Differentially Private Robust Low-Rank Approximation

Raman Arora
Johns Hopkins University
Baltimore, MD-21201
arora@cs.jhu.edu

Vladimir Braverman
Johns Hopkins University
Baltimore, MD-21201
vova@cs.jhu.edu

Jalaj Upadhyay
Johns Hopkins University
Baltimore, MD-21201
jalaj@jhu.edu

Abstract

In this paper, we study the following robust low-rank matrix approximation problem: given a matrix $A \in \mathbb{R}^{n \times d}$, find a rank-$k$ matrix M, while satisfying differential privacy, such that $\|A - M\|_p \leq \alpha \cdot \mathsf{OPT}_k(A) + \tau$, where $\|B\|_p$ is the entry-wise ℓ_p-norm of B and $\mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p$. It is well known that low-rank approximation w.r.t. entrywise ℓ_p-norm, for $p \in [1, 2)$, yields robustness to gross outliers in the data. We propose an algorithm that guarantees $\alpha = \widetilde{O}(k^2), \tau = \widetilde{O}(k^2(n + kd)/\varepsilon)$, runs in $\widetilde{O}((n + d)\mathsf{poly}\ k)$ time and uses $O(k(n + d) \log k)$ space. We study extensions to the streaming setting where entries of the matrix arrive in an arbitrary order and output is produced at the very end or continually. We also study the related problem of differentially private robust principal component analysis (PCA), wherein we return a rank-k projection matrix Π such that $\|A - A\Pi\|_p \leq \alpha \cdot \mathsf{OPT}_k(A) + \tau$.

1 Introduction

Low rank matrix approximation is a well studied problem, where given a data matrix A, the goal is to find a low-rank matrix B that approximates A in the sense that $\mu(A - B)$ is small under some function $\mu(\cdot)$. It finds application in numerous machine learning tasks, such as recommendation systems [10], clustering [9, 25], and learning distributions [2].

Often, the real-world data used in these applications is plagued with gross outliers, and it is desirable to impart robustness to low-rank approximation algorithms against such corruptions. Furthermore, these applications increasingly rely on sensitive data which raises the need for preserving privacy of the underlying data. The focus of this paper, therefore, is to compute a low-rank approximation of a given matrix under a strong privacy guarantee while being robust to outliers in data.

For robustness to outliers, we choose the measure $\mu(\cdot)$ to be the entrywise ℓ_p-norm for $p \in [1, 2)$, defined as $\|A\|_p = (\sum_{i,j} |A_{i,j}|^p)^{1/p}$. It is well known that low-rank approximation w.r.t. entrywise ℓ_p-norm, for $p \in [1, 2)$, yields robustness to gross outliers in the data [5, 7, 22, 23, 24, 29]. To address the need for privacy, we rely on the notion of differential privacy [11] that has become the *de facto* standard for private data analysis in recent years. Formally, we define differential privacy as follows.

Definition 1. A randomized algorithm $\mathcal{M}$ is said to be (ε, δ)-differentially private if for all neighboring datasets, A and A', and all subsets $S \subseteq \mathsf{range}(\mathcal{M})$ in the range of $\mathcal{M}$, we have that $\mathsf{Pr}[\mathcal{M}(A) \in S] \leq e^{\varepsilon} \mathsf{Pr}[\mathcal{M}(A') \in S] + \delta$.

The notion of what makes two datasets neighboring determines the granularity of differential privacy [13]. At the finest scale, we consider two matrices as neighboring if they differ in at most one entry by a unit value [17, 19, 20]; this corresponds to *feature-level privacy*. At the coarsest granularity, two matrices are deemed neighboring if they differ in one row by a unit norm [18, 14]; this

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

corresponds to the *user-level privacy*. Note that since we do not make any boundedness assumption on the entries of the data-matrix, we need to establish a normalized scale to limit the influence of a single entry or a single row of a given matrix. In this paper, we say that two matrices A and A' are neighboring if the matrices are within a unit (entrywise) ℓ_1 ball of each other, i.e., $\|A - A'\|_1 \leq 1$. This notion of neighboring datasets provides stronger guarantees than the feature-level privacy.

