PROVABLY COMMUNICATION-EFFICIENT FEDERATED ROBUST MATRIX COMPLETION AND PCA

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust low rank matrix completion (LRMC) and PCA find important applications in computer vision, recommender system design, and, most recently, in new approaches to parameter-efficient fine-tuning of large language models (LLMs). In many of these applications, the data is either available in a federated fashion, or it is available centrally, but we need to distribute it in order to speed up computing. In both settings, there is a need for communication-efficient solutions. In this work, we introduce a novel solution approach and prove that it is r-times more communication efficient than, and order-wise as fast as, the fastest existing robust LRMC solution, while tolerating the same number of outliers and having comparable sample complexity (order-wise). Here r denotes the rank of the unknown low rank matrix to be recovered. Our proposed solution relies on a multi-block generalization of the recently introduced Alternating Gradient Descent and Minimization (AltGDmin) framework, combined with a new application of a mergeable quantile sketching algorithm, KLL, for federated computation of the global row thresholds needed for outlier support estimation. We prove that the resulting federated algorithm converges under the same assumptions, and with the same outlier tolerance and sample complexity, as its centralized counterpart. Experiments on simulated data and on real videos corroborate our claims.

1 INTRODUCTION

Robust low rank matrix completion (LRMC) and PCA find important applications in computer vision Pan et al. (2024); Narayanamurthy & Vaswani (2018), recommender system design Koren et al. (2009); Zhao et al. (2024), and, most recently, in new approaches to parameter-efficient fine-tuning of large language models (LLMs) Nikdan et al. (2024); Makni et al. (2025). In many of these applications, the data is either available in a federated fashion, or it is available centrally, but we need to distribute it in order to speed up computing. In both settings, there is a need for communication-efficient solutions. In this work, we develop and study a novel solution approach for efficiently solving these problems in a distributed setting. Our approach relies on a generalization of the basic two-block Alternating Gradient Descent and Minimization (AltGDmin) algorithm Nayer & Vaswani (2023); Collins et al. (2021); Vaswani (2024) and a novel application of a well-known quantile sketching algorithm.

1.1 PROBLEM SETTING AND ASSUMPTIONS

Robust PCA (RPCA) involves recovering a low rank (LR) matrix $\boldsymbol{\Theta}^* \in \Re^{n\times d}$ from

$$\boldsymbol{Z} = \boldsymbol{\Theta}^* + \boldsymbol{S}^* \tag{1}$$

Here $\boldsymbol{S}^*$ is a sparse matrix with nonzero entries denoting the outliers; we use $\boldsymbol{\Omega}^*$ to denote the support of $\boldsymbol{S}^*$. $\boldsymbol{\Theta}^*$ is a rank r matrix with $r \ll \min(n,d)$. Thus it can be factored as $\boldsymbol{\Theta}^* = \boldsymbol{U}^*\boldsymbol{B}^*$ where $\boldsymbol{U}^*$ is the $n \times r$ matrix of its top r left singular vectors. We use κ to denote the condition number of $\boldsymbol{B}^{*\top}$. Robust LR matrix completion is a generalization of robust PCA and matrix completion. It involves recovering $\boldsymbol{\Theta}^*$ from a subset of observed entries of $\boldsymbol{Z}$, i.e. from

$$\boldsymbol{Y} := \Pi_{\boldsymbol{\Phi}}(\boldsymbol{Z}) = \Pi_{\boldsymbol{\Phi}}(\boldsymbol{\Theta}^*) + \Pi_{\boldsymbol{\Phi}}(\boldsymbol{S}^*) \tag{2}$$

where $\boldsymbol{\Phi}$ denotes the set of indices of the observed entries, and $\Pi_{\boldsymbol{\Phi}}$ zeros out the unobserved ones. In this case, there is no way to estimate the unobserved sparse entries and thus, to keep notation simple, we assume w.l.o.g. that $\Pi_{\boldsymbol{\Phi}^c}(\boldsymbol{S}^*) = \boldsymbol{0}$.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: *Robust LRMC and PCA comparisons. Table assumes μ and κ are numerical constants. $n_{mx} = \max(n, d)$. For Robust PCA ($p = 1$) Sample Cost is not applicable.*

Robust LRMC				
Algorithm	**Computation Cost**	**Communication Cost**	**Sample Cost**	**Outlier Tolerance**
PG-RMC Cherapanamjeri et al. (2017)	$\frac{\lvert\mathbf{\Phi}\rvert}{L} n \log^2(\frac{1}{\epsilon})$	$\frac{\lvert\mathbf{\Phi}\rvert}{L} \log^2(\frac{n}{\epsilon})$	$nr^2 \log^2 n \log^2(\frac{1}{\epsilon})$	$\frac{1}{r}$
FactGD Yi et al. (2016)	$\frac{\lvert\mathbf{\Phi}\rvert}{L} r^2 \log(\frac{1}{\epsilon})$	$nr^2 \log(\frac{1}{\epsilon})$	$nr^2 \log n$	$\frac{1}{Lr^{3/2}}$ & $\lvert\mathcal{S}_\ell\rvert = d/L$
Gen-AltGDmin (local threshold)	$\frac{\lvert\mathbf{\Phi}\rvert}{L} r^2 \log(\frac{1}{\epsilon})$	$nr \log(\frac{1}{\epsilon})$	$nr^2 \log n \log(\frac{1}{\epsilon})$	$\frac{1}{Lr^{3/2}}$ & $\lvert\mathcal{S}_\ell\rvert = d/L$
Gen-AltGDmin (global exact thresh)	$\frac{\lvert\mathbf{\Phi}\rvert}{L} \max(r^2, L) \log(\frac{1}{\epsilon})$	$\max(\frac{\lvert\mathbf{\Phi}\rvert}{r^{3/2}}, nr) \log(\frac{1}{\epsilon})$	$nr^2 \log n \log(\frac{1}{\epsilon})$	$\frac{1}{r^{3/2}}$
Gen-AltGDmin-KLL (proposed)	$\frac{\lvert\mathbf{\Phi}\rvert}{L} \max(r^2, L^2) \log(\frac{1}{\epsilon})$	$\max(nr, nL) \log(\frac{1}{\epsilon})$	$nr^2 \log n \log(\frac{1}{\epsilon})$	$\frac{1}{r^{3/2}}$
Robust PCA ($p = 1$)				
Algorithm	**Computation Cost**	**Communication Cost**		**Outlier Tolerance**
AltProj Netrapalli et al. (2014)	$n\frac{n_{mx}^2}{L} \log(\frac{1}{\epsilon})$	$\frac{nd}{L} \log^2(\frac{n}{\epsilon})$		$\frac{1}{r}$
Gen-AltGDmin (global exact threshold)	$\frac{n_{mx}^2}{L} \max(r^2, L) \log(\frac{1}{\epsilon})$	$\max(\frac{n_{mx}^2}{r^{3/2}}, nr) \log(\frac{1}{\epsilon})$		$\frac{1}{r^{3/2}}$
Gen-AltGDmin-KLL (proposed)	$\frac{n_{mx}^2}{L} \max(r^2, L^2) \log(\frac{1}{\epsilon})$	$\max(nr, nL) \log(\frac{1}{\epsilon})$		$\frac{1}{r^{3/2}}$

Notation. $^\top$ denotes transpose; $\|.\|_F$ the Frobenius norm and $\|.\|$ the induced ℓ_2 norm. For $\boldsymbol{M} \in \Re^{n\times d}$, $\boldsymbol{m}_{jk}$, $\boldsymbol{m}^j$, $\boldsymbol{m}_k$ denote its (j,k)-th entry, j-th row and k-th column; similarly for an index set $\mathbf{\Phi} \subseteq [n] \times [d]$. We denote the γ-th largest magnitude element in any vector $\boldsymbol{y}$ by $\boldsymbol{y}^{(\gamma)}$. We let $n_{mx} = \max(n, d)$.

Assumptions. We make the following three assumptions; all standard Yi et al. (2016); Candès et al. (2011).

Assumption 1 (μ-Incoherence). Assume row incoherence of $\boldsymbol{U}^*$: $\max_{j\in[n]} \|\boldsymbol{u}^{*j}\| \le \mu\sqrt{r/n}$, and column incoherence of $\boldsymbol{B}^*$, i.e., $\max_{k\in[d]} \|\boldsymbol{b}_k^*\| \le \mu\sqrt{r/d}\sigma_{\max}(\mathbf{\Theta}^*)$ for a constant $\mu \ge 1$.

Assumption 2 (Bernoulli(p) model on observed entries). Each entry of $\mathbf{\Theta}^*$ is observed, independently of all other observations, with probability p. In other words, index $(j,k) \in \mathbf{\Phi}$ with probability p independent of all others. Full or complete robust PCA is this setting with $p = 1$.

Assumption 3 (Fraction of sparse corruptions in observed entries of each row or column at most α). Robust PCA: The fraction of outliers (nonzero entries of $\boldsymbol{S}^*$) in any row or any column is at most α. Robust LRMC: this fraction is now the fraction of the observed entries in each row or column. To be precise $|\mathbf{\Omega}^{*j}| \le \alpha|\mathbf{\Phi}^j|$ for all $j \in [n]$ and $|\mathbf{\Omega}_k^*| \le \alpha|\mathbf{\Phi}_k|$ for all $k \in [d]$. Thus the fraction of nonzeros in each row or column of $\boldsymbol{S}^*$ is approximately αp with high probability (w.h.p.).

Federation. We use L to denote the number of distributed nodes. Let $\mathcal{S}_\ell, \ell \in [L]$ be a partition of $[d] := \{1, 2, \dots, d\}$. Node ℓ observes a column sub-matrix of $\boldsymbol{Y}$ with columns in the index set $\mathcal{S}_\ell$. We denote this by $\boldsymbol{Y}_{(\ell)} := [\boldsymbol{y}_k, k \in \mathcal{S}_\ell]$. Here $\boldsymbol{y}_k := \Pi_{\mathbf{\Phi}_k}(\boldsymbol{\theta}_k^* + \boldsymbol{s}_k^*)$ with $\mathbf{\Phi}_k := \{j : (j,k) \in \mathbf{\Phi}\}$ is the set of indices of the observed entries of column k. Similarly, we can define $\mathbf{\Theta}_{(\ell)}^*$ and $\boldsymbol{S}_{(\ell)}^*$. The goal is to recover $\boldsymbol{U}^*$ globally, and to recover $\boldsymbol{B}_{(\ell)}^*$ at node ℓ for each $\ell \in [L]$.

While we assume vertical federation here, since the problem is row/column symmetric, everything we prove also applies for the horizontal setting.

Remark (unbalanced column partition). We place no lower bound on $|\mathcal{S}_\ell|$: the column partition may be arbitrarily unbalanced (heterogeneous) across nodes, and a node may hold even a single column of $\boldsymbol{Y}$, since row thresholds are computed globally from merged sketches (Sec. 2.3); local-thresholding extensions need every $|\mathcal{S}_\ell|$ large enough. See Fig. 1b.

1.2 Related Work

In the centralized setting, both robust PCA and robust matrix completion have been extensively studied. Well known solutions with provable guarantees include convex relaxation (nuclear norm minimization) Candès et al. (2011); Wright et al. (2009), alternating minimization over $\mathbf{\Theta}$ and $\boldsymbol{S}$ (AltProj) Netrapalli et al. (2014), AltMin over $\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}$ with $\mathbf{\Theta}$ factored as $\mathbf{\Theta} = \boldsymbol{U}\boldsymbol{B}$ Gu et al. (2016), projected GD (PG-RMC) Cherapanamjeri et al. (2017) and GD over $\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}$ with a norm balancing term added to account for the fact that the factorization $\mathbf{\Theta} = \boldsymbol{U}\boldsymbol{B}$ is not unique Yi et al. (2016); we refer to this last method as factorized GD (factGD). Solving a convex relaxation is very slow. AltProj is faster, but is still quite slow because it updates $\mathbf{\Theta}$ using an SVD step. The same is true for projected-GD. AltMin-$\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}$ is very fast but the guarantees provided in Gu et al. (2016) for it require an impractically small bound on allowed fraction of outliers per row or column of

$\mathcal{O}(1/dr)$. This means, number of outliers per row needs to be $O(1/r)$, which means essentially zero outliers are allowed. Factorized GD (FactGD) is as fast as AltMin-$\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}$ and has a reasonable outlier tolerance. Its outlier tolerance is worse than that of AltProj by a factor of $\sqrt{r}$. AltProj is studied only for robust PCA though. FactGD has a provably correct extension for robust LRMC also.

AltGDmin was introduced as a fast solution for the LR column-wise sensing (LRCS) problem. LRCS involves learning an $n \times d$ rank-r matrix $\boldsymbol{\Theta}^* := [\boldsymbol{\Theta}_1^*, \boldsymbol{\Theta}_2^*, ..., \boldsymbol{\Theta}_d^*]$ from column-wise undersampled measurements, $\boldsymbol{y}_k := \boldsymbol{X}_k \boldsymbol{\theta}_k^*, k \in [d]$ when $\boldsymbol{X}_k \in \Re^{m \times n}$ with $m < n$. Factoring the unknown LR matrix $\boldsymbol{\Theta}$ as $\boldsymbol{\Theta} = \boldsymbol{U}\boldsymbol{B}$, the loss function to be minimized becomes $f(\boldsymbol{U}, \boldsymbol{B}) := \sum_{k=1}^{d} \|\boldsymbol{y}_k - \boldsymbol{X}_k \boldsymbol{U} \boldsymbol{b}_k\|^2$. Clearly this problem is column-wise decoupled for $\boldsymbol{B}$, but not for $\boldsymbol{U}$. Minimization over $\boldsymbol{B}$ is thus fast. Also, it can be done locally at the node at which $\boldsymbol{y}_k$ is available. AltGDmin is an efficient solution for such partly decoupled optimization problems. It alternates between using GD for the coupled block and minimization for the decoupled one. It was later also developed for efficiently federating matrix completion Abbasi & Vaswani (2025).

There has been some work on federated PCA Fan et al. (2019); Chen et al. (2020); Huang & Pan (2020); Shi & Al Kontar (2024); Singh & Vaswani (2024b); Singh et al. (2025); Singh & Vaswani (2024a), but these methods cannot handle arbitrary sparse outliers and missing entries in all columns of the matrix.

1.3 Contributions

This work introduces a novel solution approach that significantly generalizes the two-block AltGDmin idea and its analysis. The resulting generalized AltGDmin algorithm needs r-times fewer iterations than the fastest existing solution for robust LRMC, factorized GD (factGD) Yi et al. (2016), is order-wise as fast as it, and can be federated efficiently. The two most direct approaches to federate it either require extra assumptions (AltGDmin-local-thresholding) or too much communication (AltGDmin-global-exact-thresholding). We show how to carefully adapt a quantile sketching algorithm called KLL Karnin et al. (2016) for addressing both these issues. Our resulting algorithm, generalized-AltGDmin-KLL, is communication-efficient and converges under the same assumptions as those needed by gen-AltGDmin for the centralized setting. It is thus r-times more communication-efficient than, and order-wise as fast as, a naive federated extension of factGD. See Table 1.

Novelty. (1) All past works have studied the simple two-block AltGDmin algorithm, which suffices for two-block problems such as LRCS or LRMC. The natural split of unknowns in our problems involves three blocks, which introduces new challenges. Consider full robust PCA: minimize $f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}) = \sum_{k=1}^{d} \|\boldsymbol{y}_k - \boldsymbol{U}\boldsymbol{b}_k - \boldsymbol{s}_k\|^2$ with $\boldsymbol{U}$ orthonormal and $\boldsymbol{S}$ sparse. Simple AltGDmin would treat $\boldsymbol{U}$ as the coupled block and $\{\boldsymbol{B}, \boldsymbol{S}\}$ as the column-wise decoupled one, initialize $\boldsymbol{U}$, and alternate between minimization over $\{\boldsymbol{B}, \boldsymbol{S}\}$ and GD for $\boldsymbol{U}$. But the sparse component has no bound on its entries' magnitudes, while those of $\boldsymbol{U}, \boldsymbol{B}$ are bounded by incoherence; so $\boldsymbol{S}$ must be initialized and updated first, else the large sparse corruptions in $\boldsymbol{Y}$ make the $\boldsymbol{U}$ error large (and similarly in the iterations). Our proposed solution is a careful multi-block extension of simple AltGDmin that updates $\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{B}$ in this order, one at a time; $\boldsymbol{U}$ by GD and $\boldsymbol{S}, \boldsymbol{B}$ by minimization. A second advantage is that closed-form, efficient minimizers exist for $\boldsymbol{S}$ and for $\boldsymbol{B}$ separately but not jointly.

(2) For LRMC and LRCS, the federated solution was a straightforward modification of the centralized setting (just update columns of $\boldsymbol{B}$ locally while updating $\boldsymbol{U}$ using federated GD) with no changes needed to its analysis. If we extend this approach to our problems, we would update columns of $\boldsymbol{B}$ and of $\boldsymbol{S}$ locally at the nodes at which the corresponding $\boldsymbol{y}_k$ is observed, while using federated GD to update $\boldsymbol{U}$. There is one major difference here though. To update $\boldsymbol{S}$, we use the row and column thresholding approach from Yi et al. (2016); see (3). Use of this approach is needed to allow us to obtain the desired outlier tolerance guarantees. However, this requires computing a row threshold and a column threshold for each row and column at each iteration. One approach to obtain the row thresholds is to compute them locally for the matrix subset at each node. This approach, which we refer to as AltGDmin (local thresholding) in Table 1, requires either assuming that the number of outliers per row is roughly equal at each node, or tolerates L-times fewer outliers. It also needs $|\mathcal{S}_\ell|$ to be large enough for each node ℓ for local thresholding to work accurately. One obvious way to remove the above limitation is to have nodes share a certain number of largest magnitude

entries of each row of $(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B})$ with the center which computes the (global) row thresholds. These are broadcast back to the nodes. This approach, which we refer to as AltGDmin (global exact thresholding) would not need extra assumptions, but it would need too much communication, especially for the full robust PCA setting. See Table 1.
To obtain a solution that is communication-efficient and does not need extra assumptions, we adapt a quantile sketching algorithm, called KLL Karnin et al. (2016), introduced for efficiently compressing streaming data while preserving enough information about it so that quantile queries can be answered approximately correctly. This algorithm can be adapted because it is "mergeable", which means that disjoint subsets of the data can be sketched independently (in our case at the different nodes) and the combined sketch is provably as accurate as a single sketch of the entire data.

(3) *Analysis.* Since this is the first work to study gen-AltGDmin, and one in which one of the three blocks is a sparse matrix, there are several new challenges in analyzing it. (i) We need to bound the error in updating the sparse matrix $\boldsymbol{S}$ by a quantity that is proportional to the subspace recovery error in updating $\boldsymbol{U}$ at the previous iteration, so that exponential error decay can be proved for both. (ii) We need to prove incoherence of the updated $\boldsymbol{U}$ at each iteration, instead of enforcing it by construction as done for FactGD. This step requires a tight bound on the maximum row norm of $(\boldsymbol{S}^{+} - \boldsymbol{S}^{*})\boldsymbol{B}^{\top}$ that is of order $\sqrt{r/n}$. See Sec. 3.
Consider gen-AltGDmin-KLL. To show that this is communication-efficient and still provably correct, we need careful analysis to combine guarantees for the quantile sketching algorithm with our centralized case proofs in a way that the row threshold approximation error due to sketching is small enough and yet the communication cost remains small. By doing this we are able to show that the resulting algorithm is as communication efficient as the local-thresholding solution while being as outlier tolerant as the global-exact-thresholding solution. Our proof requires three new bounds: a sparse estimation error bound for the thresholding-based $\boldsymbol{S}$ update that holds with a constant γ (Lemma A.3); a cross-term bound on $\|\boldsymbol{U}^{\top}\Pi_{\boldsymbol{\Phi}}(\boldsymbol{S} - \boldsymbol{S}^{*})\|_F$ re-derived on the fresh sample subset in every iteration (Lemma A.7); and an initialization bound (Lemma A.1).

2 Federated Robust LRMC and PCA

We give more notation and definitions next, then develop the algorithm and state its guarantees.

