\min}^*(\boldsymbol{B}) \le C\kappa\mu\sqrt{r/q}$, where the last inequality follows from Lemmas 3 and 4. Thus, $\boldsymbol{V}$ is $\kappa\mu$-incoherent.

U-Update (29) / $\mathrm{SD}_F(\mathbf{U}^{(t+1)}, \mathbf{U}^*)$ **Bound and Incoherence of** $\mathbf{U}^{(t+1)}$ The two AltMin steps are symmetric, so arguments analogous to the above help show that

$$\mathrm{SD}_F(\boldsymbol{U}^{(t+1)}, \boldsymbol{U}^*) \le \frac{1}{2}\mathrm{SD}_F(\boldsymbol{V}^{(t)}, \boldsymbol{U}^*) \tag{32}$$

Combining (31) and (32), we have $\mathrm{SD}_F(\boldsymbol{U}^{(t+1)}, \boldsymbol{U}^*) \le \mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*)/4$. For initialization we use the few lines of our Algorithm 1 and Lemma 1. Consequently, AltMinComplete [Netrapalli et al., 2013] initialized as given in the first few lines of our Algorithm 1 needs $T = \log(1/\epsilon)$ iterations for $\mathrm{SD}_F(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*) \le \epsilon$.

| Algorithm	Computation Comp.	Communic. Comp.	Sample Comp.
AltGDMin (Priv) (Proposed)	$\kappa^2 \frac{\lvert\Omega\rvert}{\gamma} r^2 \log(\frac{1}{\epsilon})$	$\kappa^2 nr \log(\frac{1}{\epsilon})$	$\kappa^6 \mu^2 nr^2 \log n \log(\frac{1}{\epsilon})$
FactGD (Priv) [Yi et al., 2016, Zheng & Lafferty, 2016]	$\kappa\mu \frac{\lvert\Omega\rvert}{\gamma} r^2 \log(\frac{1}{\epsilon})$	$\kappa\mu nr^2 \log(\frac{1}{\epsilon})$	$\kappa^4 \mu^2 nr^2 \log n$
AltMin (Priv) [Netrapalli et al., 2013]	$\frac{\lvert\Omega\rvert}{\gamma} r \log^2(\frac{1}{\epsilon})$	$nr \log^2(\frac{1}{\epsilon})$	$\kappa^4 \mu^2 nr^{4.5} \log n \log(\frac{1}{\epsilon})$
AltMin (Not-Priv) [Netrapalli et al., 2013]	$\frac{\lvert\Omega\rvert}{\gamma} r^2 \log(\frac{1}{\epsilon})$	$\frac{\lvert\Omega\rvert}{\gamma} \log(\frac{1}{\epsilon})$	$\kappa^4 \mu^2 nr^{4.5} \log n \log(\frac{1}{\epsilon})$
Smooth-AltMin (Not-Priv) [Hardt, 2014]	$\frac{\lvert\Omega\rvert}{\gamma} r^2 \log n \log(\frac{1}{\epsilon})$	$\frac{\lvert\Omega\rvert}{\gamma} \log n \log(\frac{1}{\epsilon})$	$\kappa^2 \mu^2 nr^3 \log n \log(\frac{1}{\epsilon})$
ProjGD (Not-Priv) [Jain & Netrapalli, 2015, Cherapanamjeri et al., 2016]	$\mu^4 \frac{\lvert\Omega\rvert}{\gamma} r \log^2(\frac{1}{\epsilon})$	$\mu^4 \frac{\lvert\Omega\rvert}{\gamma} \log^2(\frac{n}{\epsilon})$	$\mu^4 nr^2 \log^2 n \log^2(\frac{1}{\epsilon})$

Table 1. The table assumes $n \approx q$, γ is a numerical constant, $\max(\log(1/\epsilon), r) = \log(1/\epsilon)$, and $\lvert\Omega\rvert \geq nr$ (necessary). Here Communic Comp $= T \cdot \max(\text{Communic.(node)}, \text{Communic.(center)})$. Similarly for the computation cost.
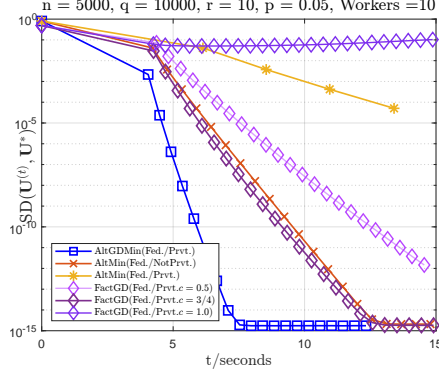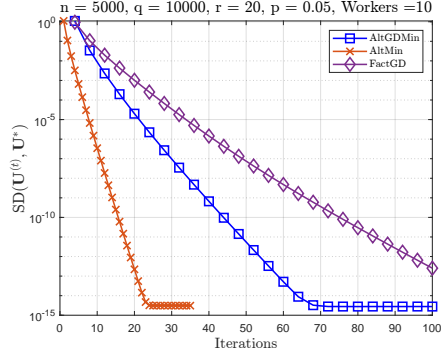
(a) Federated, $r = 10$.
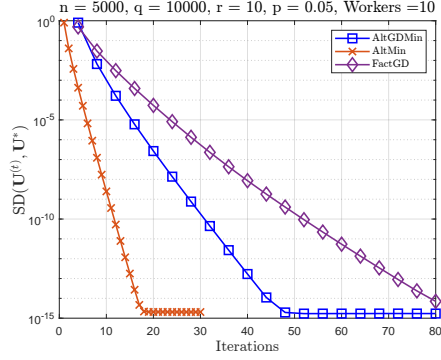
(b) Error against iterations, $r = 10$.
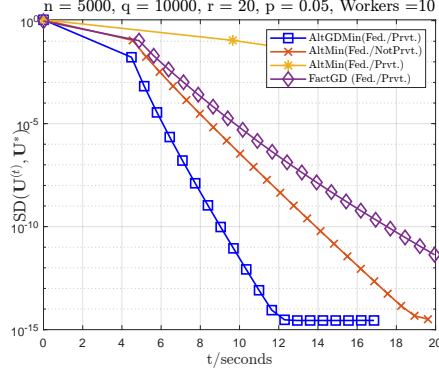
(c) Federated, $r = 20$.

(d) Error against iterations, $r = 20$.

Figure 1. Figures (a),(c) compare federated implementations of AltGDmin (proposed), FactGD and AltMin. The results match what our theory (sufficient conditions) predicts. AltGDmin is the fastest due to its lowest communication-efficiency and due to all three having comparable computation cost. Figures (b), (d) compare iteration complexity numerically. The results match theory once again. Two versions of AltMin are compared: AltMin (Fed/Not-Prvt) uses exact LS solution for updating both $\boldsymbol{B}$ and $\boldsymbol{U}$. AltMin (Fed/Prvt) uses multiple gradient descent iterations to solve the LS problem for updating $\boldsymbol{U}$. In (a), we also compare FactGD with three choices of step size. See experiments' description for details.