We are interested in private robust data analysis, specifically, robust low-rank approximation of a matrix with respect to entrywise ℓ_p-norm for $p \in [1, 2)$, under the constraints of differential privacy. Even without privacy, low-rank matrix approximation with respect to entrywise ℓ_p-norm for $p \neq 2$ is a non-trivial problem: it does not have a closed form solution and computing the optimal low-rank approximation with respect to ℓ_1-norm is known to be NP-hard [16]. A natural question then is whether we can compute a good enough approximation to the best rank-k approximation. This question has formed the basis for many recent results [5, 7, 22, 23, 24, 29]. However, prior to this work, differentially private low-rank approximation with respect to entrywise ℓ_p-norm has been an open problem. We give the first time- and space-efficient differentially private algorithm for low-rank matrix approximation with respect to entrywise ℓ_p-norm.

1.1 Formal Problem Statement and Contributions

In this section, we formally define the problem of differentially private robust low-rank matrix approximation, and state our main results. For the ease of presentation, we assume that $\delta = \Theta(n^{-\log n})$. We use the notation $\widetilde{O}(\cdot)$ to hide $\mathsf{poly}\log$ factors.

Definition 2 (Robust low-rank approximation)**.** Given a matrix $A \in \mathbb{R}^{n \times d}$, and $p \in [1, 2)$, output a rank-k matrix M such that with probability at least $1 - \beta$,

$$\|A - M\|_p \leq \alpha \mathsf{OPT}_k(A) + \tau, \text{ where } \mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p . \tag{1}$$

Our first contribution is Algorithm 1, ROBUST-LRA, which given an input matrix $A \in \mathbb{R}^{n \times d}$ returns a differentially private rank-k approximation to A with a multiplicative approximation factor of $\alpha = O((k \log k)^{2(2-p)/p} \log d \log n)$ and an additive approximation error of $\tau = \widetilde{O}\left(\varepsilon^{-1} k^2 (n + kd)\right)$. In particular, for $p = 1$, we have $\alpha = O(k^2 \log^2 k \log d \log n)$ and $\tau = \widetilde{O}\left(\varepsilon^{-1} k^2 (n + kd)\right)$. We note that the best known algorithm in a non-private setting [29] achieves the same multiplicative factor, albeit with no additive error. Therefore, the price we pay for privacy is in terms of an additional additive error.

In many machine learning problems, e.g. feature selection and representation learning, all we are interested in is recovering the low-dimensional subspace spanned by the data. One such example is principal component analysis using data with gross outliers or corruptions (e.g. face recognition in the presence of occlusions). Of course, the proposed Algorithm 1 can also output the projection matrix associated with the right singular vectors of matrix M with the same accuracy guarantee as for robust low-rank approximation (see Remark 1 for more details). However, the additive error we incur still scales with n whereas intuitively making the basis for a k-dimensional subspace in $\mathbb{R}^d$ should require only adding noise proportional to $k \ll d$. This motivates a slightly different treatment for the robust principal component analysis problem, which can be formulated as follows.

Definition 3 (Robust principal component analysis)**.** Given a matrix $A \in \mathbb{R}^{n \times d}$, output a rank-$k$ orthonormal projection matrix Π such that with probability at least $1 - \beta$,

$$\|A - A\Pi\|_p \leq \alpha \mathsf{OPT}_k(A) + \tau, \text{ where } \mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p . \tag{2}$$

The second main contribution of this paper is an algorithm that returns a differentially private rank-k orthonormal projection matrix with $\alpha = O((kd \log k)^{(2-p)/p} \log^3 d \log n)$ $\tau = \widetilde{O}\left(k^2 d / \epsilon\right)$.

Many variants of differentially private low-rank approximation have been studied in the literature [14, 18, 19, 17, 20, 21, 31, 32] for both the Frobenius norm and spectral norm. We give the first (ε, δ)-differentially private algorithm for robust PCA. Unlike PCA under Frobenius and spectral norm, computing an exact robust PCA is a computationally hard problem (NP-hard when $p = 1$).