2.1 More Notation and Definitions

For tall matrices with orthonormal columns $\boldsymbol{U}_1, \boldsymbol{U}_2$, we use $\boldsymbol{SD}(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1\boldsymbol{U}_1{}^{\top})\boldsymbol{U}_2\|_F$ is used as the Subspace Distance (SD) measure between the column spans of the two matrices. For a matrix $\boldsymbol{M} = [\boldsymbol{m}_1, ..., \boldsymbol{m}_d]$, we denote its column sub-matrix with columns corresponding to indices in $\mathcal{S}_\ell$ by $\boldsymbol{M}_{(\ell)}$. Thus, $\boldsymbol{M} = [\boldsymbol{M}_{(1)}, \ldots, \boldsymbol{M}_{(\ell)}, \ldots, \boldsymbol{M}_{(L)}]$. *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 $\boldsymbol{\Theta}^* \overset{\text{SVD}}{=} \boldsymbol{U}^*(\boldsymbol{\Sigma}^*)\boldsymbol{V}^* := \boldsymbol{U}^*\boldsymbol{B}^*$, where $\boldsymbol{U}^* \in \Re^{n\times r}$ and has orthonormal columns and $\boldsymbol{V}^* \in \Re^{r\times d}$ with orthonormal rows. Also, we let $\boldsymbol{B}^* := \boldsymbol{\Sigma}^*\boldsymbol{V}^*$ so that $\boldsymbol{\Theta}^* = \boldsymbol{U}^*\boldsymbol{B}^*$. We state guarantees in terms of $\kappa = \frac{\sigma_1^*}{\sigma_r^*}$ where $\sigma_1^* = \sigma_{\max}(\boldsymbol{\Theta}^*)$, and $\sigma_r^* = \sigma_r(\boldsymbol{\Theta}^*) = \sigma_{\min}(\boldsymbol{B}^{*\top})$.

For any index set $\boldsymbol{\Phi} \subseteq [n] \times [d]$,

$$\Pi_{\boldsymbol{\Phi}}(\boldsymbol{M}) = \begin{cases} \boldsymbol{m}_{jk}, \text{ if } (j,k) \in \boldsymbol{\Phi} \\ 0, \text{ otherwise} \end{cases} .$$

We also use $\Pi_{\mu,row}(\boldsymbol{M})$, and $\Pi_{\mu,col}(\boldsymbol{M})$ for projecting $\boldsymbol{M}$ onto space of μ-row, and μ-column incoherent matrices respectively; these are minor modifications of similar operators used in Yi et al. (2016). It is easy to see that this projection can be obtained in closed form as follows:

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

$$[\Pi_{\mu,col}(\boldsymbol{M})]_k = \boldsymbol{m}_k \cdot \min\left(1, \frac{\mu\sqrt{r/d}\sigma_1^*}{\|\boldsymbol{m}_k\|}\right), \text{ for all } k \in [d].$$

Algorithm 1 Centralized setting: Generalized AltGDmin

1: **Input:** Matrix $\boldsymbol{Y}$ with support $\Pi_{\boldsymbol{\Phi}}$, thresholding parameter γ, and corruption fraction α.
2: **Initialization**
3: $\boldsymbol{S} \leftarrow \mathcal{T}_{2\alpha p}(\Pi_{\boldsymbol{\Phi}}(\boldsymbol{Y}))$
4: $\boldsymbol{U} \leftarrow QR(\Pi_{\mu,row}(\text{SVD}_r(p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0))))$
5: **GD step**
6: **for** $t = 1$ to T **do**
7: $\quad \tilde{\boldsymbol{b}}_k \leftarrow (\boldsymbol{U}^\top \Pi_{\boldsymbol{\Phi}_k} \boldsymbol{U})^{-1} \boldsymbol{U}^\top \Pi_{\boldsymbol{\Phi}_k}(\boldsymbol{y}_k - \boldsymbol{s}_k)$ for $k \in [d]$ and $\tilde{\boldsymbol{B}} \leftarrow [\tilde{\boldsymbol{b}}_1, \tilde{\boldsymbol{b}}_2, ..., \tilde{\boldsymbol{b}}_d]$
8: $\quad \boldsymbol{B} \leftarrow \Pi_{\mu,col}(\tilde{\boldsymbol{B}})$
9: $\quad \boldsymbol{S}^+ \leftarrow \mathcal{T}_{\gamma\alpha p}(\Pi_{\boldsymbol{\Phi}}(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B}))$
10: $\quad \boldsymbol{U}^+ \leftarrow QR(\boldsymbol{U} - \eta \nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+))$ where $\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+) \leftarrow (\Pi_{\boldsymbol{\Phi}}(\boldsymbol{U}\boldsymbol{B} + \boldsymbol{S}^+ - \boldsymbol{Y}))\boldsymbol{B}^\top$
11: $\quad$ Set $\boldsymbol{U} \leftarrow \boldsymbol{U}^+$, $\boldsymbol{S} \leftarrow \boldsymbol{S}^+$
12: **end for**

For updating the sparse component, we will use the following sparsification operator Yi et al. (2016):

$$\mathcal{T}_\beta(\boldsymbol{M}) = \begin{cases} \boldsymbol{m}_{jk}, & \text{if } |\boldsymbol{m}_{jk}| \geq \max(|\boldsymbol{m}^{j\,(\beta d)}|, |\boldsymbol{m}_k^{(\beta n)}|), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In words, this thresholding operator zeros out small magnitude entries of $\boldsymbol{M}$ to ensure that the fraction of nonzeros in any row or in any column is at most β: any entry whose magnitude is smaller than either the βd-th largest entry in that row or the βn-th largest entry in that column is zeroed out.

Recall that $\boldsymbol{\Phi}_k = \{j \in [n] | (j,k) \in \boldsymbol{\Phi}\}$ is the support of $\boldsymbol{y}_k$. In matrix notation, $\Pi_{\boldsymbol{\Phi}_k} = \text{diag}(\xi_{jk}, j \in [n])$ where $\xi_{jk} \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$ for $j \in [n], k \in [d]$. For fully observed case $p = 1$ therefore $\Pi_{\boldsymbol{\Phi}_k}$ becomes identity matrix $\boldsymbol{I}_n$.

2.2 Centralized setting: Generalized AltGDmin

By using squared loss to define our objective, the goal is to minimize

$$f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}) = \sum_{k=1}^{d} \|\Pi_{\boldsymbol{\Phi}_k}(\boldsymbol{y}_k - \boldsymbol{U}\boldsymbol{b}_k - \boldsymbol{s}_k)\|^2 = \|\Pi_{\boldsymbol{\Phi}}(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B} - \boldsymbol{S})\|_F^2,$$

over $\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}$ with $\boldsymbol{U} \in \Re^{n \times r}$ having orthonormal columns, i.e., $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{B} \in \Re^{r \times d}$, and $\boldsymbol{S} \in \Re^{n \times d}$ being a sparse matrix. We alternatively update one of the three variable subsets, keeping the other two fixed. For reasons explained earlier in Sec. 1.3, the initialization and update sequence needs to be $\boldsymbol{S}, \boldsymbol{U}, \boldsymbol{B}$. Because $\boldsymbol{U}$ is coupled, we update it by one step of GD for it, while updating the other two by minimization.

We present the proposed centralized algorithm in Algorithm 1. It proceeds as follows: We initialize $\boldsymbol{S}$ by thresholding $\boldsymbol{Y}$ as $\boldsymbol{S} = \mathcal{T}_{2\alpha p}(\boldsymbol{Y})$, and then initialize $\boldsymbol{U}$ by computing the top r left singular vectors of $(\boldsymbol{Y} - \boldsymbol{S})$. We update $\boldsymbol{B}$ by minimization over it, keeping $\boldsymbol{S}, \boldsymbol{U}$ fixed. This is clearly a decoupled least squares (LS) problem with a simple closed form solution; see line 7. We update $\boldsymbol{S}$ using $\boldsymbol{S}^+ = \mathcal{T}_{\gamma\alpha p}(\boldsymbol{Y} - \Pi_{\boldsymbol{\Phi}}(\boldsymbol{U}\boldsymbol{B}))$. Here γ is an algorithm parameter that we set in our main guarantee. We update $\boldsymbol{U}$ by one step of GD followed by orthonormalization, see line 10.

2.3 Federated setting: Generalized-AltGDmin-KLL

The simplest federated extension of the above algorithm proceeds as follows. We update columns of $\boldsymbol{B}$ and of $\boldsymbol{S}$ locally at the nodes at which the corresponding $\boldsymbol{y}_k$ is observed. We use federated GD to update $\boldsymbol{U}$. However, this requires computing row thresholds for updating each row of $\boldsymbol{S}$ locally at the nodes; for reasons explained earlier in Sec. 1.3, this requires extra assumptions (fewer outliers). A naive approach to compute the exact global threshold for each row is to send the top βd elements of the row at each node to the center with $\beta = \gamma\alpha p$. The center then sorts these to compute the βd-th largest element of the full row and broadcasts this back. Doing this for all the rows has very high communication cost, in the worst case for $p = 1$ (full robust PCA) it is $\frac{nd}{\mu^2 r^{3/2} \kappa^3}$.

We address this issue by sending sketches of the largest β quantiles of the rows at each node. We adapt a mergeable quantile sketching algorithm called KLL Karnin et al. (2016) which was designed

Algorithm 2 Federated setting: Generalized-AltGDmin-KLL

Parameters: ϵ, γ, η, K, H
Initialization: Obtain $\boldsymbol{S}, \boldsymbol{U}$ using Algorithm 3
for $t = 1$ to T **do**
  Nodes $\ell = 1, ..., L$
  $\tilde{\boldsymbol{b}}_k \leftarrow (\boldsymbol{U}^\top \Pi_{\boldsymbol{\Phi}_k} \boldsymbol{U})^{-1} \boldsymbol{U}^\top \Pi_{\boldsymbol{\Phi}_k}(\boldsymbol{y}_k - \boldsymbol{s}_k)$, $k \in [\mathcal{S}_\ell]$ and $\tilde{\boldsymbol{B}}_{(\ell)} \leftarrow [\tilde{\boldsymbol{b}}_1, \tilde{\boldsymbol{b}}_2, ..., \tilde{\boldsymbol{b}}_{\tilde{d}}]$
  $\boldsymbol{B}_{(\ell)} \leftarrow \Pi_{\mu,col}(\tilde{\boldsymbol{B}}_{(\ell)})$
  $\mathcal{X}_\ell^j \leftarrow$ largest $\gamma\alpha pd$ elements of $\Pi_{\boldsymbol{\Phi}}(\boldsymbol{y}_{(\ell)}^j - \boldsymbol{u}^j \boldsymbol{B}_{(\ell)})$ for $j \in [n]$
  $\mathrm{KLL}_\ell^j \leftarrow \mathrm{KLLsketch}(\mathcal{X}_\ell^j)$ for $j \in [n]$ (Algorithm 4)
  Central Server
  $\mathrm{KLL}_{glob}^j \leftarrow \mathrm{KLLmerge}\{\mathrm{KLL}_\ell^j\}_{\ell=1}^L$ for $j \in [n]$ (Algorithm 5)
  $\mathrm{Thresh}^j \leftarrow \mathrm{KLLquery}(\mathrm{KLL}_{glob}^j, \gamma\alpha pd)$, $j \in [n]$ (Algorithm 6)
  $\mathrm{Thresh} \leftarrow \{\mathrm{Thresh}^j\}_{j=1}^n$.
  Nodes $\ell = 1, ..., L$
  Set $\boldsymbol{S}_{(\ell)}^+ \longleftarrow \mathcal{T}_{\{\gamma\alpha p, \mathrm{Thresh}\}}(\Pi_{\boldsymbol{\Phi}}(\boldsymbol{Y}_{(\ell)} - \boldsymbol{U}\boldsymbol{B}_{(\ell)}))$ where $\mathcal{T}_{\{\gamma, \mathrm{Thresh}\}}(\boldsymbol{M})$ is a modified sparsification operator:

$$\mathcal{T}_{\gamma,\mathrm{Thresh}}(\boldsymbol{M}) = \begin{cases} \boldsymbol{m}_{jk}, & \text{if } |\boldsymbol{m}_{jk}| \geq \max(|\mathrm{Thresh}^j|, |\boldsymbol{m}_k^{(\gamma n)}|), \\ 0, & \text{otherwise} \end{cases}$$

  $\nabla_\ell \leftarrow (\Pi_{\boldsymbol{\Phi}}(\boldsymbol{U}\boldsymbol{B}_{(\ell)} + \boldsymbol{S}_{(\ell)}^+ - \boldsymbol{Y}_{(\ell)}))\boldsymbol{B}_{(\ell)}^\top$
  Central Server
  Compute $\boldsymbol{U}^+ \leftarrow QR(\boldsymbol{U} - \eta \sum_{\ell=1}^L \nabla_\ell)$. Set $\boldsymbol{U} \leftarrow \boldsymbol{U}^+$
end for
Output $\boldsymbol{U}_T$.

for sketching streaming data with the goal of being able to answer quantile queries approximately while using much less space; and being able to do this even when disjoint subsets of the data are sketched independently. Each node computes a KLL sketch of the $\gamma\alpha pd$-th largest elements for each of its rows (procedure KLLsketch in Appendix). The nodes send these sketches to the central server. The center merges the sketches received from the different nodes (procedure KLLmerge in Appendix) for each row. It then queries this merged sketch to approximate the βd-th largest entry of each row (procedure KLLquery in Appendix). This is then broadcast back to all nodes which then apply the sparsification operator.

Our generalized-AltGDmin-KLL algorithm can tolerate quantile approximation error of order $\epsilon = c$. Using results from Karnin et al. (2016), we can show that the communication cost per iteration for threshold computation reduces to order $nL\sqrt{\log\big(Ln_{mx}\log\big(\frac{1}{\epsilon}\big)\big)}$. We mention here that the KLL sketch can also be replaced by various other mergable sketching approaches.

We give the complete algorithm in Algorithm 2. For initialization of $\boldsymbol{S}$, we use the procedure described above with $\boldsymbol{M} = \boldsymbol{Y}$ instead of $\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B}$. For initializing $\boldsymbol{U}$, we use the federated power method to compute the singular vectors of $\boldsymbol{Y} - \boldsymbol{S}$ Hardt & Price (2014); followed by projecting the output onto the space of row incoherent matrices as explained earlier.

2.4 Theoretical Guarantees

To make our main result easy to follow, we first state its corollary that studies the simpler centralized algorithm. This shows that, if the maximum fraction of outliers per row or per column, α, is small enough, and if we observe roughly order $n_{mx} r^2 \log d \log(1/\epsilon)$ matrix entries then, w.h.p., then the subspace distance between $\boldsymbol{U}^*$ and $\boldsymbol{U}$ decays exponentially with iterations. This holds with a step size $\eta = c/p\sigma_{\max}^{*2}$ (nearly constant) and thus, we can show that our approach reaches the desired error level ϵ in roughly order $\log(1/\epsilon)$ iterations.

Corollary 2.1. *(Gen-AltGDmin - Centralized case) Consider Algorithm 1 with sample splitting. Let $T = C\kappa^2 \log(1/\epsilon)$, $\eta = 0.5/p\sigma_{\max}^{*2}$, $\gamma = C$. Assume that Assumptions 1, 2, 3 hold. If $\alpha \leq \min\Big(\frac{c}{\kappa^8\mu^2 r}, \frac{c}{\kappa^3\mu^2 r^{3/2}}\Big)$, $p \cdot nd \geq C\kappa^{10}\mu^2 n_{mx} r^2 \log n_{mx} \log(1/\epsilon)$, and $d \geq C\kappa^2 \log(1/\epsilon)$ then w.p.*

at least $1 - CT/d$,

$$\boldsymbol{SD}(\boldsymbol{U}^*, \boldsymbol{U}_T) \leq \epsilon$$

Computation cost per node is order $|\boldsymbol{\Phi}|r^2 \cdot T = \frac{1}{L}\kappa^2|\boldsymbol{\Phi}|r^2 \log(1/\epsilon)$.

Sample splitting for analysis. As in Abbasi & Vaswani (2025), iteration t uses two fresh, disjoint subsets $\boldsymbol{\Phi}_t^B, \boldsymbol{\Phi}_t^U$ of the observed entries, each with observation fraction $p/(2T)$: the LS update of $\boldsymbol{B}$ uses $\boldsymbol{\Phi}_t^B$ and the GD update of $\boldsymbol{U}$ uses $\boldsymbol{\Phi}_t^U$. On either subset, the sparse estimate used is $\boldsymbol{S} = \mathcal{T}_{\gamma\alpha p}(\Pi_{\boldsymbol{\Phi}_t^{(\cdot)}}(\boldsymbol{Y} - \boldsymbol{UB}))$ computed with the most recent $(\boldsymbol{U}, \boldsymbol{B})$, which were fixed before that subset was drawn; this makes the concentration bounds borrowed from Yi et al. (2016); Abbasi & Vaswani (2025) applicable and removes the outliers on each subset by an estimate computed on that subset. With a single sample set this returns the $\boldsymbol{S}^+$ of the previous iteration.

We now state our main result for generalized AltGDmin-KLL.

Theorem 2.2. *(Gen-AltGDmin-KLL - Federated case) Consider Algorithm 2 with sample splitting and the setting of Corollary 2.1. If we set the KLL sketching parameters* K, H *as* $K = CL\sqrt{\log\big(Ln_{mx}\log\big(\frac{1}{\epsilon}\big)\big)}$ *and* $H = 1 + \big\lceil \log\big(\frac{\alpha d}{K}\big)\big\rceil$*, the conclusion of the corollary holds. Total communication cost per node*

$$\underbrace{T \cdot nr}_{\textit{Gradient Step}} + \underbrace{(T+1) \cdot nKH}_{\textit{KLL Sketch}} = \kappa^2 \log(1/\epsilon) \cdot (nr + nL\log\left(\frac{\alpha d}{L}\right))$$

Total computation cost per node is $\frac{|\Phi|}{L}(r^2 + \log(|\Phi|/LK)) + nKH = \frac{|\Phi|}{L}(r^2 + \log(|\Phi|/L^2)) + nL\log(\alpha d/L)$.

Thus, with a KLL sketch of size $\mathcal{O}(L)$ per row, the corruption tolerance α and the sample complexity npd are unchanged from the centralized setting and no extra assumptions are needed. A proof outline is in Sec. 3; proofs are in Appendices A and C.

Comparisons: Advantages and Limitations of our approach. Our result above shows that our proposed algorithm, gen-AltGDmin-KLL, is order-wise as fast as, r-times more communication-efficient than, and L-times more tolerant of outliers than the fastest existing solution, factorized GD Yi et al. (2016) or its naive distributed extension. For the robust PCA special case, only the third advantage remains. The r-times fewer iterations do not give r-times lower computation for robust LRMC since our LS step costs $|\boldsymbol{\Phi}|r^2$ per iteration versus $|\boldsymbol{\Phi}|r$ for a gradient step; empirically our method is still much faster (Fig. 1). A disadvantage of our approach compared to factGD is that it needs $\log(1/\epsilon)$ (and κ^2) times more samples for robust LRMC than what factGD needs. The reason is that we need to use sample splitting to prove our guarantees. We note here though that, to our best knowledge, sample-splitting is required for analyzing any iterative algorithm in which at least one of the updates involves a full minimization; see Jain et al. (2013); Cherapanamjeri et al. (2017); Nayer & Vaswani (2023); Abbasi & Vaswani (2025). A second disadvantage of our approach is a worse dependence on the condition number κ. This is a feature of minimization-based alternating schemes. *Remark (κ dependence).* Our κ dependence is not good, but can be reduced significantly by replacing the GD step for $\boldsymbol{U}$ with a scaled GD step (pre-conditioning by $(\boldsymbol{BB}^\top)^{-1}$ as in ScaledGD Tong et al. (2021)) at no extra communication, since $\boldsymbol{BB}^\top$ is $r \times r$; extending our incoherence-propagation argument to it is left for future work.

We note that factGD requires more communication even if its federated extension were also given KLL-based thresholding: the KLL sketch only reduces the threshold-communication term, which is common to both algorithms. The r-times communication (and iteration-count) advantage of gen-AltGDmin-KLL comes from its larger, nearly constant, GD step size $\eta = c/p\sigma_{\max}^{*2}$; factGD's guarantee requires an r-times smaller step size and hence r-times more iterations, each of which communicates a gradient of the same order.

Gen-AltGDmin-KLL is L-times more outlier tolerant than the most direct local thresholding based federated extension of AltGDmin, and is n_{mx}/r times more communication efficient than the global-exact-thresholding based extension. See Table 1. A measured communication accounting is provided in Appendix C. How to set K, and the resulting communication–computation trade-off, are discussed in Appendix F.

3 Proof Outline

We discuss centralized case first and then federated one.

3.1 CENTRALIZED SETTING: PROOF OF COROLLARY 2.1

To make the ideas easy to follow with less notation, we first discuss the $p = 1$ special case.

Robust PCA ($p = 1$). Our goal is to show the following: if the fraction of sparse entries per row and per column is at most $\alpha \lesssim c_{\kappa,\mu}/r\sqrt{r}$, and if the step size $\eta \leq c/\sigma_1^{*2}$, then, the subspace recovery error, $\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}) := \|(\boldsymbol{I} - \boldsymbol{U}^*\boldsymbol{U}^{*\top})\boldsymbol{U}\|_F$ in estimating the column span of $\boldsymbol{U}^*$ decays as

$$\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}_t) \leq \left(1 - \frac{(0.8 - c_1)c_\eta}{\kappa^2}\right)^t c_0.$$

The initialization step and its guarantee are an easy modification of the result proved in Yi et al. (2016). This shows that, under the upper bound on α, $\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}) \leq c_0$. For analyzing the iterations, we use induction. At iteration t, assume that $\|\boldsymbol{U}^\top(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \leq \frac{c_1}{2}\delta_t\sigma_1^*$ and $\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}) \leq \delta_t$ for a δ_t small enough. Assume also that $\boldsymbol{U}$ satisfies μ_t row incoherence with $\mu_t \leq \mu_u := C\kappa^2\mu$.

The flow of the argument is: the bound on $\|\boldsymbol{U}^\top(\boldsymbol{S} - \boldsymbol{S}^*)\|_F$ gives the bound on $\|\boldsymbol{B} - \boldsymbol{G}\|_F$ (the $\boldsymbol{B}$ update step below); this gives the bounds on $\|\boldsymbol{S}^+ - \boldsymbol{S}^*\|_F$ and $\|(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top\|_F$ (the $\boldsymbol{S}$ update step); these give the subspace error decay for $\boldsymbol{U}^+$ (the $\boldsymbol{U}$ update step) and the row incoherence of $\boldsymbol{U}^+$; and the last two, in turn, give the bound on $\|\boldsymbol{U}^{+\top}(\boldsymbol{S}^+ - \boldsymbol{S}^*)\|_F$, i.e., the induction hypothesis at iteration $t + 1$.

Analyzing $\boldsymbol{B}$ update. For robust PCA, $\boldsymbol{B} = \boldsymbol{U}^\top(\boldsymbol{Y} - \boldsymbol{S})$. Using the induction assumption, $\|\tilde{\boldsymbol{B}} - (\boldsymbol{U}^\top\boldsymbol{U}^*)\boldsymbol{B}^*\|_F = \|\boldsymbol{U}^\top(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \leq \delta_t\sigma_1^*$. Using the non-expansive property of the column incoherent projection and that fact that $\boldsymbol{B}^*$ is column incoherent, this then implies that $\|\boldsymbol{B} - (\boldsymbol{U}^\top\boldsymbol{U}^*)\boldsymbol{B}^*\|_F$ satisfies the same bound. Thus, by triangle inequality and the definition of SD, $\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_F = \|\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*\|_F \leq 2\delta_t\sigma_1^*$. Finally, by using simple ideas from Vaswani (2024), this implies that $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma_r^*$ and $\sigma_{\max}(\boldsymbol{B}) \leq 1.1\sigma_1^*$. For robust LRMC, more work is needed, see Lemma A.4 in Appendix.

Incoherence of $\boldsymbol{B}$. This holds by construction. See Lemma A.4, first part, in Appendix.

Analyzing $\boldsymbol{S}$ update. We use the above bounds and incoherence to bound $\|(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top\|_F$ by $c\,\delta_t\sigma_r^{*2} = \frac{c}{\kappa^2}\delta_t\sigma_1^{*2}$ for a $c < 1$. This is done by carefully adapting ideas from Yi et al. (2016). We first show that $\|\boldsymbol{S}^+ - \boldsymbol{S}^*\|_F \leq c_2\,\delta_t\sigma_1^*$. The product $(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top$ is then bounded *directly*. Since each row of $\boldsymbol{S}^+ - \boldsymbol{S}^*$ has at most $(\gamma + 1.5)\alpha d$ nonzero entries, and $\boldsymbol{B}$ is column incoherent with $\|\boldsymbol{b}_k\| \leq \mu\sqrt{r/d}\,\sigma_1^*$; Cauchy-Schwarz then gives $\|(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top\|_F \leq \sqrt{(\gamma + 1.5)\alpha r}\,\mu\sigma_1^*\,\|\boldsymbol{S}^+ - \boldsymbol{S}^*\|_F$. Bounding the product directly is what allows a *constant* $\gamma = C$. With $\alpha \leq c/\kappa^8\mu^2 r$, we get the desired bound. See Lemma A.3 in Appendix.

Analyzing $\boldsymbol{U}$ update. Using the approach from Vaswani (2024) and above bounds, we can bound $\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}^+)$ as follows. Since $\nabla_{\boldsymbol{U}} f = (\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^* + \boldsymbol{S} - \boldsymbol{S}^*)\boldsymbol{B}^\top$ and $\Pi_{\boldsymbol{U}^*,\perp}\boldsymbol{U}^* = \boldsymbol{0}$, we get

$$\begin{aligned}
\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}^+) \leq & \frac{\delta_t\|\boldsymbol{I} - \eta\boldsymbol{B}\boldsymbol{B}^\top\| + \eta\|(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top\|_F}{1 - \eta\|(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*)\boldsymbol{B}^\top\| - \eta\|(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top\|} \\
\leq & \frac{\delta_t(1 - 0.9c_\eta/\kappa^2) + 1.1c_1(c_\eta/\kappa^2)}{1 - 2(c_\eta/\kappa^2)\delta_t - 1.1c_1(c_\eta/\kappa^2)\delta_t} \leq \delta_t\left(1 - \frac{(0.8 - c_1)c_\eta}{\kappa^2}\right) := \delta_{t+1}
\end{aligned}$$

if $\eta = c_\eta/\sigma_1^{*2}$ with $c_\eta \leq 0.5$ and if $\delta_t \leq \delta_0 \leq 0.1$ (used in denominator only). This bound on η and $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma_r^*$ from step 1 helps show that $(\boldsymbol{I} - \eta\boldsymbol{B}\boldsymbol{B}^\top)$ is psd and so $\|\boldsymbol{I} - \eta\boldsymbol{B}\boldsymbol{B}^\top\| = \lambda_{\max}(\boldsymbol{I} - \eta\boldsymbol{B}\boldsymbol{B}^\top) = 1 - \eta\sigma_{\min}(B)^2 \leq 1 - \eta 0.9\sigma_r^{*2} = 1 - 0.9c_\eta/\kappa^2$. See Lemma A.2.

Showing Incoherence of updated $\boldsymbol{U}$. The most difficult step is to show incoherence of $\boldsymbol{U}^+$. The approach introduced in Yi et al. (2016), that involves enforcing this by construction using $\Pi_{\mu,\mathrm{row}}(\boldsymbol{U}^+)$, cannot be combined with the above subspace error bounding approach that is critical for proving convergence with a constant GD step size η We instead need to show incoherence of $\boldsymbol{U}^+$ at each iteration. To show this, we obtain a recursion for the incoherence parameter of $\boldsymbol{U}$ at iteration $(t+1)$, μ_{t+1} in terms of that at iteration t, μ_t. The recursion needs to be such that $\mu_t \leq \mu_u$ can be proved.

Obtaining such a recursion requires a small enough bound on the maximum row norm of $(\boldsymbol{S}^+ - \boldsymbol{S}^*)\boldsymbol{B}^\top$. This requires realizing that, when showing incoherence, we do not need the bound to contain δ_t. But, we do need the bound to contain a factor of the form $\sqrt{r/n}$. To get such a

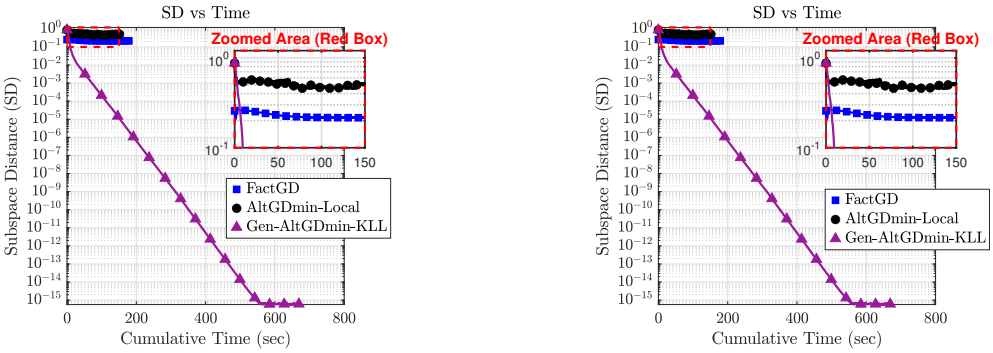
bound, it helps to use the seemingly loose scalar bound $|x - y| \leq 2\max(|x|, |y|)$. Using this and some ideas from (Yi et al., 2016, Sec 6.1), we can show that $\|\boldsymbol{S}^+ - \boldsymbol{S}^*\|_\infty \leq 2\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_\infty \leq 2\max(\|\boldsymbol{\Theta}\|_\infty, \|\boldsymbol{\Theta}^*\|_\infty) \leq 4\mu_t^2(r/\sqrt{nd})\sigma_1^*$. The last bound uses incoherence of $\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{U}^*, \boldsymbol{B}^*$. Here $\|.\|_\infty$ denotes the maximum magnitude of the matrix entries. Using the fact that any row of $(\boldsymbol{S}^+ - \boldsymbol{S}^*)$ contains at most $\gamma\alpha d$ nonzero entries, the maximum row norm of $(\boldsymbol{S} - \boldsymbol{S}^*)\boldsymbol{B}^\top$ is bounded by $\sqrt{\gamma\alpha d} \cdot 4\mu_t^2(r/\sqrt{nd})\sigma_1^{*2}$. Using $\gamma = C$ and the bound on α, we get the desired bound. See Lemmas A.5, A.6.

Induction Claim Proof Complete. Finally, we can show that the updated $\boldsymbol{S}^+$ and $\boldsymbol{U}^+$ satisfy $\|\boldsymbol{S}^+ - \boldsymbol{S}^*\|_F \leq c_2\,\delta_t\sigma_1^*$ and $\mathrm{SD}(\boldsymbol{U}^*, \boldsymbol{U}) \leq (1 - (0.8 - c_1)/\kappa^2)\delta_t := \delta_{t+1}$. Since $\delta_{t+1} \geq 0.2\,\delta_t$, using the per-column support bound and the row incoherence of $\boldsymbol{U}^+$ (same Cauchy–Schwarz idea as in the $\boldsymbol{S}$ update step), $\|\boldsymbol{U}^{+\top}(\boldsymbol{S}^+ - \boldsymbol{S}^*)\|_F \leq \frac{c_1}{2}\,\delta_{t+1}\sigma_1^*$, i.e., the induction hypothesis holds at iteration $t+1$. Also, $\boldsymbol{U}^+$ is μ_{t+1} incoherent with $\mu_{t+1} \leq \mu_t(1 - c_2/\kappa^2) + \mu_t(c_3/\kappa^2) + \mu$ with $c_3 < c_2$. Solving this recursion gives $\mu_t \leq \mu_u = C\kappa^2\mu$.

Robust LRMC ($p \leq 1$). The proof for this case is an easy modification of the above ideas combined with some results borrowed and modified from Abbasi & Vaswani (2025) and standard concentration bounds Vershynin (2018).

3.2 Proving Theorem 2.2 for Gen-AltGDmin-KLL: extra steps

We need a simple result, Lemma C.3 (Appendix C), along with ideas from above and from Karnin et al. (2016). Applying (Karnin et al., 2016, Corollary 1), we show in Lemma C.2 (Appendix C) that with $K = CL\sqrt{\log(L/\delta)}$ and $H = 1 + \lceil \log(\alpha d/K) \rceil$, the order of the estimated row threshold lies in $[0.9, 1.1]\,\gamma\alpha pd$ w.p. at least $1 - \delta$, at communication cost $\mathcal{O}(KH)$ per node. The two-sided guarantee changes only the constants in Lemma A.3 (kept count $\leq 1.1\gamma\alpha pd$; magnitudes below the $0.9\gamma\alpha pd$-th entry), so $\gamma = C$ still suffices and Corollary 2.1 applies. A union bound over n rows and T iterations with $\delta = 1/nd\kappa^2\log(1/\epsilon)$ gives total KLL communication $TnKH = \mathcal{O}(nL\log(1/\epsilon))$ per node; see Appendix C.



(a) Uniform data distribution across nodes

(b) Unequal data distribution across nodes

Figure 1: $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ *vs Time(seconds).*

4 Experiments

Simulation Experiment. We compare AltGDmin (local thresholding), Gen-AltGDmin-KLL (proposed), and FactGD Yi et al. (2016) in a federated setting with $L = 8$ nodes, where each node is implemented as one core of a multi-core CPU. We set $n = d = 1000$, $r = 10$, $\alpha = 0.1$, $p = 0.6$, $\eta = 0.54/(p\sigma^2_{\max}(\boldsymbol{\Theta}^*))$, $\gamma = 3$, and $K = 100$, and plot the average subspace distance $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ against time (in seconds) over 100 Monte Carlo runs. Figure 1a uses an equal split with $|\mathcal{S}_\ell| = 125$ for all $\ell \in [8]$, while Figure 1b uses a highly skewed split where Nodes 1 to 4 contain 1 column each and Nodes 5 to 8 contain 249 columns each. Figure 1b illustrates the remark in Sec. 1.1: four nodes hold a single column of $\boldsymbol{Y}$, so no per-node row threshold is meaningful, yet Gen-AltGDmin-KLL converges exactly as in Figure 1a because the merged KLL sketch recovers the global row order statistics for any column split. See Appendix E.1–E.3 for more details and experiments.

REPRODUCIBILITY STATEMENT

We have made several efforts to ensure the reproducibility of this work. All modeling assumptions are stated formally in Sec. 1.1, and complete proofs of every claim, including all lemmas, the main induction argument, and the analysis of the KLL-based thresholding, are provided in Appendices A and C. The algorithms are specified in full pseudo-code in the main text and appendix, together with all parameter choices used in our theory and experiments. Anonymous source code implementing all algorithms, along with the real dataset used, is available at `https://anonymous.4open.science/r/code-1910/`; the synthetic data generation procedure and all experimental settings (matrix dimensions, rank, corruption levels, sampling probability, number of nodes, and sketch parameters) are described in the experiments section and in Appendix E.1.

ETHICS STATEMENT

This paper is theoretical and presents work whose goal is to advance the field of Machine Learning. We do not foresee societal risks specific to this work.

AI USE STATEMENT

The authors acknowledge the use of generative AI for assistance with the grammatical corrections, and improving the clarity of the manuscript. Our use of AI did not impact the core methodology, scientific rigor, or originality of our research.

REFERENCES

Ahmed Ali Abbasi and Namrata Vaswani. Efficient federated low rank matrix completion. *IEEE Transactions on Information Theory*, 2025.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

Xi Chen, Jason D Lee, He Li, and Yun Yang. Distributed estimation for principal component analysis: a gap-free approach. *arXiv preprint arXiv:2004.02336*, 2020.

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

Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain. Nearly optimal robust matrix completion. In *International Conference on Machine Learning*, pp. 797–805. PMLR, 2017.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

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

Michael Greenwald and Sanjeev Khanna. Space-efficient online computation of quantile summaries. *ACM SIGMOD Record*, 30(2):58–66, 2001.

Quanquan Gu, Zhaoran Wang Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In *Artificial Intelligence and Statistics*, pp. 600–609. PMLR, 2016.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

Long-Kai Huang and Sinno Pan. Communication-efficient distributed pca by riemannian optimization. In *International Conference on Machine Learning*, pp. 4465–4474. PMLR, 2020.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, 2013.

Zohar Karnin, Kevin Lang, and Edo Liberty. Optimal quantile approximation in streams. In *2016 ieee 57th annual symposium on foundations of computer science (focs)*, pp. 71–78. IEEE, 2016.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Mehdi Makni, Kayhan Behdin, Zheng Xu, Natalia Ponomareva, and Rahul Mazumder. A unified framework for sparse plus low-rank matrix decomposition for llms. In *Conference on Parsimony and Learning*, Proceedings of Machine Learning Research, pp. 484–499. PMLR, 2025.

Praneeth Narayanamurthy and Namrata Vaswani. Provable dynamic robust pca or robust subspace tracking. *IEEE Transactions on Information Theory*, 65(3):1547–1577, 2018.

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

Praneeth Netrapalli, Niranjan UN, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. *Advances in neural information processing systems*, 27, 2014.

Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv preprint arXiv:2401.04679*, 2024.

Boxiao Pan, Zhan Xu, Chun-Hao Huang, Krishna Kumar Singh, Yang Zhou, Leonidas J Guibas, and Jimei Yang. Actanywhere: Subject-aware video background generation. *Advances in Neural Information Processing Systems*, 37:29754–29776, 2024.

Naichen Shi and Raed Al Kontar. Personalized pca: Decoupling shared and unique features. *Journal of machine learning research*, 25(41):1–82, 2024.

Ankit Pratap Singh and Namrata Vaswani. Byzantine-resilient federated principal subspace estimation. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pp. 2514–2519. IEEE, 2024a.

Ankit Pratap Singh and Namrata Vaswani. Byzantine-resilient federated pca and low-rank column-wise sensing. *IEEE Transactions on Information Theory*, 2024b.

Ankit Pratap Singh, Ahmed Ali Abbasi, and Namrata Vaswani. Byzantine-resilient federated alternating gradient descent and minimization for partly-decoupled low rank matrix learning. *International Conference on Machine Learning*, 2025.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

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

Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE signal processing magazine*, 35(4):32–55, 2018.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pp. 2080–2088, 2009.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust pca via gradient descent. *Advances in neural information processing systems*, 29, 2016.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11), 2024.

A PROVING COROLLARY 2.1

A.1 MAIN LEMMAS

The following lemma is a direct modification of (Yi et al., 2016, Theorem 3).

Lemma A.1 (Initialization). *Consider the initialization part of Algorithm 1. Let* $n_{mx} = \max(n, d)$. *Assume that Assumptions 1, 3, 2 hold. For* $\delta_0 = \frac{c}{\kappa^2}$, $\alpha \leq \frac{c}{\mu^2 r\sqrt{r}\kappa^3}$,*and* $npd \geq C\mu^2\kappa^4 n_{mx} r^2 \log n_{mx}$ *then w.p. at least* $1 - \mathcal{O}(d^{-1})$ *we have*

1. $\mathrm{SD}_F(\boldsymbol{U}_0, \boldsymbol{U}^*) \leq \delta_0$.
2. $\boldsymbol{U}_0$ *is* 1.5μ *row-incoherent, i.e.,* $\|\boldsymbol{u}_0^j\| \leq 1.5\mu\sqrt{r/n}$ *for all* $j \in [n]$.
3. $\|\boldsymbol{U}_0^\top \Pi_{\boldsymbol{\Phi}}(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F \leq \frac{c_1}{4}\, p\, \delta_0 \sigma_r^*$.

Lemma A.2 (Algebra Lemma). *For* $[\boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+)] \overset{QR}{=} \boldsymbol{U}^+\boldsymbol{R}^+$ *we have*

$$\boldsymbol{SD}(\boldsymbol{U}^*, \boldsymbol{U}^+) \leq \frac{\delta_t\|I_r - \eta p \boldsymbol{B}\boldsymbol{B}^\top\| + \eta\|\mathrm{Err}\|_F + \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F}{1 - \eta\|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| - \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|} \tag{4}$$

Here $\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+) = (\Pi_\Phi((\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*) + (\boldsymbol{S}^+ - \boldsymbol{S}^*)))\boldsymbol{B}^\top$, *and* $\mathrm{Err} = (\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top$.