Besides the objective function, our work differs from existing work also in terms of the privacy granularity and efficiency. A detailed comparison and review of previous works is presented in Table 1.

Table 1: Comparison of Models for Differentially Private k-Rank Approximation (u and v are unit vectors, e_s is the s-th standard basis, η is an arbitrary constant, $\omega_k := \sigma_k(A) - \sigma_{k+1}(A)$ is the singular value separation, μ is coherence of the matrix A, and $p \in [1, 2)$).

	Assumptions	Accuracy (α, τ)	Metric
Theorem 10	$\|A - A'\|_1 = 1$	$\left(\widetilde{O}(k^{2p(2-p)/2} \log k \log d), \widetilde{O}\left(\frac{k^2(n+kd)}{\varepsilon}\right)\right)$	ℓ_p-norm
Hardt-Roth [18]	$A - A' = e_s v^\top$ μ-coherence	$(\sqrt{2}, \widetilde{O}(\frac{\sqrt{kn}}{\varepsilon} + k\left(\frac{\mu\|A\|_F}{\varepsilon}\right)^{1/2}\left(\frac{d}{n}\right)^{1/4}))$	Frobenius
Upadhyay [32]	$A - A' = uv^\top$	$\left((1+\eta), \widetilde{O}\left(\varepsilon^{-1}(\sqrt{kn} + \sqrt{kd})\right)\right)$	
Kapralov-Talwar [21]	$\|A\|_{op} - \|A'\|_{op} = 1$ σ-value separation	$\left(1, \widetilde{O}\left((nk^3)\,\varepsilon^{-1}\right)\right)$	Spectral
Hardt-Price [17]	$A - A' = e_s e_t^\top$ μ-coherence	$(1, \widetilde{O}(\frac{\sigma_1\sqrt{k\mu \log(\log d\sigma_k/(\omega_k))}}{\varepsilon\omega_k}))$	
Dwork et al. [14]	$A - A' = e_s v^\top$	$\left(1, \widetilde{O}\left(\varepsilon^{-1}k\sqrt{n}\right)\right)$	
Jiang et al. [20]	$A - A' = e_s e_t^\top$	$\left(1, \widetilde{O}\left(n\varepsilon^{-1}\right)\right)$	

2 Basic Preliminaries

One of the key features of differential privacy is that it is preserved under arbitrary post-processing, i.e., an analyst, without additional information about the private database, cannot compute a function that makes an output less differentially private. This is formalized in the form of following lemma:

Lemma 4 (Dwork *et al.* [11])**.** *Let $\mathcal{M}(D)$ be an (ε, δ)-differential private algorithm for a data matrix D , and let h be any function, then any mechanism $\mathcal{M}' := h(\mathcal{M}(D))$ is also (ε, δ)-differentially private.*

A key ingredient in our algorithms is a p-stable distribution which can be defined in terms of a limit of normalized sums of i.i.d. random variables [33].

Definition 5 (p-stable distirbution)**.** A distribution $\mathcal{D}_p$ over $\mathbb{R}$ is called *p-stable*, if there exists $p \geq 0$, such that for any $(v_1, \cdots, v_n) \in \mathbb{R}^n$, and n i.i.d. random variables $X_1, \cdots, X_n$ with distribution $\mathcal{D}_p$, the random variable $\sum_i v_i X_i$ has the same distribution as the variable $\|v\|_p X$, where $X \sim \mathcal{D}_p$.

We use the notation $\mathcal{D}_p^{(r,c)}$ to denote a distribution over $r \times c$ random matrices, where every entry of the matrix is sampled from the distribution $\mathcal{D}_p$. It is known that p-stable distributions exist for all $p \in (0, 2]$ [33], and that Gaussian distribution is 2-stable and the Cauchy distribution is 1-stable. Moreover, one can use the method of Chambers et al. [8] to sample from $\mathcal{D}_p$ $(1 < p < 2)$.

Our analysis uses the fact that $S \sim \mathcal{D}_p^{(r,c)}$ satisfies the no-dilation and no-contraction property [28].