Proof. Proof is in Appendix B.2. □

Lemma A.3 (Bounding Sparse Estimation Error). *Assume 1)* $\boldsymbol{SD}_F(\boldsymbol{U}, \boldsymbol{U}^*) \leq \delta_t$, *2)* $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta_t\sigma_1^*$, *3)* $\|\boldsymbol{b}_k\| \leq \mu\sigma_1^*\sqrt{\frac{r}{d}}$, *4)* $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{\frac{r}{n}}$ *for all* $j \in [n]$. *Set* $\gamma = C > 1.5$. *If* $p \geq \frac{C\mu_u^2\kappa^4 r^2 \log d}{n}$, *and* $\alpha \leq \frac{c}{\kappa^8\mu^2 r}$ *then w.p. at least* $1 - \mathcal{O}(d^{-1})$

$$\|\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)\|_F \leq c_2\sqrt{p}\,\delta_t\sigma_1^* \;\; \textit{and} \;\; ||(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top||_F \leq c\,p\,\delta_t\sigma_r^{*2}$$

Proof. Proof is in Appendix B.3. □

Lemma A.4 (Bounds for B). *Assume* $\|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \leq \frac{c_1}{2}\, p\, \delta_t\sigma_1^*$, $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$, *and* $\delta_t \leq \delta_0$. *If* $p \geq \frac{C\mu_u^2\kappa^4 r^2 \log d}{n}$ *then w.p. at least* $1 - \mathcal{O}(d^{-1})$

1. $\|\boldsymbol{b}_k\| \leq \mu\sqrt{\frac{r}{d}}\sigma_1^*$ *for all* $k \in [d]$
2. $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq c_1\delta_t\sigma_1^*$, *where* $\boldsymbol{G} = \boldsymbol{U}^\top\boldsymbol{\Theta}^*$
3. $\|\boldsymbol{B}\| \leq 1.1\sigma_1^*$
4. $\sigma_{\min}(\boldsymbol{B}^\top) \geq 0.9\sigma_r^*$
5. $\|\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*\|_F \leq 2\delta_t\sigma_1^*$ *and* $\|\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\|_F \leq 2.6\sqrt{p}\,\delta_t\sigma_1^*$
6. $\|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| \leq 2.6\,p\,\delta_t\sigma_1^{*2}$

Proof. Most of this proof is similar to that given in (Abbasi & Vaswani, 2025, Lemma 4.2). The main change here is that, in this algorithm, we need to impose incoherence of $\boldsymbol{B}$ Proof is in Appendix B.4. □

Lemma A.5 (Row Incoherence of Sparse Estimation Error). *Assume* $\delta_t \leq \delta_0$, $\|\boldsymbol{b}_k\| \leq \mu\sigma_1^*\sqrt{r/d}$ *for all* $k \in [d]$, *and* $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$ *for all* $j \in [n]$. *If* $p \geq \frac{C\mu^2\kappa^4 r^2 \log d}{n}$, $\gamma > 1.5$, *and* $\alpha \leq \frac{c_2}{C(\gamma+1.5)\mu^2 r\kappa^2}$ *then w.p. at least* $1 - \mathcal{O}(d^{-1})$

$$\|\boldsymbol{e}_j^\top((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)\| \leq c_2 p\Big(\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \mu_u\sqrt{r/n}\Big)\sigma_r^{*2}$$

Proof. Proof is in Appendix B.6. □

Lemma A.6 (Incoherence of $\boldsymbol{U}$)**.** *Consider the setting of Lemma A.5. For* $[\boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+)] \overset{QR}{=} \boldsymbol{U}^+\boldsymbol{R}^+$*, we have* $\|\boldsymbol{u}^{+j}\| \le (1 - \frac{0.25}{\kappa^2})\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\| + \frac{0.05}{\kappa^2}\mu_u\sqrt{r/n}$

Proof. Proof is in Appendix B.7. □

Lemma A.7 (Cross Term Bound for the LS step)**.** *Let* $\boldsymbol{S}$ *be the sparse estimate computed on the current LS subset* $\boldsymbol{\Phi}$ *using* $(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{t-1})$*. Assume* $\|\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \le c_2\sqrt{p}\,\delta_{t-1}\sigma_1^*$ *with* $\boldsymbol{S} - \boldsymbol{S}^*$ *having at most* $(\gamma + 1.5)\alpha pn$ *nonzeros per column,* $\|\boldsymbol{u}^j\| \le \mu_u\sqrt{r/n}$ *for all* $j \in [n]$ *(for the current* $\boldsymbol{U} = \boldsymbol{U}_t$*), and* $\delta_{t-1} \le 5\delta_t$*. Set* $\gamma = C > 1.5$*. If* $\alpha \le \frac{c\,c_1^2}{c_2^2\,\mu_u^2 r}$ *then w.p. at least* $1 - \mathcal{O}(d^{-1})$

$$\|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \le \frac{c_1}{2}\,p\,\delta_t\sigma_1^*.$$

Proof. Proof is in Appendix B.5. □

A.2 Proof of Corollary 2.1

We prove the theorem using induction for which we give the Claim A.8.

Claim A.8. Let $\mu_u := (1.5 + 10\kappa^2)\mu$. Assume that $\delta_0 = \frac{c}{\kappa^2}$, $\alpha \le \min\Big(\frac{c}{\mu_u^2 r\kappa^4}, \frac{c}{\mu^2 r^{3/2}\kappa^3}\Big)$, $p \ge \frac{C\mu_u^2\kappa^4 r^2\log d}{n}$. Then w.p. at least $1 - \mathcal{O}(d^{-1})$

(i) $\delta_t \le (1 - \frac{0.5\eta p\sigma_1^{*2}}{\kappa^2})^t\delta_0$,

(ii) $\|\boldsymbol{u}_t^j\| \le \mu_u\sqrt{\frac{r}{n}}$ for all $j \in [n]$.

Proof. Base case ($t \equiv 0$): By Lemma A.1, $\boldsymbol{SD}_F(\boldsymbol{U}_0, \boldsymbol{U}^*) \le \delta_0$, $\|\boldsymbol{u}_0^j\| \le 1.5\mu\sqrt{r/n} \le \mu_u\sqrt{r/n}$, and $\|\boldsymbol{U}_0^\top\Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F \le \frac{c_1}{4}\,p\,\delta_0\sigma_r^* \le \frac{c_1}{2}\,p\,\delta_0\sigma_1^*$ (for $t = 1$ the LS subset uses $\boldsymbol{S} = \mathcal{T}_{2\alpha p}(\Pi_{\boldsymbol{\Phi}_1^B}(\boldsymbol{Y}))$, and Lemma A.1 part 3 applies since $\boldsymbol{U}_0$ is independent of $\boldsymbol{\Phi}_1^B$).

Induction assumption: Assume that the Claim A.8 holds for iteration t, i.e., $\|\boldsymbol{u}^j\| \le \mu_u\sqrt{\frac{r}{n}}$ for all j, $\delta_t \le \delta_0$, and, from the previous iteration, $\|\boldsymbol{B}_{t-1} - \boldsymbol{U}_{t-1}^\top\boldsymbol{\Theta}^*\|_F \le c_1\delta_{t-1}\sigma_1^*$ with $\boldsymbol{B}_{t-1}$ column incoherent and $\delta_{t-1} \le 5\delta_t$ (by (13)).

Induction proof: We now use the induction assumption to prove the claim for $t + 1$.

LS subset $\boldsymbol{\Phi}_t^B$. The pair $(\boldsymbol{U}_{t-1}, \boldsymbol{B}_{t-1})$ was fixed before $\boldsymbol{\Phi}_t^B$ was drawn, so Lemma A.3 applies to $\boldsymbol{S} = \mathcal{T}_{\gamma\alpha p}(\Pi_{\boldsymbol{\Phi}_t^B}(\boldsymbol{Y} - \boldsymbol{U}_{t-1}\boldsymbol{B}_{t-1}))$ with δ_{t-1} in place of δ_t, giving $\|\Pi_{\boldsymbol{\Phi}_t^B}(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \le c_2\sqrt{p}\,\delta_{t-1}\sigma_1^*$ and the per-column support bound. Since $\|\boldsymbol{u}^j\| \le \mu_u\sqrt{r/n}$ for the current $\boldsymbol{U} = \boldsymbol{U}_t$ and $\delta_{t-1} \le 5\delta_t$, Lemma A.7 gives $\|\boldsymbol{U}^\top\Pi_{\boldsymbol{\Phi}_t^B}(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \le \frac{c_1}{2}\,p\,\delta_t\sigma_1^*$. Also $\boldsymbol{U} = \boldsymbol{U}_t$ is independent of $\boldsymbol{\Phi}_t^B$.

Using this cross-term bound, $\|\boldsymbol{u}^j\| \le \mu_u\sqrt{r/n}$ for all $j \in [n]$, and $\delta_t \le \delta_0$, Lemma A.4 holds: $\|\boldsymbol{b}_k\| \le \mu\sqrt{\frac{r}{d}}\sigma_1^*$, $\|\boldsymbol{B} - \boldsymbol{G}\|_F \le c_1\delta_t\sigma_1^*$, $\|\boldsymbol{B}\| \le 1.1\sigma_1^*$, and $\sigma_{\min}(\boldsymbol{B}^\top) \ge 0.9\sigma_r^*$.

GD subset $\boldsymbol{\Phi}_t^U$. The pair $(\boldsymbol{U}_t, \boldsymbol{B}_t)$ was fixed before $\boldsymbol{\Phi}_t^U$ was drawn, so the concentration bounds of Lemma A.4 items 5–6 and Lemma A.3 apply on $\boldsymbol{\Phi}_t^U$ to $\boldsymbol{S}^+ = \mathcal{T}_{\gamma\alpha p}(\Pi_{\boldsymbol{\Phi}_t^U}(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B}))$; all quantities below are on $\boldsymbol{\Phi}_t^U$. This then implies that Lemma A.3 holds. Thus, we have

$$\|\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)\|_F \le c_2\sqrt{p}\,\delta_t\sigma_1^* \;\text{ and }\; \|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F \le c\,p\,\delta_t\sigma_r^{*2}.$$

Next we use these to bound $\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}^+)$. From Lemma A.2 we have

$$\boldsymbol{SD}_F(\boldsymbol{U}^*, \boldsymbol{U}^+) \le \frac{\delta_t\|I_r - \eta p\boldsymbol{B}\boldsymbol{B}^\top\| + \eta\|\mathrm{Err}\|_F + \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F}{1 - \eta\|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| - \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|}. \tag{5}$$

Using $\|\boldsymbol{B}\| \le 1.1\sigma_1^*$ from Lemma A.4,

$$\lambda_{\min}(\boldsymbol{I}_r - \eta p \boldsymbol{B}\boldsymbol{B}^\top) = 1 - \eta p \|\boldsymbol{B}\|^2 \ge 1 - 1.2\eta p \sigma_1^{*2}$$

Thus, if $\eta \le 0.5/p\sigma_1^{*2}$, then the above matrix is p.s.d. This implies that

$$\|\boldsymbol{I}_r - \eta p \boldsymbol{B}\boldsymbol{B}^\top\| = \lambda_{\max}(\boldsymbol{I} - \eta p \boldsymbol{B}\boldsymbol{B}^\top) \le 1 - 0.8\eta p \sigma_r^{*2}$$

Now

$$\delta_t \|I_r - \eta p \boldsymbol{B}\boldsymbol{B}^\top\| + \eta \|\mathrm{Err}\|_F + \eta \|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F \le (1 - 0.8\eta p \sigma_r^{*2})\delta_t + \eta \epsilon p \delta_t \sigma_r^{*2} + \eta c_1 p \delta_t \sigma_r^{*2} \tag{6}$$

$$= (1 - (0.8 - \epsilon - c_1)\eta p \sigma_r^{*2})\delta_t \tag{7}$$

Above we have used $\|\mathrm{Err}\|_F \le \epsilon p \delta_t \sigma_r^{*2}$ from (Abbasi & Vaswani, 2025, Lemma 4.5).

Consider the denominator term from the $\boldsymbol{S}\boldsymbol{D}_F$ bound. We have

$$(1 - \eta \|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| - \eta \|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|)^{-1} \le \big(1 - \eta 2.6 p \delta_t \sigma_1^{*2} - \eta p \delta_t \sigma_1^{*2}\big)^{-1} \tag{8}$$

$$= \big(1 - 3.6\delta_t \eta p \sigma_1^{*2}\big)^{-1} \tag{9}$$

$$= \big(1 - 0.1\eta p \sigma_r^{*2}\big)^{-1} \tag{10}$$

$$\le \big(1 + 0.2\eta p \sigma_r^{*2}\big) \tag{11}$$

Above we have used Lemma A.4, item 6, $\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\| \le c\,p\,\delta_t \sigma_r^{*2} \le p\,\delta_t \sigma_1^{*2}$ from Lemma A.3, and $\delta_t \le \delta_0 \le \frac{0.1}{3.6\kappa^2}$.

Thus, using $c_1 = 0.1/2$, $\epsilon = 0.1/2$,

$$\boldsymbol{S}\boldsymbol{D}_F(\boldsymbol{U}^*, \boldsymbol{U}^+) \le \big(1 + 0.2\eta p \sigma_r^{*2}\big)(1 - (0.8 - \epsilon - c_1)\eta p \sigma_r^{*2})\delta_t \le (1 - 0.5\eta p \sigma_r^{*2})\delta_t \tag{12}$$

This then implies Claim A.8 part (i) holds. Also, since $\eta = \frac{0.5}{p\sigma_1^{*2}}$,

$$\delta_{t+1} = \big(1 - 0.5\eta p \sigma_r^{*2}\big)\delta_t = \left(1 - \frac{0.25}{\kappa^2}\right)\delta_t \ge 0.2\,\delta_t. \tag{13}$$

Now using Lemma A.6and $\|\boldsymbol{u}^{*j}\| \le \mu\sqrt{r/n}$,

$$\|\boldsymbol{u}^{+j}\| \le (1 - \frac{0.25}{\kappa^2})\|\boldsymbol{u}^j\| + 2\|\boldsymbol{u}^{*j}\| + \frac{0.05}{\kappa^2}\mu_u\sqrt{\frac{r}{n}} \le (1 - \frac{0.25}{\kappa^2})\mu_u\sqrt{\frac{r}{n}} + 2\mu\sqrt{\frac{r}{n}} + \frac{0.05}{\kappa^2}\mu_u\sqrt{\frac{r}{n}}$$

$$= \mu_u\sqrt{\frac{r}{n}}\Big(1 - \frac{0.2}{\kappa^2}\Big) + 2\mu\sqrt{\frac{r}{n}} \le \mu_u\sqrt{\frac{r}{n}},$$

since $\mu_u \ge 10\kappa^2\mu$ gives $0.2\mu_u/\kappa^2 \ge 2\mu$. This then implies Claim A.8 part (ii) holds.

Finally, Lemma A.4 gives $\|\boldsymbol{B}_t - \boldsymbol{U}_t^\top \boldsymbol{\Theta}^*\|_F \le c_1\delta_t\sigma_1^*$ with $\boldsymbol{B}_t$ column incoherent, and (13) gives $\delta_t \le 5\delta_{t+1}$, which are the hypotheses needed at iteration $t+1$. □

B Proof of Lemmas used for Proving Corollary 2.1

B.1 Proof of Initialization Lemma A.1

Proof. Denote the support of $\Pi_\Phi(\boldsymbol{S}^*)$, $\boldsymbol{S}_0$ by $\boldsymbol{\Omega}_o^*$ and Ω respectively. Since $\boldsymbol{Y} = \Pi_\Phi(\boldsymbol{\Theta}^*) + \Pi_\Phi(\boldsymbol{S}^*)$ and $\boldsymbol{S}_0 = \mathcal{T}_{2\alpha p}(\boldsymbol{Y})$, we can write

$$p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^* \;=\; p^{-1}\big(\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0\big) \;+\; \big(p^{-1}\Pi_\Phi(\boldsymbol{\Theta}^*) - \boldsymbol{\Theta}^*\big).$$

Consider the first term. $\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0$ is supported on $\Omega \cup \boldsymbol{\Omega}_o^*$. By construction of $\mathcal{T}_{2\alpha p}$, $|\Omega^j| \le 2\alpha p d$ and $|\Omega_k| \le 2\alpha p n$; and, by (Yi et al., 2016, Lemma 10) (which needs $p \ge \frac{C\log d}{\alpha d}$), w.h.p.

$|\boldsymbol{\Omega}_o^{*j}| \le \frac{3}{2}\alpha pd$ and $|\boldsymbol{\Omega}_{ok}^*| \le \frac{3}{2}\alpha pn$. Thus each row (column) of $\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0$ has at most $3.5\alpha pd$ ($3.5\alpha pn$) nonzero entries. Also, as shown in the proof of (Yi et al., 2016, Theorem 3, eq. (28)),

$$\|\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0\|_\infty \le 2\|\boldsymbol{\Theta}^*\|_\infty \le \frac{2\mu^2 r}{\sqrt{nd}}\sigma_1^*. \tag{14}$$

Using (Yi et al., 2016, Lemma 1),

$$\|p^{-1}\big(\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0\big)\| \le \frac{1}{p}\cdot 3.5\alpha p\sqrt{nd}\,\|\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0\|_\infty \le 7\alpha\mu^2 r\sigma_1^*.$$

Consider the second term. By (Yi et al., 2016, Lemma 5), as shown in the proof of (Yi et al., 2016, Theorem 3, eq. (30)) if $p \ge \frac{4\mu r\log d}{\epsilon_0^2 d}$ then w.p. at least $1-\mathcal{O}(d^{-1})$, $\|p^{-1}\Pi_\Phi(\boldsymbol{\Theta}^*) - \boldsymbol{\Theta}^*\| \le c_0'\,\frac{\epsilon_0}{\sqrt{r}}\,\sigma_1^*$. Combining,

$$\|p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^*\| \le 7\alpha\mu^2 r\sigma_1^* + c_0'\,\frac{\epsilon_0}{\sqrt{r}}\,\sigma_1^*.$$

Setting $\epsilon_0 = \frac{c\,c_0}{\kappa}$ this holds w.p. at least $1-\mathcal{O}(d^{-1})$ if $p \ge \frac{C\kappa^2\mu r\log d}{c_0^2 d}$ and substituting $\alpha \le \frac{c\,c_0}{\mu^2 r^{3/2}\kappa}$, each of the two terms above is at most $\frac{c_0\sigma_r^*}{8\sqrt{2r}}$, and thus

$$\|p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^*\| \le \frac{c_0\,\sigma_r^*}{4\sqrt{2r}}.$$

Part 1. Using Wedin's $\sin\theta$ theorem for the top-r left singular vectors $\boldsymbol{U}_{00}$ of $p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0)$,

$$\boldsymbol{SD}_F(\boldsymbol{U}_{00}, \boldsymbol{U}^*) \le \frac{\sqrt{2r}\,\|p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^*\|}{\sigma_r^* - \|p^{-1}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^*\|} \le \frac{c_0/4}{1 - c_0/4} \le \frac{c_0}{3}.$$

Recall that $\mathrm{dist}(\boldsymbol{U}, \boldsymbol{U}^*) := \min_{\boldsymbol{R}:\boldsymbol{R}^\top\boldsymbol{R}=\boldsymbol{I}} \|\boldsymbol{U} - \boldsymbol{U}^*\boldsymbol{R}\|_F$ is also a commonly used definition of subspace distance, and the two definitions are equivalent up to a $\sqrt{2}$ factor: $\boldsymbol{SD}_F(\boldsymbol{U}, \boldsymbol{U}^*) \le \mathrm{dist}(\boldsymbol{U}, \boldsymbol{U}^*) \le \sqrt{2}\,\boldsymbol{SD}_F(\boldsymbol{U}, \boldsymbol{U}^*)$ (Chen et al., 2021, Lemmas 2.5, 2.6). Since $\boldsymbol{U}^*$ is μ-row-incoherent, $\Pi_{\mu,row}(\boldsymbol{U}^*\boldsymbol{R}) = \boldsymbol{U}^*\boldsymbol{R}$ for any unitary $\boldsymbol{R}$. Using the property that projection onto a convex set is non-expansive in Frobenius norm, with $\boldsymbol{R}_*$ denoting the minimizing rotation in $\mathrm{dist}(\boldsymbol{U}_{00}, \boldsymbol{U}^*)$, we have

$$\|\Pi_{\mu,row}(\boldsymbol{U}_{00}) - \boldsymbol{U}^*\boldsymbol{R}_*\|_F = \|\Pi_{\mu,row}(\boldsymbol{U}_{00}) - \Pi_{\mu,row}(\boldsymbol{U}^*\boldsymbol{R}_*)\|_F \le \mathrm{dist}(\boldsymbol{U}_{00}, \boldsymbol{U}^*) \le \sqrt{2}\,\boldsymbol{SD}_F(\boldsymbol{U}_{00}, \boldsymbol{U}^*) \le \frac{\sqrt{2}c_0}{3}. \tag{15}$$

Let $\Pi_{\mu,row}(\boldsymbol{U}_{00}) \overset{\text{QR}}{=} \boldsymbol{U}_0\boldsymbol{R}_0$. Since $\boldsymbol{U}_0$ has orthonormal columns, $\sigma_i(\boldsymbol{U}_0\boldsymbol{R}_0) = \sigma_i(\boldsymbol{R}_0)$ for all i, i.e., $\sigma_{\min}(\boldsymbol{R}_0) = \sigma_{\min}(\Pi_{\mu,row}(\boldsymbol{U}_{00}))$. Using Weyl's inequality for singular values, $|\sigma_{\min}(\boldsymbol{A}) - \sigma_{\min}(\boldsymbol{B})| \le \|\boldsymbol{A} - \boldsymbol{B}\|$, with $\boldsymbol{A} = \Pi_{\mu,row}(\boldsymbol{U}_{00})$, $\boldsymbol{B} = \boldsymbol{U}^*\boldsymbol{R}_*$, (15), and the fact that $\boldsymbol{U}^*\boldsymbol{R}_*$ has orthonormal columns,

$$\sigma_{\min}(\boldsymbol{R}_0) = \sigma_{\min}(\Pi_{\mu,row}(\boldsymbol{U}_{00})) \ge \sigma_{\min}(\boldsymbol{U}^*\boldsymbol{R}_*) - \|\Pi_{\mu,row}(\boldsymbol{U}_{00}) - \boldsymbol{U}^*\boldsymbol{R}_*\| \ge 1 - \frac{\sqrt{2}c_0}{3}. \tag{16}$$

Next, since $\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}^*\boldsymbol{R}_* = \boldsymbol{0}$ and $\|\mathcal{P}_{\boldsymbol{U}^*,\perp}\| = 1$,

$$\|\mathcal{P}_{\boldsymbol{U}^*,\perp}\Pi_{\mu,row}(\boldsymbol{U}_{00})\|_F = \|\mathcal{P}_{\boldsymbol{U}^*,\perp}\big(\Pi_{\mu,row}(\boldsymbol{U}_{00}) - \boldsymbol{U}^*\boldsymbol{R}_*\big)\|_F \le \|\Pi_{\mu,row}(\boldsymbol{U}_{00}) - \boldsymbol{U}^*\boldsymbol{R}_*\|_F \le \frac{\sqrt{2}c_0}{3}. \tag{17}$$

Thus, using $\boldsymbol{U}_0 = \Pi_{\mu,row}(\boldsymbol{U}_{00})\boldsymbol{R}_0^{-1}$, so that $\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}_0 = \mathcal{P}_{\boldsymbol{U}^*,\perp}\Pi_{\mu,row}(\boldsymbol{U}_{00})\,\boldsymbol{R}_0^{-1}$, and combining (16) and (17),

$$\boldsymbol{SD}_F(\boldsymbol{U}_0, \boldsymbol{U}^*) = \|\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}_0\|_F \le \|\mathcal{P}_{\boldsymbol{U}^*,\perp}\Pi_{\mu,row}(\boldsymbol{U}_{00})\|_F\,\|\boldsymbol{R}_0^{-1}\| \le \frac{\sqrt{2}c_0/3}{1 - \sqrt{2}c_0/3} \le c_0 = \delta_0.$$

This proves Part 1.

Part 2. Each row of $\Pi_{\mu,row}(\boldsymbol{U}_{00})$ has norm at most $\mu\sqrt{r/n}$ by construction, and $\|\boldsymbol{u}_0^j\| = \|\boldsymbol{e}_j^\top \Pi_{\mu,row}(\boldsymbol{U}_{00})\boldsymbol{R}_0^{-1}\| \le \mu\sqrt{r/n}\,\|\boldsymbol{R}_0^{-1}\| \le 1.5\mu\sqrt{r/n}$.

Part 3. $\Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*) = -(\Pi_\Phi(\boldsymbol{S}^*) - \boldsymbol{S}_0)$ is supported on $\Omega_f := \Omega \cup \boldsymbol{\Omega}_o^*$, with $|\Omega_f^j| \le 3.5\alpha pd$ and $|\Omega_{f,k}| \le 3.5\alpha pn$ w.h.p. (from above). For $jk \in \Omega$, $\boldsymbol{s}_{0,jk} = \boldsymbol{y}_{jk}$ and thus $(\boldsymbol{s}_0 - \boldsymbol{s}^*)_{jk} = \boldsymbol{\theta}^*_{jk}$, with $|\boldsymbol{\theta}^*_{jk}| \le \|\boldsymbol{u}^{*j}\|\,\|\boldsymbol{b}_k^*\| \le \|\boldsymbol{u}^{*j}\|\,\mu\sqrt{r/d}\,\sigma_1^*$. For $jk \in \boldsymbol{\Omega}_o^* \setminus \Omega$, $(\boldsymbol{s}_0 - \boldsymbol{s}^*)_{jk} = -\boldsymbol{s}^*_{jk}$ and $|\boldsymbol{s}^*_{jk}| \le |\boldsymbol{y}_{jk}| + |\boldsymbol{\theta}^*_{jk}|$. Since $jk \notin \Omega$, $|\boldsymbol{y}_{jk}| < \max\big(|\boldsymbol{y}^{j\,(2\alpha pd)}|,\, |\boldsymbol{y}_k{}^{(2\alpha pn)}|\big)$. Since $\boldsymbol{y}^j = \Pi_\Phi(\boldsymbol{\theta}^{*j} + \boldsymbol{s}^{*j})$ with $|\boldsymbol{\Omega}_o^{*j}| \le 1.5\alpha pd$, using Lemma B.1,

$$|\boldsymbol{y}^{j\,(2\alpha pd)}| \le |(\Pi_\Phi(\boldsymbol{\Theta}^*))^{j\,(0.5\alpha pd)}| \le \max_{k'} |\boldsymbol{\theta}^*_{jk'}| \le \|\boldsymbol{u}^{*j}\|\,\mu\sqrt{r/d}\,\sigma_1^* \;\text{ and, similarly, }\; |\boldsymbol{y}_k{}^{(2\alpha pn)}| \le \mu\sqrt{r/n}\,\|\boldsymbol{b}_k^*\|.$$

Thus, in both cases,

$$|(\boldsymbol{s}_0 - \boldsymbol{s}^*)_{jk}|^2 \le 4\big(\|\boldsymbol{u}^{*j}\|^2\mu^2\tfrac{r}{d}\sigma_1^{*2} + \mu^2\tfrac{r}{n}\|\boldsymbol{b}_k^*\|^2\big).$$

Using $\sum_j \|\boldsymbol{u}^{*j}\|^2 = r$ and $\sum_k \|\boldsymbol{b}_k^*\|^2 \le r\sigma_1^{*2}$,

$$\|\Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F^2 \le 4\sum_j |\Omega_f^j|\,\|\boldsymbol{u}^{*j}\|^2\mu^2\tfrac{r}{d}\sigma_1^{*2} + 4\sum_k |\Omega_{f,k}|\,\mu^2\tfrac{r}{n}\|\boldsymbol{b}_k^*\|^2 \le 28\,\alpha p\,\mu^2 r^2\,\sigma_1^{*2}. \tag{18}$$

Using Cauchy–Schwarz, Part 2, $|\Omega_{f,k}| \le 3.5\alpha pn$, and (18),

$$\|\boldsymbol{U}_0^\top \Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F^2 = \sum_k \|\textstyle\sum_{j:jk\in\Omega_f}(\boldsymbol{s}_0 - \boldsymbol{s}^*)_{jk}\,\boldsymbol{u}_0^j\|^2 \le \textstyle\sum_k |\Omega_{f,k}| \sum_{j:jk\in\Omega_f} |(\boldsymbol{s}_0 - \boldsymbol{s}^*)_{jk}|^2\,\|\boldsymbol{u}_0^j\|^2$$

$$\le 3.5\alpha pn\cdot(1.5\mu)^2\tfrac{r}{n}\,\|\Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F^2 \le \Big(15\,\alpha p\,\mu^2 r^{3/2}\,\sigma_1^*\Big)^2.$$

For $\alpha \le \frac{c_1\,\delta_0}{60\,\mu^2 r^{3/2}\kappa}$,

$$\|\boldsymbol{U}_0^\top \Pi_\Phi(\boldsymbol{S}_0 - \boldsymbol{S}^*)\|_F \le \frac{c_1}{4}\,p\,\delta_0\sigma_r^*.$$

□

B.2 Proof of Lemma A.2

Proof. We have $\nabla_{\boldsymbol{U}} f(\boldsymbol{U},\boldsymbol{B},\boldsymbol{S}^+) = (\Pi_\Phi((\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*) + (\boldsymbol{S}^+ - \boldsymbol{S}^*)))\boldsymbol{B}^\top = (\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top + (\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top$.

GD step is given as $\hat{\boldsymbol{U}}^+ = \boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U},\boldsymbol{B},\boldsymbol{S}^+)$. Adding and subtracting $\eta p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top$ we get

$$\hat{\boldsymbol{U}}^+ = \boldsymbol{U} - \eta p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top - \eta((\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top) - \eta(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top.$$

Let $\mathcal{P}_{\boldsymbol{U}^*,\perp} := \boldsymbol{I} - \boldsymbol{U}^*\boldsymbol{U}^{*T}$. Multiplying both sides by $\mathcal{P}_{\boldsymbol{U}^*,\perp}$, and using the fact that $\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}^*\boldsymbol{B}^* = \boldsymbol{0}$, we get

$$\begin{aligned}\mathcal{P}_{\boldsymbol{U}^*,\perp}\hat{\boldsymbol{U}}^+ &= \mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U} - \eta p\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}\boldsymbol{B}\boldsymbol{B}^\top - \eta\mathcal{P}_{\boldsymbol{U}^*,\perp}\mathrm{Err} - \eta\mathcal{P}_{\boldsymbol{U}^*,\perp}(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top \\ &= \mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}(\boldsymbol{I}_r - \eta p\boldsymbol{B}\boldsymbol{B}^\top) - \eta\mathcal{P}_{\boldsymbol{U}^*,\perp}\mathrm{Err} - \eta\mathcal{P}_{\boldsymbol{U}^*,\perp}(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top \end{aligned} \tag{19}$$

Here $\mathrm{Err} = (\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top$. Taking Frobenius norm and using $\|M_1M_2\|_F \le \|M_1\|_F\|M_2\|$, $\|M_1M_2\|_F \le \|M_1\|\|M_2\|_F$, and $\|\mathcal{P}_{\boldsymbol{U}^*,\perp}\| = 1$ we get

$$\|\mathcal{P}_{\boldsymbol{U}^*,\perp}\hat{\boldsymbol{U}}^+\|_F \le \|\mathcal{P}_{\boldsymbol{U}^*,\perp}\boldsymbol{U}\|_F\|I_r - \eta p\boldsymbol{B}\boldsymbol{B}^\top\| + \eta\|\mathrm{Err}\|_F + \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F \tag{20}$$

Now $\hat{\boldsymbol{U}}^+ \overset{QR}{=} \boldsymbol{U}^+R^+$, this means that $\boldsymbol{S}\boldsymbol{D}_F(\boldsymbol{U}^*,\boldsymbol{U}^+) \le \|\mathcal{P}_{\boldsymbol{U}^*,\perp}\hat{\boldsymbol{U}}^+\|_F\|(R^+)^{-1}\|$. Since $\|(R^+)^{-1}\| = 1/\sigma_{min}(R^+) = 1/\sigma_{min}(\hat{\boldsymbol{U}}^+)$, we get

$$\begin{aligned}\|(R^+)^{-1}\| &= \frac{1}{\sigma_{min}(\boldsymbol{U} - \eta(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - \eta(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)} \\ &\le \frac{1}{1 - \eta\|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| - \eta\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|}\end{aligned}$$

Combining the last two bounds we get,

$$\boldsymbol{SD}(\boldsymbol{U}^*, \boldsymbol{U}^+) \leq \frac{\delta_t \|I_r - \eta p \boldsymbol{B}\boldsymbol{B}^\top\| + \eta \|\mathrm{Err}\|_F + \eta \|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F}{1 - \eta \|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| - \eta \|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|} \tag{21}$$

□

B.3 Proof of Lemma A.3

Proof. Denote the support of $\Pi_\Phi(\boldsymbol{S}^*), \boldsymbol{S}$ by $\boldsymbol{\Omega}^*, \Omega$ respectively. By default, $\boldsymbol{\Omega}^* \subseteq \boldsymbol{\Phi}$ and $\Omega \subseteq \boldsymbol{\Phi}$. Recall that,

$$\boldsymbol{S}^+ = \mathcal{T}_{\gamma\alpha p}(\Pi_{\boldsymbol{\Phi}}(\boldsymbol{Y} - \boldsymbol{\Theta})),$$

Thus, $|\Omega^j| \leq \gamma\alpha p d$. We can show easily that $|\boldsymbol{\Omega}^{*j}| \leq 1.5p\alpha d$ w.h.p.: Since $|\Phi^j|$ can be expressed as a summation of d i.i.d. Bernoulli(p) random variables, using bounded Bernstein inequality (Vershynin, 2018, Theorem 2.8.4), with $t = pd/2$, $\Pr[||\Phi^j| - pd| \leq \frac{1}{2}pd] \geq 1 - 2\exp(-cpd)$. By union bounding over all $j \in [n]$ we get $\Pr[\sup_{j\in[n]} ||\Phi^j| - pd| \leq \frac{1}{2}pd] \geq 1 - \exp(\log n - cpd)$. Thus, if $p \geq C\frac{\log d}{d}$ then with probability at least $1 - \mathcal{O}(d^{-1})$, $|\Phi^j| \leq 1.5pd$. Now out of $1.5pd$ observed entries we have at most α sparse outliers and this implies that $|\boldsymbol{\Omega}^{*j}| \leq 1.5p\alpha d$ w.h.p.. Similar argument holds for $|\Omega_k^*| \leq 1.5p\alpha n$.

For $jk \in \Omega$, $(\boldsymbol{s}^+ - \boldsymbol{s}^*)_{jk} = (\boldsymbol{y} - \boldsymbol{\theta} - \boldsymbol{s}^*)_{jk} = (\boldsymbol{\theta}^* - \boldsymbol{\theta})_{jk}$. For $jk \in \boldsymbol{\Omega}^* \setminus \Omega$, $(\boldsymbol{s}^+ - \boldsymbol{s}^*)_{jk} = -\boldsymbol{s}^*_{jk}$. Thus, adding/subtracting $\Pi_\Phi(\boldsymbol{\Theta}^* - \boldsymbol{\Theta})$ from the second term and simplifying gives

$$\begin{aligned}
\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*) &= \Pi_\Omega(\boldsymbol{S}^+ - \boldsymbol{S}^*) + \Pi_{\boldsymbol{\Omega}^*\setminus\Omega}(\boldsymbol{S}^+ - \boldsymbol{S}^*) \\
&= \Pi_\Omega(\boldsymbol{\Theta}^* - \boldsymbol{\Theta}) + \Pi_{\boldsymbol{\Omega}^*\setminus\Omega}(\boldsymbol{\Theta}^* - \boldsymbol{\Theta}) - \Pi_{\boldsymbol{\Omega}^*\setminus\Omega}(\boldsymbol{S}^* + \boldsymbol{\Theta}^* - \boldsymbol{\Theta}) \\
&= \Pi_{\Omega\cup\boldsymbol{\Omega}^*}((\boldsymbol{\Theta}^* - \boldsymbol{\Theta}) - \Pi_{\boldsymbol{\Omega}^*\setminus\Omega}((\boldsymbol{Y} - \boldsymbol{\Theta}))
\end{aligned}$$

The second term is due to small sparse entries that could not get thresholded out. To bound these terms, we use a few ideas from proof of (Yi et al., 2016, Lemma 2).

Consider the first term. Let $\Omega_f = \Omega \cup \boldsymbol{\Omega}^*$. We have $|\Omega_f^j| \leq (\gamma+1.5)\alpha p d$ and $|\Omega_{fk}| \leq (\gamma+1.5)\alpha p n$. Thus,

$$\begin{aligned}
\|\Pi_{\Omega\cup\boldsymbol{\Omega}^*}(\boldsymbol{\Theta}^* - \boldsymbol{\Theta})\|_F^2 &\leq 2\|\Pi_{\Omega_f}(\boldsymbol{U}(\boldsymbol{B} - \boldsymbol{G}))\|_F^2 + 2\|\Pi_{\Omega_f}(\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*\boldsymbol{B}^*)\|_F^2 \\
&\leq 2\sum_{jk\in\Omega_f} (\boldsymbol{U}(\boldsymbol{B} - \boldsymbol{G}))_{jk}^2 + 2\sum_{jk\in\Omega_f} (\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*\boldsymbol{B}^*)_{jk}^2 \\
&\leq 2\sum_k \sum_{j\in\Omega_{fk}} \langle \boldsymbol{u}^j, (\boldsymbol{b}_k - \boldsymbol{g}_k)\rangle^2 + 2\sum_j \sum_{k\in\Omega_f^j} \langle (\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*)^j, \boldsymbol{b}_k^*\rangle^2 \\
&\leq 2\sum_k \sum_{j\in\Omega_{fk}} \|\boldsymbol{u}^j\|^2\|\boldsymbol{b}_k - \boldsymbol{g}_k\|^2 + 2\sum_j \sum_{k\in\Omega_f^j} \|(\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*)^j\|^2\|\boldsymbol{b}_k^*\|^2 \\
&\leq 2\mu_u^2(r/n)|\Omega_{fk}| \sum_k \|\boldsymbol{b}_k - \boldsymbol{g}_k\|^2 + 2\mu^2(r/d)\sigma_1^{*2}|\Omega_f^j| \sum_j \|(\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*)^j\|^2 \\
&\leq 2\mu_u^2(r/n)\|\boldsymbol{B} - \boldsymbol{G}\|_F^2(\gamma + 1.5)\alpha p n + 2\mu^2(r/d)\sigma_1^{*2}\|\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*\|_F^2(\gamma + 1.5)\alpha p d \\
&\leq 2\mu_u^2(\gamma + 1.5)\alpha p r\|\boldsymbol{B} - \boldsymbol{G}\|_F^2 + 2\mu^2(\gamma + 1.5)\alpha p r\sigma_1^{*2}\|\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*\|_F^2
\end{aligned}$$

Consider second term for a $jk \in \boldsymbol{\Omega}^* \setminus \Omega$, we have

$$\begin{aligned}
|\boldsymbol{y}_{jk} - \boldsymbol{\theta}_{jk}|^2 &\leq \max((\boldsymbol{y} - \boldsymbol{\theta})^{j(\gamma\alpha pd)}, (\boldsymbol{y} - \boldsymbol{\theta})_k^{(\gamma\alpha pn)})^2 \\
&\leq (\boldsymbol{y} - \boldsymbol{\theta})^{j(\gamma\alpha pd)2} + (\boldsymbol{y} - \boldsymbol{\theta})_k^{(\gamma\alpha pn)2} \\
&= (\boldsymbol{s}^* + \boldsymbol{\theta}^* - \boldsymbol{\theta})^{j(\gamma\alpha pd)2} + (\boldsymbol{s}^* + \boldsymbol{\theta}^* - \boldsymbol{\theta})_k^{(\gamma\alpha pn)2} \\
&\leq (\boldsymbol{\theta}^* - \boldsymbol{\theta})^{j(\gamma\alpha pd - 1.5\alpha pd)2} + (\boldsymbol{\theta}^* - \boldsymbol{\theta})_k^{(\gamma\alpha pn - 1.5\alpha pn)2} \\
&\leq \frac{\|(\boldsymbol{\theta}^* - \boldsymbol{\theta})^j\|^2}{(\gamma - 1.5)\alpha pd} + \frac{\|(\boldsymbol{\theta}^* - \boldsymbol{\theta})_k\|^2}{(\gamma - 1.5)\alpha pn}
\end{aligned}$$

The third row uses the fact that the j-th row (k-th column) of $\Pi_\Phi(\boldsymbol{S}^*)$ has at most $1.5\alpha pd$ ($1.5\alpha pn$) nonzero entries and follows using Lemma B.1. The fourth row uses the fact that $\boldsymbol{z}^{(m)2} \leq \|\boldsymbol{z}\|^2/m$: m-th largest entry is smaller than equal to all the m largest entries. Thus,

$$\begin{aligned}
\|\Pi_{\boldsymbol{\Omega}^*\setminus\Omega}(\boldsymbol{Y}-\boldsymbol{\Theta})\|_F^2 &= \sum_{jk\in\boldsymbol{\Omega}^*\setminus\Omega} |\boldsymbol{y}_{jk}-\boldsymbol{\theta}_{jk}|^2 \\
&\leq \sum_{jk\in\boldsymbol{\Omega}^*\setminus\Omega} \frac{\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})^j\|^2}{(\gamma-1.5)\alpha pd} + \frac{\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})_k\|^2}{(\gamma-1.5)\alpha pn} \\
&\leq \sum_j \frac{\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})^j\|^2}{(\gamma-1.5)\alpha pd} \sum_{k\in(\boldsymbol{\Omega}^*\setminus\Omega)^j} 1 + \sum_k \frac{\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})_k\|^2}{(\gamma-1.5)\alpha pn} \sum_{j\in(\boldsymbol{\Omega}^*\setminus\Omega)_k} 1 \\
&\leq \frac{\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2}{(\gamma-1.5)\alpha pd} |(\boldsymbol{\Omega}^*\setminus\Omega)^j| + \frac{\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2}{(\gamma-1.5)\alpha pn} |(\boldsymbol{\Omega}^*\setminus\Omega)_k| \\
&\leq \frac{\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2}{(\gamma-1.5)\alpha pd} 1.5\alpha pd + \frac{\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2}{(\gamma-1.5)\alpha pn} 1.5\alpha pn \\
&\leq 3\frac{\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2}{(\gamma-1.5)}
\end{aligned}$$

Thus,

$$\|\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*)\|_F^2 \leq \frac{3}{(\gamma-1.5)}\|\Pi_\Phi(\boldsymbol{\Theta}^*-\boldsymbol{\Theta})\|_F^2 + 2\mu_u^2(\gamma+1.5)\alpha pr\|\boldsymbol{B}-\boldsymbol{G}\|_F^2 + 2\mu^2(\gamma+1.5)\alpha pr\|\mathcal{P}_{\boldsymbol{U},\perp}\boldsymbol{U}^*\|_F^2$$

From Lemma A.4, item 5, with high probability

$$\|\Pi_\Phi(\boldsymbol{\Theta}-\boldsymbol{\Theta}^*)\|_F \leq 2.6\sqrt{p}\delta_t\sigma_1^*.$$

Substituting this bound, we get

$$\|\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*)\|_F^2 \leq 3\frac{2.6^2 p\delta_t^2\sigma_1^{*2}}{(\gamma-1.5)} + 2(\mu_u^2(\gamma+1.5)+\mu^2(\gamma+1.5))\alpha pr\delta_t^2\sigma_1^{*2} \tag{22}$$

We now bound the term $\|(\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*))\boldsymbol{B}^\top\|_F$. Each j-th row of $(\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*))\boldsymbol{B}^\top$ can be written as

$$((\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*))\boldsymbol{B}^\top)^j = \sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}\boldsymbol{b}_k^\top.$$

Applying Cauchy–Schwarz we get

$$\begin{aligned}
\|((\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*))\boldsymbol{B}^\top)^j\|^2 &= (\sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}\boldsymbol{b}_k^\top)^2 && (23)\\
&\leq (\sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}^2)(\sum_{k\in\Omega_f^j} \|\boldsymbol{b}_k\|^2) && (24)\\
&\leq (\sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}^2)|\Omega_f^j|\mu^2(r/d) && (25)\\
&\leq (\sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}^2)(\gamma+1.5)\alpha pd\mu^2(r/d) && (26)\\
&= (\sum_{k\in\Omega_f^j} (\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}^2)(\gamma+1.5)\alpha pr\mu^2\sigma_1^{*2} && (27)
\end{aligned}$$

Now $\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F^2 = \sum_j \|((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)^j\|^2$ implies

$$\|(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\|_F^2 = \sum_j \|((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)^j\|^2 \tag{28}$$

$$\leq \sum_j \sum_{k\in\Omega_f^j} (\boldsymbol{s}^+ - \boldsymbol{s}^*)_{jk}^2(\gamma+1.5)\alpha pr\mu^2 \tag{29}$$

$$= (\gamma+1.5)\alpha pr\mu^2\|\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)\|_F^2 \tag{30}$$

$$\leq \gamma\alpha pr\mu^2\sigma_1^{*2}\left(3\frac{2.6^2 p\delta_t^2\sigma_1^{*2}}{(\gamma-1.5)} + 2(\mu_u^2(\gamma+1.5) + \mu^2(\gamma+1.5))\alpha pr\delta_t^2\sigma_1^{*2}\right) \tag{31}$$

where the last row used (22).

In this bound, γ is an algorithm parameter and α is the outlier fraction per row or column that needs to be upper bounded for our guarantee to hold. Our algebra lemma needs this bound to be smaller than $cp^2\sigma_r^{*4}\delta_t^2$. Thus, we need

$$\gamma = C > 1.5 \text{ and } \alpha \leq \frac{c}{\mu_u^2\kappa^4 r} = \frac{c}{\kappa^8\mu^2 r}$$

using $\mu_u = C\kappa^2\mu$.

□

B.4 Proof of Lemma A.4

Proof. Most of this proof is similar to that given in (Abbasi & Vaswani, 2025, Lemma 4.2). The main change here is that, in this algorithm, we need to impose incoherence of $\boldsymbol{B}$ which we do via projection step. This is needed because $\boldsymbol{b}_k$ contains sparse component term $(\boldsymbol{s}_k - \boldsymbol{s}_k^*)$ which makes the matrix Bernstein inequality bound inapplicable.

1. Since $\boldsymbol{B} = \Pi_{\mu,col}(\tilde{\boldsymbol{B}})$, thus by this construction, we have $\|\boldsymbol{b}_k\| \leq \mu\sqrt{\frac{r}{d}}\sigma_1^*$

2. Let $\boldsymbol{G} = \boldsymbol{U}^\top\boldsymbol{\Theta}^*$, we can write $\boldsymbol{b}_k - \boldsymbol{g}_k = (\boldsymbol{U}^\top\Pi_{\Phi_k}\boldsymbol{U})^{-1}\big(\boldsymbol{U}^\top\Pi_{\Phi_k}(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)\boldsymbol{U}^*\boldsymbol{b}_k^* - \boldsymbol{U}^\top\Pi_{\Phi_k}(\boldsymbol{s}_k - \boldsymbol{s}_k^*)\big)$. Let $\boldsymbol{F} := \text{blockdiag}(\boldsymbol{F}_k, k \in [q])$ and $\boldsymbol{D} := [\boldsymbol{d}_k]_{k\in[q]}$, where $\boldsymbol{F}_k := \boldsymbol{U}^\top\Pi_{\Phi_k}\boldsymbol{U}$, $\boldsymbol{d}_k := \boldsymbol{U}^\top\Pi_{\Phi_k}[\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top]\boldsymbol{U}^*\boldsymbol{b}_k^*$. This then implies $\|\tilde{\boldsymbol{B}} - \boldsymbol{G}\|_F = \|\boldsymbol{F}^{-1}vec(\boldsymbol{D} - \boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*))\| \leq \|\boldsymbol{F}^{-1}\| \cdot (\sqrt{r}\|\boldsymbol{D}\| + \|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F)$. Now

$$\|\boldsymbol{B} - \boldsymbol{G}\|_F = \|\Pi_{\mu,col}(\tilde{\boldsymbol{B}}) - \boldsymbol{U}^\top\boldsymbol{\Theta}^*\|_F \tag{32}$$

$$= \|\Pi_{\mu,col}(\tilde{\boldsymbol{B}}) - \Pi_{\mu,col}(\boldsymbol{U}^\top\boldsymbol{\Theta}^*)\|_F \tag{33}$$

$$\text{Using non-expansive property on convex set} \tag{34}$$

$$\leq \|\tilde{\boldsymbol{B}} - \boldsymbol{U}^\top\boldsymbol{\Theta}^*\|_F \tag{35}$$

$$\leq \|\boldsymbol{F}^{-1}\| \cdot (\sqrt{r}\|\boldsymbol{D}\| + \|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F) \tag{36}$$

$$\leq \frac{1}{(1-\epsilon)p}(\sqrt{r}\epsilon p\delta_t\sigma_1^* + \frac{c_1}{2}p\delta_t\sigma_1^*) \tag{37}$$

$$\leq c_1\delta_t\sigma_1^* \tag{38}$$

In above we have used (Abbasi & Vaswani, 2025, Lemma 4.2) for bound on $\|\boldsymbol{F}^{-1}\|$, and $\|\boldsymbol{D}\|$ which holds w.p. at least $1 - \exp(\log d - c\frac{c_1^2 pn}{r\mu_u^2})$.

In the above we used the assumed bound $\|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S} - \boldsymbol{S}^*)\|_F \leq \frac{c_1}{2}\, p\, \delta_t\sigma_1^*$.

3. As shown in Abbasi & Vaswani (2025), this then implies $\|\boldsymbol{B}\| \leq \|\boldsymbol{B} - \boldsymbol{G}\| + \|\boldsymbol{G}\| \leq c_1\delta_t\sigma_1^* + \sigma_1^* \leq 1.1\sigma_1^*$ using $\delta_t \leq \delta_0 \leq 0.01/c_1\kappa$.

4. As shown in Abbasi & Vaswani (2025), we have $\sigma_{\min}(\boldsymbol{B}) \geq \sigma_r^*(\boldsymbol{G}) - \|\boldsymbol{G}-\boldsymbol{B}\|_F \geq \sqrt{1-\delta_t^2}\sigma_r^* - c_1\delta_t\sigma_1^* \geq 0.9\sigma_r^*$ using $\delta_t \leq \delta_0 \leq 0.01/c_1\kappa$.
5. From (Abbasi & Vaswani, 2025, Lemma 4.3), since $\|\boldsymbol{B}-\boldsymbol{G}\|_F \leq c_1\delta_t\sigma_1^* \leq \delta_t\sigma_1^*$, we have $\|\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*\|_F \leq 2\delta_t\sigma_1^*$. This, from (Yi et al., 2016, Lemma 9), implies that if $p \geq C\frac{\mu_u r \log d}{d}$ then w.h.p.
$$\|\Pi_\Phi(\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*)\|_F \leq 2.6\sqrt{p}\,\delta_t\sigma_1^*.$$
6. Writing $(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top = p(\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top + \mathrm{Err}$, with $\|\mathrm{Err}\|_F \leq \epsilon p\delta_t\sigma_r^{*2}$ from (Abbasi & Vaswani, 2025, Lemma 4.5), and using items 3 and 5,
$\|(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top\| \leq p\,\|\boldsymbol{U}\boldsymbol{B}-\boldsymbol{U}^*\boldsymbol{B}^*\|_F\,\|\boldsymbol{B}\| + \epsilon p\delta_t\sigma_r^{*2} \leq 2.2\,p\,\delta_t\sigma_1^{*2} + \epsilon p\delta_t\sigma_r^{*2} \leq 2.6\,p\,\delta_t\sigma_1^{*2}$.

□

B.5 Proof of Lemma A.7

Proof. The k-th column of $\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S}-\boldsymbol{S}^*)$ equals $\sum_{j:jk\in\Omega_f}(\boldsymbol{s}-\boldsymbol{s}^*)_{jk}\,\boldsymbol{u}^j$, where Ω_f denotes the support of $\Pi_\Phi(\boldsymbol{S}-\boldsymbol{S}^*)$, with $|\Omega_{f,k}| \leq (\gamma+1.5)\alpha pn$ w.h.p. (proof of Lemma A.3). Using Cauchy–Schwarz and the assumed bounds,

$$\|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S}-\boldsymbol{S}^*)\|_F^2 = \sum_k \|\textstyle\sum_{j:jk\in\Omega_f}(\boldsymbol{s}-\boldsymbol{s}^*)_{jk}\,\boldsymbol{u}^j\|^2 \leq \sum_k |\Omega_{f,k}| \sum_{j:jk\in\Omega_f} |(\boldsymbol{s}-\boldsymbol{s}^*)_{jk}|^2\,\|\boldsymbol{u}^j\|^2$$
$$\leq (\gamma+1.5)\alpha pn\,\mu_u^2\tfrac{r}{n}\,\|\Pi_\Phi(\boldsymbol{S}-\boldsymbol{S}^*)\|_F^2 \leq (\gamma+1.5)\alpha pr\,\mu_u^2\,c_2^2\,p\,\delta_{t-1}^2\sigma_1^{*2}.$$

Since $\delta_{t-1} \leq 5\delta_t$, with $\alpha \leq \frac{c\,c_1^2}{c_2^2\,\mu_u^2 r}$ for $c \leq 0.01/(\gamma+1.5)$, we get
$$\|\boldsymbol{U}^\top\Pi_\Phi(\boldsymbol{S}-\boldsymbol{S}^*)\|_F \leq \frac{c_1}{2}\,p\,\delta_t\sigma_1^*.$$

□

B.6 Proof of Lemma A.5

Proof. $\boldsymbol{S}^+ = \mathcal{T}_{\gamma\alpha p}(\Pi_\Phi(\boldsymbol{S}^*+\boldsymbol{\Theta}^*-\boldsymbol{\Theta}))$, $\boldsymbol{S}_{dif} = \Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*)$ with support on $\Omega\cup\boldsymbol{\Omega}^*$

1) Case 1, for $jk\in\Omega$ we have

$$|(\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}| = |(\boldsymbol{\theta}^*-\boldsymbol{\theta})_{jk}| \tag{39}$$
$$\leq |\langle\boldsymbol{u}^{*j},\boldsymbol{b}_k^*\rangle| + |\langle\boldsymbol{u}^j,\boldsymbol{b}_k\rangle| \tag{40}$$
$$\leq \|\boldsymbol{u}^{*j}\|\|\boldsymbol{b}_k^*\| + \|\boldsymbol{u}^j\|\|b_k\| \tag{41}$$
$$\leq 2\max(\|\boldsymbol{u}^{*j}\|,\|\boldsymbol{u}^j\|)\mu\sqrt{r/d}\sigma_1^* \tag{42}$$

2) Case 2, for $jk\in\boldsymbol{\Omega}^*\setminus\Omega$ we have

$$|(\boldsymbol{s}^+-\boldsymbol{s}^*)_{jk}| = |\boldsymbol{s}_{jk}^*| \leq |y_{jk}-x_{jk}| + |(\boldsymbol{\theta}^*-\boldsymbol{\theta})_{jk}| \tag{43}$$
$$\leq 2\max\left(\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})^j\|_\infty, \|(\boldsymbol{\theta}^*-\boldsymbol{\theta})_k\|_\infty\right) \tag{44}$$
$$\leq 4\mu_u\sqrt{r/n}\,\mu\sqrt{r/d}\sigma_1^* \tag{45}$$

The second line holds because the entry was zeroed by $\mathcal{T}_{\gamma\alpha p}$, i.e., $|y_{jk}-x_{jk}|$ is below the larger of the row and column thresholds; by Lemma B.1 (since $\boldsymbol{s}^{*j}$, resp. $\boldsymbol{s}_k^*$, has at most $1.5\alpha pd$, resp. $1.5\alpha pn$, nonzeros and $\gamma > 1.5$) the row threshold is at most $\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})^j\|_\infty$ and the column threshold at most $\|(\boldsymbol{\theta}^*-\boldsymbol{\theta})_k\|_\infty$. The third line uses $|(\boldsymbol{\theta}^*-\boldsymbol{\theta})_{ik}| \leq \|\boldsymbol{u}^{*i}\|\|\boldsymbol{b}_k^*\| + \|\boldsymbol{u}^i\|\|\boldsymbol{b}_k\| \leq 2\mu_u\sqrt{r/n}\,\mu\sqrt{r/d}\sigma_1^*$ for every (i,k). Now combining both cases we get

$$\|\boldsymbol{e}_j^\top\Pi_\Phi(\boldsymbol{S}^+-\boldsymbol{S}^*)\|^2 \leq \sum_{k\in\Omega^j} 4\mu^2\frac{r}{d}\sigma_1^{*2}\max(\|\boldsymbol{u}^{*j}\|,\|\boldsymbol{u}^j\|)^2 + \sum_{k\in\boldsymbol{\Omega}^{*j}} 16\mu^2\frac{r}{d}\sigma_1^{*2}\mu_u^2\frac{r}{n} \tag{46}$$
$$\leq p\gamma\alpha d\,4\mu^2\frac{r}{d}\sigma_1^{*2}\max(\|\boldsymbol{u}^{*j}\|,\|\boldsymbol{u}^j\|)^2 + 1.5p\alpha d\,16\mu^2\frac{r}{d}\sigma_1^{*2}\mu_u^2\frac{r}{n} \tag{47}$$
$$\leq 4p\alpha\mu^2 r\sigma_1^{*2}\Big(\gamma\max(\|\boldsymbol{u}^{*j}\|,\|\boldsymbol{u}^j\|)^2 + 6\mu_u^2\frac{r}{n}\Big) \tag{48}$$

Above we have used (Yi et al., 2016, Lemma 10) if $p \geq \frac{56}{3}\frac{\log d}{\alpha d}$ then $|\mathbf{\Omega}^{*j}| \leq \frac{3}{2}\alpha p d$ w.p. at least $1-6d^{-1}$, and the fact that $|\Omega^j| \leq p\gamma\alpha d$. Therefore

$$\|\boldsymbol{e}_j^\top((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)\| = \Big\| \sum_{k\in\Omega_f^j} (\boldsymbol{s}^+ - \boldsymbol{s}^*)_{jk}\boldsymbol{b}_k^\top \Big\| \leq \|\boldsymbol{e}_j^\top \Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)\| \Big(\sum_{k\in\Omega_f^j} \|\boldsymbol{b}_k\|^2 \Big)^{1/2} \tag{49}$$

$$\leq \|\boldsymbol{e}_j^\top \Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)\| \sqrt{(\gamma+1.5)\alpha p r}\, \mu\sigma_1^* \tag{50}$$

$$\leq 2(\gamma+1.5)\, p\, \alpha r \mu^2 \sigma_1^{*2} \Big(\sqrt{\gamma}\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \sqrt{6}\,\mu_u\sqrt{r/n} \Big) \tag{51}$$

$$= 2(\gamma+1.5)\, p\, \alpha r \mu^2 \kappa^2 \Big(\sqrt{\gamma}\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \sqrt{6}\,\mu_u\sqrt{r/n} \Big) \sigma_r^{*2} \tag{52}$$

Here $\Omega_f = \Omega \cup \mathbf{\Omega}^*$ is the support of $\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*)$, $|\Omega_f^j| \leq (\gamma+1.5)\alpha p d$, and we used Cauchy–Schwarz with $\|\boldsymbol{b}_k\| \leq \mu\sqrt{r/d}\sigma_1^*$.

For $\|\boldsymbol{e}_j^\top((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)\| \leq c_2 p\big(\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \mu_u\sqrt{r/n}\big)\sigma_r^{*2}$, we need $\alpha \leq \frac{c_2}{2\sqrt{6}\,(\gamma+1.5)\sqrt{\gamma}\,\mu^2 r\kappa^2}$. □

B.7 Proof of Lemma A.6

Proof. Since $\hat{\boldsymbol{U}}^+ = \boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+)$, and $\hat{\boldsymbol{U}}^+ \overset{QR}{=} \boldsymbol{U}^+ R^+$ we have

$$\hat{\boldsymbol{u}}^{+j} = \boldsymbol{e}_j^T(\boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}, \boldsymbol{S}^+)) \tag{53}$$

$$= \boldsymbol{e}_j^T(\boldsymbol{U} - \eta\big(\big(\Pi_\Phi((\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*) + (\boldsymbol{S}^+ - \boldsymbol{S}^*))\big)\boldsymbol{B}^\top\big) \pm \eta p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top) \tag{54}$$

$$= \boldsymbol{e}_j^T\boldsymbol{U}(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top) + \eta p\boldsymbol{e}_j^T\boldsymbol{U}^*\boldsymbol{B}^*\boldsymbol{B}^\top - \eta\boldsymbol{e}_j^\top(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top \tag{55}$$

$$- \eta\boldsymbol{e}_j^T((\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*)\boldsymbol{B}^\top) \tag{56}$$

$$\|\hat{\boldsymbol{u}}^{+j}\| \leq \|\boldsymbol{u}^j\|(1 - \eta p\sigma_{min}^2(\boldsymbol{B})) + \eta p\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2} + \eta\|\boldsymbol{e}_j^\top(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top\| + \eta\|\boldsymbol{e}_j^\top \mathrm{Err}\| \tag{57}$$

Using Lemma A.5 we get w.p. at least $1 - \mathcal{O}(d^{-1})$ $\|\boldsymbol{e}_j^\top((\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)\| \leq c_2 p\big(\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \mu_u\sqrt{r/n}\big)\sigma_r^{*2}$. Also using (Abbasi & Vaswani, 2025, Lemma 4.6 item 1) assuming $\|\boldsymbol{b}_k\| \leq 1.1\sigma_1^*\mu\sqrt{r/d}$, then w.p. at least $1 - \exp(\log d - c\epsilon^2 pn/\kappa^4\mu^2 r^2)$

$$\|\boldsymbol{e}_j^\top \mathrm{Err}\| \leq \epsilon p \max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|)\sigma_r^{*2}$$

Combining we get

$$\|\hat{\boldsymbol{u}}^{+j}\| \leq (1 - 0.8\eta p\sigma_r^{*2})\|\boldsymbol{u}^j\| + \eta p\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2} + \eta p c_2\big(\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \mu_u\sqrt{r/n}\big)\sigma_r^{*2} + \eta\epsilon p\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|)\sigma_r^{*2} \tag{58}$$

$$\leq (1 - (0.8 - c_2 - \epsilon)\eta p\sigma_r^{*2})\max(\|\boldsymbol{u}^{*j}\|, \|\boldsymbol{u}^j\|) + \eta p\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2} + c_2\eta p\sigma_r^{*2}\mu_u\sqrt{r/n} \tag{59}$$

Now $\hat{\boldsymbol{U}}^+ \overset{\text{QR}}{=} \boldsymbol{U}\boldsymbol{R}^+$ implies $\|\boldsymbol{u}^j\| \leq \|\hat{u}^{+j}\|\|(\boldsymbol{R}^+)^{-1}\|$

$$\|(R^+)^{-1}\| = \frac{1}{\sigma_{min}(\boldsymbol{U} - \eta(\Pi_\Phi(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{U}^*\boldsymbol{B}^*))\boldsymbol{B}^\top - \eta(\Pi_\Phi(\boldsymbol{S}^+ - \boldsymbol{S}^*))\boldsymbol{B}^\top)} \tag{60}$$

$$\leq \big(1 - \eta 2.6p\delta_t\sigma_1^{*2} - \eta p\delta_t\sigma_1^{*2}\big)^{-1} \tag{61}$$

$$= \big(1 - 3.6\delta_t\eta p\sigma_1^{*2}\big)^{-1} \tag{62}$$

$$= \big(1 - 0.1\eta p\sigma_r^{*2}\big)^{-1} \tag{63}$$

$$\leq \big(1 + 0.2\eta p\sigma_r^{*2}\big) \tag{64}$$

Above we have used $\delta_t \leq \delta_0 \leq \frac{0.1}{3.6\kappa^2}$. Combining everything

$$\|\boldsymbol{u}^{+j}\| \leq (1+0.2\eta p\sigma_r^{*2})(1-(0.8-c_2-\epsilon)\eta p\sigma_r^{*2})\max(\|\boldsymbol{u}^j\|,\|\boldsymbol{u}^{*j}\|)+(1+0.2\eta p\sigma_r^{*2})\eta p\|\boldsymbol{u}^{*j}\|\sigma_{\max}^{*2}+(1+0.2\eta p\sigma_r^{*2})c_2\eta \tag{65}$$

$$\leq (1-(0.6-c_2-\epsilon)\eta p\sigma_r^{*2})\max(\|\boldsymbol{u}^j\|,\|\boldsymbol{u}^{*j}\|)+(1+0.1)0.5\|\boldsymbol{u}^{*j}\|+1.1\,c_2\frac{0.5}{\kappa^2}\mu_u\sqrt{r/n} \tag{66}$$

$$\leq (1-(0.6-c_2-\epsilon)\eta p\sigma_r^{*2})\max(\|\boldsymbol{u}^j\|,\|\boldsymbol{u}^{*j}\|)+0.6\|\boldsymbol{u}^{*j}\|+\frac{0.03}{\kappa^2}\mu_u\sqrt{r/n} \tag{67}$$

$$\leq (1-\frac{0.25}{\kappa^2})\max(\|\boldsymbol{u}^j\|,\|\boldsymbol{u}^{*j}\|)+0.6\|\boldsymbol{u}^{*j}\|+\frac{0.03}{\kappa^2}\mu_u\sqrt{r/n} \tag{68}$$

$$\leq (1-\frac{0.25}{\kappa^2})\|\boldsymbol{u}^j\|+2\|\boldsymbol{u}^{*j}\|+\frac{0.05}{\kappa^2}\mu_u\sqrt{r/n} \tag{69}$$

Using $c_2 = 0.1/2$, $\epsilon = 0.1/2$, and $\eta = \frac{0.5}{p\sigma_{\max}^{*2}}$. □

Lemma B.1. *Let v_1, v_2 be two d dimensional vectors. Let v_2 has m_1 non-zero entries. Then for $m_2 > m_1$*

$$(v_1+v_2)^{(m_2)} \leq v_1^{(m_2-m_1)}$$

Proof. Case 1: In the worst case largest m_1 out of m_2 entries can come from v_2. This then implies the remaining $m_2 - m_1$ largest entries are of v_1. So in worst case $(v_1+v_2)^{(m_2)} \leq v_1^{(m_2-m_1)}$ see Figure 2 for illustration.

Case 2: v_2 doesn't contribute to largest entries then $(v_1+v_2)^{(m_2)} = v_1^{(m_2)} \leq v_1^{(m_2-m_1)}$.

Case 3: v_2 contribute to $m_3 < m_1$ largest entries then $(v_1+v_2)^{(m_2)} \leq v_1^{(m_2-m_3)} \leq v_1^{(m_2-m_1)}$.

Case 4: v_1+v_2 both contribute to m_4 largest entries. Note in this case since v_2 can perturb at most m_1 entries therefore $m_4 \leq m_1$. Hence, $(v_1+v_2)^{(m_2)} \leq v_1^{(m_2-m_4)} \leq v_1^{(m_2-m_1)}$.

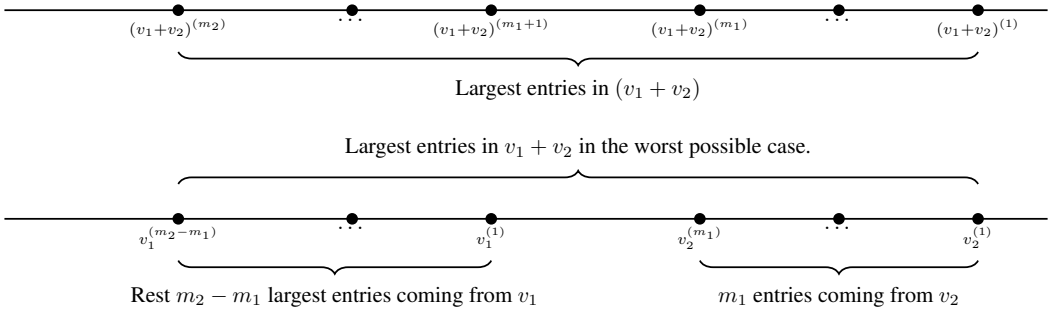


Figure 2: Illustration

□

C Details for Federated Algorithm Initialization and for KLL sketching

C.1 Initialization

We need to compute the top r singular vectors of $p^{-1}(\boldsymbol{Y}-\boldsymbol{S}_0) = \boldsymbol{M}$ in a federated fashion. But each node has only a few columns of this matrix (call them $\boldsymbol{M}_{(\ell)}$). So we use the federated power method (Hardt & Price, 2014, Theorem 1.1) to estimate $\boldsymbol{U}_0$. For each power method iteration t, each node sends $\boldsymbol{M}_{(\ell)}\boldsymbol{M}_{(\ell)}^\top\boldsymbol{G}_t$ to the server, where $\boldsymbol{G}$ is an $n \times r$ matrix initialized at $t = 0$ with

a random Gaussian matrix. The central server computes $\boldsymbol{G}_{t+1} = \mathrm{QR}\Big(\sum_{\ell=1}^{L} \boldsymbol{M}_{(\ell)}\boldsymbol{M}_{(\ell)}^{\top}\boldsymbol{G}_t\Big)$. It is well known Hardt & Price (2014) that $\boldsymbol{G}$ converges to $\boldsymbol{U}_0$ with error δ_0 in $\mathcal{O}(\log(dr)/\delta_0)$ iterations.

Power method in Initialization. From (Hardt & Price, 2014, Theorem 1.1) using $T_{power} \geq C\frac{\sigma_r^*+b}{\sigma_r^*-2b}\log\Big(d\frac{\sigma_r^*}{b}\Big)$ we get w.p. at least $1-\mathcal{O}(d^{-1})$ $\boldsymbol{SD}_F(\boldsymbol{U}_0,\boldsymbol{U}^*) \leq \delta_0$, here $b = \|\Pi_{\mathcal{U}}(\boldsymbol{Y} - \boldsymbol{S}_0) - \Pi_{\mathcal{U}}(\boldsymbol{\Theta}^*)\| = \|\Pi_{\mathcal{U}}(\boldsymbol{Y} - \boldsymbol{S}_0) - \boldsymbol{\Theta}^*\| \leq \|\boldsymbol{Y} - \boldsymbol{S}_0 - \boldsymbol{\Theta}^*\| \leq 7\alpha\mu^2 r\sigma_1^*$.

C.2 Background on Quantile Sketching

The problem of estimating quantiles in streaming data, where elements $\boldsymbol{y} = \{y_1, y_2, \ldots, y_d\}$ arrive sequentially and cannot all be stored in memory, is fundamental in large-scale data analytics. The goal is to estimate, for any query value y, its $\mathrm{order}(y)$

$$\mathrm{order}(y) = |\{\, y_k \in \boldsymbol{y} \mid y \leq y_k \,\}|,$$

that is, the number of items in the stream larger than or equal to y. An *ϵ-approximate quantile sketch* provides an estimate $\mathrm{order}(\widehat{y})$ such that

$$|\,\mathrm{order}(\widehat{y}) - \mathrm{order}(y)\,| \leq \epsilon d,$$

with probability at least $1-\delta$.

A **sketch** is thus a compressed summary of the stream that retains sufficient information to answer quantile queries approximately. Its *size* (or space complexity) refers to the number of items or equivalent memory units the sketch must maintain to guarantee the (ϵ, δ) accuracy bound. The theoretical minimum possible sketch size is $\mathcal{O}(1/\epsilon)$. The best known deterministic comparison based algorithm is called GK Greenwald & Khanna (2001), and it consumes $\mathcal{O}(\epsilon^{-1}\log \epsilon d)$ space. This is not tight because it grows with d. KLL Karnin et al. (2016) is a randomized algorithm that solves the problem with probability $1-\delta$ using only $\mathcal{O}(\epsilon^{-1}\log\delta)$ space.

Compactors and Hierarchical Levels. The KLL sketch is a way to store a large sequence of numbers in a very small space while still preserving their order information. Instead of keeping all values, it keeps a carefully chosen subset that summarizes the full data.

The sketch is organized into several levels indexed by $h \in [H]$. Each level has a small buffer called a compactor, denoted by $C[h]$. Each compactor can store at most K_h values. The lowest level receives the data first, and higher levels store increasingly compressed summaries.

When new values arrive, they are first added to the lowest level. As long as the buffer $C[h]$ does not exceed its capacity K_h, nothing happens. When a compactor becomes full, it performs a compression step.

In this step, the values in the buffer are sorted. Sorting is important because it preserves the order of the data. After sorting, the compactor randomly chooses whether to keep the values at even positions or odd positions, and discards the rest. This removes about half of the values. The random choice ensures that no part of the data is favored, so the order information is preserved on average.

The values that remain after this step are passed to the next level. Since half of the values were discarded, each remaining value now represents two original values. For this reason, every value stored at level h is assigned a weight $w_h = 2^{h-1}$. This weight records how many original values the stored element summarizes. Each higher level repeats the same process. Because each level reduces the number of values by about half, the number of levels needed is small. In fact, the total number of levels satisfies $H \leq \lceil \log(\frac{d}{K}) \rceil$. Here K is the base capacity of the lowest level and d is the total number of data points.

Lower levels contain more values and capture fine details of the data. Higher levels contain fewer values but represent larger portions of the data using larger weights. The capacities of higher levels are chosen as $K_h \geq Kc^{H-h}$, $c \in (\frac{1}{2}, 1)$. By setting $K = \frac{1}{\epsilon}\sqrt{\log(\frac{1}{\delta})}$, the sketch guarantees that the rank error is at most ϵd with probability at least $1-\delta$. At the same time, the total memory used by the sketch is only $\mathcal{O}(KH)$ see Algorithm 4.

Mergeable property: A useful property of the KLL sketch is that sketches over two datasets are mergeable. Let KLL_1 and KLL_2 be sketches over datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. They can be

merged as $\mathrm{KLL} = \mathrm{MERGE}\{\mathrm{KLL}_1, \mathrm{KLL}_2\}$ using Algorithm 5. The resulting sketch KLL is a sketch of $\mathcal{D}_1 \cup \mathcal{D}_2$ and again satisfies the same (ϵ, δ) guarantee over $\mathcal{D}_1 \cup \mathcal{D}_2$.

We give the KLL Lemma C.1 below

Lemma C.1 (Corollary 1 Karnin et al. (2016)). *Let $\boldsymbol{y} = \{y_1, y_2, \ldots, y_d\}$ be a collection of data points, and let* KLL *denote the sketch produced by Algorithm 4 applied to $\boldsymbol{y}$. With $K = \frac{1}{\epsilon}\sqrt{\log(\frac{1}{\delta})}$, and $H = 1 + \left\lceil \log_{3/2}\left(\frac{d}{K}\right)\right\rceil$ the resulting sketch* KLL *which is of size $\mathcal{O}(KH)$ is mergeable and satisfies the following guarantee. For any value $y \in \boldsymbol{y}$, let* $\mathrm{order}(y)$ *denote its true order in $\boldsymbol{y}$, and let* $\mathrm{order}(\widehat{y})$ *denote the order estimated from the sketch then*

$$\Pr\{|\mathrm{order}(\widehat{y}) - \mathrm{order}(y)| \le \epsilon d\} \ge 1 - \delta.$$

In summary, the KLL sketch works by repeatedly sorting and compressing the data across levels. Each level keeps a smaller but weighted summary of the data below it. Together, these compactors form a compressed representation that preserves order information while using very little space and supporting efficient quantile queries.

Federated Extension. The idea of streaming data streams can be used for federated setting with L nodes, where each node ℓ observes a local dataset $\boldsymbol{y}_\ell = \{y_1, y_2, \ldots, y_{\tilde{d}}\}$ with $\tilde{d} = d/L$. Each node independently constructs its local KLL sketch

$$\mathrm{KLL}_\ell = \{C_\ell[h]\}_{h=1}^{H},$$

using the same compactor structure and parameters. At the central server, using the mergeable property of KLL sketch local sketches are merged levelwise Algorithm 5:

$$C_{(g)}[h] = \bigcup_{\ell=1}^{L} C_\ell[h], \quad h = 1, \ldots, H.$$

If a merged compactor exceeds its capacity k_h, it is uniformly downsampled to k_h items, and the excess items are promoted to level $h+1$. This process produces a global sketch $\mathrm{KLL}_{\mathrm{global}} = \{C_{(g)}[h]\}_{h=1}^{H}$ that is statistically equivalent to a single centralized sketch built on the union of all local datasets.

We restate the lemma from the main text proof outline.

Lemma C.2. *Consider a vector $\boldsymbol{y} = [y_1, ..., y_d]$ partitioned into L nodes each having d_ℓ scalars. Each node computes a KLL sketch (KLL_ℓ) of its largest αd elements with base capacity K and number of compactors H. Let $\widehat{\boldsymbol{y}^{(\alpha d)}}$ denote the approximation of the αd-th largest element of $\boldsymbol{y}$ obtained by querying the merged sketch $\mathrm{KLL}_{global} = \mathrm{KLLmerge}\{\mathrm{KLL}_\ell\}_{\ell=1}^{L}$. Pick a $0 < \delta < 1$. If $K = CL\sqrt{\log\left(\frac{L}{\delta}\right)}$ and $H = 1 + \left\lceil \log\left(\frac{\alpha d}{K}\right)\right\rceil$, then $\Pr\{|\mathrm{order}(\widehat{\boldsymbol{y}^{(\alpha d)}}) - \alpha d| \le 0.1\alpha d\} \ge 1 - \delta$. Communication cost per node is $\mathcal{O}(KH)$. Computation cost per node is $\mathcal{O}(\alpha d \log K)$. Merging cost will be $\mathcal{O}(LK)$. Here $\mathrm{order}(z)$ is the place of z when entries are arranged in decreasing order.*

Effect on thresholding: Proof of Lemma C.2. Let $\boldsymbol{y} = \{y_k\}_{k\in[d]}$ be the set of all data points. Define a modified Threshold operator $\mathcal{T}_{\gamma\alpha p, \mathrm{Threshold}}$

$$\mathcal{T}_{\gamma\alpha p, \mathrm{Threshold}}(\boldsymbol{y}) = \begin{cases} y_k, & \text{if } |y_k| \ge |\mathrm{Threshold}|, \\ 0, & \text{otherwise} \end{cases}.$$

Here $\mathrm{Threshold}$ is the approximation of $\gamma\alpha pd$-th largest element in $\boldsymbol{y}$. In centralized setting $\mathrm{Threshold} = \boldsymbol{y}^{(\gamma\alpha pd)}$ which is the true $\gamma\alpha pd$-th largest item in $\boldsymbol{y}$ i.e., $\mathrm{order}(\boldsymbol{y}^{(\gamma\alpha pd)}) = \gamma\alpha pd$. Also let $\tilde{\boldsymbol{y}} = \mathcal{T}_{\gamma\alpha p, \mathrm{Threshold}}(\boldsymbol{y})$.

Now in federated setting $\boldsymbol{y}$ is divided into L nodes $\boldsymbol{y}_\ell$, $\ell \in [L]$ each with d_ℓ data points. We estimate KLL sketches over largest $\gamma\alpha pd$ elements at each node represented as KLL_ℓ, $\ell \in [L]$ and merge them $\mathrm{KLL}_{global} = \mathrm{MERGE}\{\mathrm{KLL}_\ell\}_{\ell=1}^{L}$ to get a sketch over largest $\gamma\alpha pdL$ elements. Approximating $\gamma\alpha pd$-th largest element from KLL_{global} we get $\mathrm{Threshold}$.

By Lemma C.1, $|\mathrm{order}(\mathrm{Threshold}) - \gamma\alpha pd| \le \epsilon\gamma\alpha pdL$. Setting $\epsilon = \frac{0.1}{L}$, w.h.p.

$$0.9\,\gamma\alpha pd \le \mathrm{order}(\mathrm{Threshold}) \le 1.1\,\gamma\alpha pd. \tag{70}$$

We use both sides of (70) in the proof of Lemma A.3, as follows.

Count side (upper bound). The number of nonzero entries of $\tilde{\boldsymbol{y}}$ equals $\text{order}(\text{Threshold}) \leq 1.1\gamma\alpha pd$. Thus all support-based bounds in the proof of Lemma A.3 (and in Lemma A.7) hold with $(\gamma + 1.5)$ replaced by $(1.1\gamma + 1.5)$.

Magnitude side (lower bound). An entry y_k that is *not* kept satisfies $|y_k| < |\text{Threshold}| \leq |\boldsymbol{y}^{(0.9\gamma\alpha pd)}|$, i.e., it is bounded by the $(0.9\gamma\alpha pd)$-th largest entry rather than the $(\gamma\alpha pd)$-th. Since at most $1.5\alpha pd$ of the entries of $\boldsymbol{y}$ are corrupted w.h.p., using Lemma B.1 with $m_1 = 1.5\alpha pd$ and $m_2 = 0.9\gamma\alpha pd$, $|\boldsymbol{y}^{(0.9\gamma\alpha pd)}|$ is bounded by the $(0.9\gamma - 1.5)\alpha pd$-th largest *clean* entry, and, for any vector $\boldsymbol{z}$, its m-th largest magnitude entry satisfies $(z^{(m)})^2 \leq \|\boldsymbol{z}\|^2/m$ with $m = (0.9\gamma - 1.5)\alpha pd$. Thus the term $\frac{4}{\gamma - 1.5}$-type bounds in the proof of Lemma A.3 hold with $(\gamma - 1.5)$ replaced by $(0.9\gamma - 1.5)$.

Both effects change only the numerical constants, and hence the conclusions of Lemma A.3, and consequently Claim A.8 and Corollary 2.1, hold unchanged with $\gamma = C$ a sufficiently large constant.

Communication accounting. Per node per iteration, gen-AltGDmin-KLL uploads nr scalars (gradient) plus the threshold sketches; below the sketch's capacity, the merged sketch returns the global threshold exactly. The sketch upload is $\mathcal{O}(KH)$ per row, independent of d, while that of exact global thresholding grows as $\Theta(pd/L)$; e.g., at $d = 10^4$, the per-node upload is 194.6k scalars vs. 759.2k. Downloads are $nr + n$ per node, the server merge cost is $\mathcal{O}(nLK)$, and there are two communication rounds per iteration.

Below we combine Corollary 2.1 and Lemma C.2.

Effect on Probability. Estimating row thresholds in sparsification operator $\mathcal{T}$ via federated KLL sketch in each iteration $t \in [T]$ we will need n KLL sketches. Hence using union bound and Corollary 2.1, Lemma C.2 w.p. at least $1 - Tn\delta - \mathcal{O}(d^{-1})$ we get $\boldsymbol{SD}_F(\boldsymbol{U}_T, \boldsymbol{U}^*) \leq \epsilon$. Setting $T = C\kappa^2 \log(1/\epsilon)$ as in Corollary 2.1 and thus using $\delta = \frac{1}{nd\kappa^2 \log\left(\frac{1}{\epsilon}\right)}$ we get w.p. at least $1 - \mathcal{O}(d^{-1})$, $\boldsymbol{SD}_F(\boldsymbol{U}_T, \boldsymbol{U}^*) \leq \epsilon$, this then implies at each iteration communication cost at each node will be $\mathcal{O}\Big(nL\sqrt{\log\big(Ln_{mx}\log\big(\frac{1}{\epsilon}\big)\big)}\Big)$. Treating κ^2 and H as constant.

Merging L KLL sketches for n row costs $\mathcal{O}\Big(nL^2\sqrt{\log\big(Ln_{mx}\log\big(\frac{1}{\epsilon}\big)\big)}\Big)$. Combining power method and KLL sketch part in Corollary 2.1 gives the main Theorem 2.2 for the federated AltGDmin.

Lemma C.3. *Consider a set of scalars $\boldsymbol{a} = [a_1, a_2, ..., a_d]$ and some $\alpha \leq 1$. Let $a^{(1)} \geq a^{(2)} \geq ... \geq a^{(d)}$ denote the order of elements in $\boldsymbol{a}$. Assume $\boldsymbol{a}$ is partitioned into L subsets, $\boldsymbol{a}_\ell$. Let $\boldsymbol{f}_\ell$ be the set of top αd elements of $\boldsymbol{a}_\ell$ in magnitude. Then $a^{(\alpha d)} \in \cup_{\ell=1}^{L} \boldsymbol{f}_\ell$.*

D Limitations of Related Work

Solving a convex relaxation is both very slow and cannot be distributed efficiently. AltMin is faster, but still not efficient to distribute because it requires solving a full minimization for each block of variables at each iteration. Thus both AltProj and AltMin-U-B-S cannot be federated easily. The latter also has an unreasonably high requirement on outlier tolerance.

The fastest RPCA solution, FactGD (Yi et al., 2016, Algorithm 1), is also expensive to federate. The reason is it alternates two sets of gradient updates for $\boldsymbol{U}$ and for $\boldsymbol{B}$ and the gradients are for the new cost function $\|\boldsymbol{Y} - \boldsymbol{UB}\|_F^2 + \eta\|\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^\top\|_F^2$. For this term $\nabla_U = (\boldsymbol{Y} - \boldsymbol{UB})\boldsymbol{B}^\top + \eta\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^\top)$ and $\nabla_B = \boldsymbol{U}^\top(\boldsymbol{Y} - \boldsymbol{UB}) + \eta(\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^\top)\boldsymbol{B}^\top$. This can be expressed as

$$\begin{aligned}\nabla_U &= \sum_\ell (\boldsymbol{Y}_\ell\boldsymbol{B}_\ell^\top - \boldsymbol{U}\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top) + \eta\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{U} - \eta\boldsymbol{U}\sum_\ell \boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top \\ &= \sum_\ell (\boldsymbol{Y}_\ell\boldsymbol{B}_\ell^\top - \boldsymbol{U}\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top - \eta\boldsymbol{U}\boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top) + \eta\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{U}\end{aligned}$$

and

$$\nabla_B = \boldsymbol{U}^\top(\boldsymbol{Y} - \boldsymbol{UB}) + \eta(\boldsymbol{U}^\top\boldsymbol{U} - \sum_\ell \boldsymbol{B}_\ell\boldsymbol{B}_\ell^\top)\boldsymbol{B}$$

Algorithm 3 Federated AltGDmin Initialization

AltGDmin Initialization:
Nodes $\ell = 1, ..., L$
$\mathcal{X}_\ell^j \leftarrow$ sorted largest $2\alpha pd$ elements of $\Pi_\Phi(\boldsymbol{y}_{(\ell)}^j)$ for all $j \in [n]$
$\mathrm{KLL}_\ell^j = \mathrm{KLLsketch}(\mathcal{X}_\ell^j)$ for all $j \in [n]$ (Algorithm 4)
Push KLL_ℓ^j to center
Central Server
$\mathrm{KLL}_{global}^j = \mathrm{KLLmerge}\{\mathrm{KLL}_\ell^j\}_{\ell=1}^L$ (Algorithm 5)
Set $\mathrm{Threshold}^j \leftarrow \mathrm{KLLquery}(\mathrm{KLL}_{global}^j, 2\alpha pd)$ for all $j \in [n]$ (Algorithm 6 with $\beta = 2\alpha pd$)
Push $\mathrm{Threshold} = \{\mathrm{Threshold}^j\}_{j=1}^n$ to nodes.
Nodes $\ell = 1, ..., L$
$\boldsymbol{S}_{0,(\ell)} \longleftarrow \mathcal{T}_{\{2\alpha p, \mathrm{Threshold}\}}(\Pi_\Phi(\boldsymbol{Y}_{(\ell)}))$
Set $\boldsymbol{S}_{(\ell)} \leftarrow \boldsymbol{S}_{0,(\ell)}$
Federated Power Method for Initial Subspace Estimation:
Central Server
Initialize $\boldsymbol{G}_0 \in \mathbb{R}^{n \times r}$ with i.i.d. $\mathcal{N}(0,1)$ entries
$\boldsymbol{G}_0 \leftarrow \mathrm{QR}(\boldsymbol{G}_0)$
Send $\boldsymbol{G}_0$ to nodes
for $t = 0, 1, \ldots, T_{power} - 1$ **do**
  Nodes $\ell \in [L]$
  Compute $\boldsymbol{M}_{(\ell)} \leftarrow p^{-1}(\boldsymbol{Y}_{(\ell)} - \boldsymbol{S}_{(\ell)})$
  Send $\boldsymbol{M}_{(\ell)}\boldsymbol{M}_{(\ell)}^\top \boldsymbol{G}_t$ to the central server
  Central Server
  $\boldsymbol{G}_{t+1} \leftarrow \mathrm{QR}\Big(\sum_{\ell=1}^L \boldsymbol{M}_{(\ell)}\boldsymbol{M}_{(\ell)}^\top \boldsymbol{G}_t\Big)$
  Send $\boldsymbol{G}_{t+1}$ to nodes
end for
Central Server
Set $\boldsymbol{U}_{00} \leftarrow \boldsymbol{G}_T$
$\boldsymbol{U}_0 \longleftarrow QR(\Pi_{\mu,row}(\boldsymbol{U}_{00}))$.
Push $\boldsymbol{U} \leftarrow \boldsymbol{U}_0$ to nodes.

Algorithm 4 KLLsketch (at node ℓ)

Input: sorted data $\mathcal{X}_\ell = \{x_1 \le \cdots \le x_{\gamma d}\}$, base parameter K, levels H, decay $c = \frac{2}{3}$
Capacities: for each level $h = 1, \ldots, H$ set $K_h = \lceil K\, c^{\,H-h} \rceil$
Initialize empty buffers $C_\ell[h] \leftarrow \emptyset$ for $h = 1, \ldots, H$
for each $x \in \mathcal{X}_\ell$ **do**
  Append x to $C_\ell[1]$
  for $h = 1$ **to** $H - 1$ **do**
    if $|C_\ell[h]| > K_h$ **then**
      Randomly choose parity $b \in \{0, 1\}$
      $\{b = 0$: keep odd indices, $b = 1$: keep even indices$\}$
      Initialize empty list T
      for $i = 1 + b$ **to** $|C_\ell[h]|$ **with stepsize** 2 **do**
        Append $C_\ell[h][i]$ to T
      end for
      Clear $C_\ell[h]$
      Append all items of T to $C_\ell[h+1]$
    end if
  end for
end for
Output sketch: $\mathrm{KLL}_\ell = \{C_\ell[h]\}_{h=1}^H$

Algorithm 5 KLLmerge

Input: Sketches $\mathrm{KLL}_1, \ldots, \mathrm{KLL}_L$, where $\mathrm{KLL}_\ell = \{C_\ell[h]\}_{h=1}^{H}$, base parameter K, number of levels H, decay $c = \frac{2}{3}$

Capacities: for each level $h = 1, \ldots, H$ set

$$K_h = \lceil K\, c^{H-h} \rceil$$

Initialize merged buffers $C_g[h] \leftarrow \emptyset$ for $h = 1, \ldots, H$

Level-wise merge:

for $h = 1$ **to** H **do**
  $C_g[h] \leftarrow \bigcup_{\ell=1}^{L} C_\ell[h]$
  Sort $C_g[h]$ in ascending order
end for

Compression:

for $h = 1$ **to** $H - 1$ **do**
  while $|C_g[h]| > K_h$ **do**
    Randomly choose parity $b \in \{0, 1\}$
    $\{b = 0$: keep odd indices, $b = 1$: keep even indices$\}$
    Initialize empty list T
    for $i = 1 + b$ **to** $|C_g[h]|$ **with stepsize** 2 **do**
      Append $C_g[h][i]$ to T
    end for
    Clear $C_g[h]$
    Append all items of T to $C_g[h + 1]$
    Sort $C_g[h + 1]$ in ascending order
  end while
end for

Output merged sketch:

$$\mathrm{KLL}_{\mathrm{global}} = \{C_g[h]\}_{h=1}^{H}$$

Algorithm 6 KLLquery(sketch, β)

Input: KLL sketch $\mathrm{KLL} = \{C[h]\}_{h=1}^{H}$, target order statistic β

Prepare weighted list:

Initialize empty list $\mathcal{L} \leftarrow \emptyset$
for $h = 1$ **to** H **do**
  for each value $x \in C[h]$ **do**
    Append $(x,\ 2^{h-1})$ to $\mathcal{L}$
  end for
end for

Sort $\mathcal{L}$ in ascending order of values x

Initialize weight $W \leftarrow 0$
for $i = |\mathcal{L}|$ **to** 1 **do**
  Let $(x_i, w_i) = \mathcal{L}[i]$
  $W \leftarrow W + w_i$
  if $W \geq \beta$ **then**
    return x_i
  end if
end for

Computing $\nabla_{\boldsymbol{U}}$ requires one round of data exchanges. Computing $\boldsymbol{B}$ requires first each node sharing $\boldsymbol{B}_\ell \boldsymbol{B}_\ell^\top$, center sums it and sends back $\boldsymbol{B}\boldsymbol{B}^\top$ to nodes; then the nodes compute $(\boldsymbol{B}\boldsymbol{B}^\top)\boldsymbol{B}_\ell$; finally they share $\boldsymbol{U}^\top(\boldsymbol{Y}_\ell - \boldsymbol{U}\boldsymbol{B}_\ell)$ with center; and then share $(\boldsymbol{B}\boldsymbol{B}^\top)\boldsymbol{B}_\ell$ with the center. This entire thing requires three rounds of data exchanges.

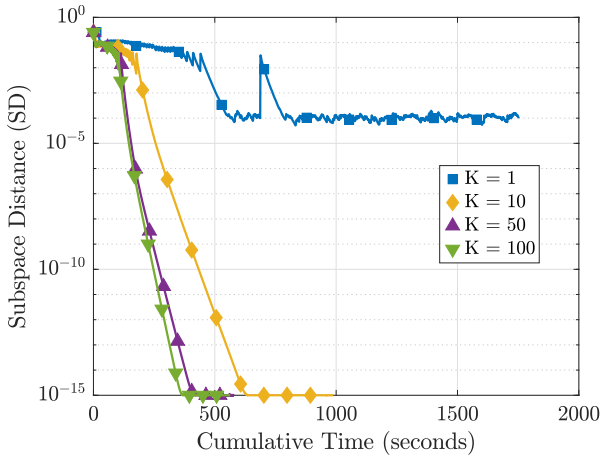


Figure 3: Comparison of $\boldsymbol{SD}_F(\boldsymbol{U}_t, \boldsymbol{U}^*)/\sqrt{r}$ vs Time for different KLL sketch base capacity $K = [1, 10, 50, 100]$. ($n = 1000$, $d = 1000$, $r = 10$, $L = 8$, $p = 0.6$, and $\alpha = 0.1$)

E Details of Experimental Evaluation and More Experiments

E.1 Data generation

All experiments were conducted in MATLAB on a PC with an Intel Core i7 processor having 12 cores and 32 GB RAM. We generate $\boldsymbol{U}^*$ by orthonormalizing a random Gaussian matrix of size $n \times r$. We generate $\boldsymbol{B}^* \in \Re^{r \times d}$ as a random Gaussian matrix and set $\boldsymbol{\Theta}^* = \boldsymbol{U}^* \boldsymbol{B}^*$. For sparse matrix we randomly selected one federated node ℓ and corrupted the same α- fraction of rows and columns of that node. The corruption magnitude is set as $0.3 \cdot \text{median}_{k \in \mathcal{S}_\ell} \|\boldsymbol{\theta}_k^*\|$. For a given p, we generate $\boldsymbol{\Phi} \in \{0, 1\}^{n \times d}$ where each entry is drawn independently from $\text{Bernoulli}(p)$. Federation is emulated with MATLAB `parfor`; the reported wall-clock time includes the emulation's inter-worker communication (within one machine rather than over a network); the transmitted communication volume is quantified in Appendix C. Sample splitting is needed only for the theoretical analysis; as is standard Jain et al. (2013); Nayer & Vaswani (2023); Abbasi & Vaswani (2025), the implemented algorithm reuses all samples at every iteration. The KLL capacity (K) ablation is in Appendix E.2, and real-video results comparing FactGD and gen-AltGDmin ($r = 40$, $\alpha = 0.2$) are in Appendix E.3.

E.2 Ablation on KLL sketch capacity K

For AltGDmin with KLL sketch (AltGDmin-KLL), we set $K = \mathcal{O}(L)$ and $H = \log(\frac{\alpha d}{K})$. Our ablation study Figure 3 confirm that when K is less than L, the row-wise summaries at the center introduce a large threshold approximation error, causing the algorithm to fail. However, for all values where $K \geq L$, the algorithm converges.

E.3 Real World

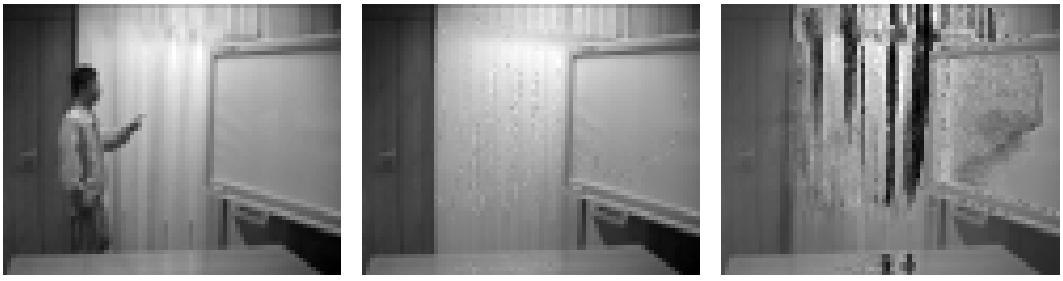
Lobby (LB) Dataset Vaswani et al. (2018): This dataset contains $d = 1205$ images of resolution $n = 128 \times 160 = 20480$. This is a challenging sequence, as the background changes often due to illumination changes, and there are multiple objects in the foreground to detect and subtract.

Meeting Room (MR) dataset Vaswani et al. (2018): The meeting room sequence is a set of 1964 images of resolution $n = 64 \times 80 = 5120$. The first 1755 frames consists of outlier-free data and so we only consider the last 1209 frames. This is a challenging video sequence because the color of the person and the color of the curtain are hard to distinguish.



(a) Original (b) AltGDmin (617.18 sec) (c) FactGD (1965.86 sec)

Figure 4: Background separation for real world Lobby dataset. We report time taken in parentheses. Notice that the FactGD result is blurred and FactGD takes longer.



(a) Original (b) AltGDmin (240s) (c) FactGD (362s)

Figure 5: Background separation for Meeting Room video. We report time taken in parentheses. Notice that the FactGD result is blurred and FactGD takes longer.

We compare the speed and foreground-background separation quality for FactGD and AltGDmin on on real world Lobby (LR) and Meeting Room (MR) dataset in Figure 4 and Figure 5 respectively. We use fixed values $r = 40$ and $\alpha = 0.2$.

F CHOICE OF K: ADDITIONAL DISCUSSION

An important practical question is how to set K. Our result above shows any choice of $K \in O(L)$ suffices to ensure that the algorithm converges. Observe that, with increasing K, the communication cost increases but the compute cost at the nodes decreases. To understand this better, let τ denote the communication delay for sending/receiving one real-valued scalar. Then the total time taken per iteration (using only terms that depend on K) is order $nK \log(\alpha d/K) \cdot \tau + \frac{|\mathbf{\Phi}|}{L} \log(|\mathbf{\Phi}|/LK)$. Consider the $p = 1$ (robust PCA) setting, in this case $|\mathbf{\Phi}| = nd$. Thus, the time simplifies to $n \log(1/K) \max(K\tau, \frac{d}{L})$. If the communication is fast enough so that $\tau \ll d/L$, then the second term dominates. This means using a larger value of K results in faster convergence. In our experiment for different choices of K, we observe this trend; we describe this in Appendix E.2.