Definition 6 (No-dilation [28])**.** Given a matrix $A \in \mathbb{R}^{n \times d}$, if a matrix $S \in \mathbb{R}^{m \times n}$ satisfies $\|SA\|_p \leq c_1 \|A\|_p$, then S has at most c_1 dilation on A with respect to entrywise ℓ_p-norm.

Definition 7 (No-contraction [28])**.** Given a matrix $A \in \mathbb{R}^{n \times d}$, a matrix $S \in \mathbb{R}^{m \times n}$ has c_2-contraction on A with respect to the entrywise ℓ_p-norm if $\forall x \in \mathbb{R}^d$, $\|SAx\|_p \geq {c_2}^{-1} \|Ax\|_p$.

Our analysis uses recent results from matrix sketching. In particular, we use the fact that we can approximately solve ℓ_p-regression problem using random matrix sketches [29].

Lemma 8 (Song et al. [29])**.** *Let $\Phi \in \mathbb{R}^{\phi \times n}$ be a projection matrix that preserves ℓ_p-norm of a vector for $p \in [1, 2)$ and let $B \in \mathbb{R}^{n \times d}, C \in \mathbb{R}^{n \times c}$ be any matrix. Let $\widetilde{X} := \operatorname{argmin}_{X \in \mathbb{R}^{d \times c}} \|\Phi(BX - C)\|_p , \widehat{X} := \operatorname{argmin}_{X \in \mathbb{R}^{d \times c}} \|BX - C\|_p$, then $\|B\widetilde{X} - C\|_p \leq C_\phi \|B\widehat{X} - C\|_p$ for some constant C_ϕ that depends only on $\log d$.*

Lemma 9 (Song et al. [29])**.** *Given matrices L, N, A of appropriate dimension, let $X^* := \operatorname{argmin}_X \|LXN - A\|_p$. Suppose S and T satisfies c_1-dilation on $LX^*N - A$ and c_2-contraction property on L. Further if $\widehat{X}$ be such that $\|S(L\widehat{X}N - A)T\|_p \leq c \cdot \min_{\mathsf{rank}(X) \leq k} \|S(LXN - A)T\|_p$, then, we have that $\|L\widehat{X}N - A\|_p \leq O(c_1 c_2 c) \cdot \min_{\mathsf{rank}(X) \leq k} \|LXN - A\|_p$.*

Algorithm 1 ROBUST-LRA

Input: Input data matrix $A \in \mathbb{R}^{n\times d}$, target rank k

Output: Rank-k matrix $M \in \mathbb{R}^{n\times d}$

1: **Initialization:** Set the variables $\phi, \psi, s, t, C_\phi, C_\psi, C_s, C_t$ as in Table 2.

2: **Initialization:** Sample $\Phi \in \mathbb{R}^{\phi\times n}$, $\Psi \in \mathbb{R}^{d\times\psi}$, $S \in \mathbb{R}^{s\times n}$, and $T \in \mathbb{R}^{d\times t}$ from distributions $\mathcal{D}_p^{(\phi,n)}$, $\mathcal{D}_p^{(d,\psi)}$, $\mathcal{D}_p^{(s,n)}$, and $\mathcal{D}_p^{(d,t)}$, respectively. All these matrices are made public.

3: **Sample:** $N_1 \in \mathbb{R}^{\phi\times d}$, $N_2 \in \mathbb{R}^{n\times\psi}$, $N_3 \in \mathbb{R}^{s\times t}$ such that $N_1 \sim \mathsf{Lap}(0, C_\psi/\varepsilon)^{n\times\psi}$, $N_2 \sim \mathsf{Lap}(0, C_\phi/\varepsilon)^{\phi\times d}$, and $N_3 \sim \mathsf{Lap}(0, C_s C_t/\varepsilon)^{s\times t}$. Keep N_1, N_2, N_3 private.

4: **Sketch:** Compute $Y_r = \Phi A + N_1$, $Y_c = A\Psi + N_2$.

5: **Sketch:** Compute $Z_r = Y_r T$, $Z_c = SY_c$, $Z = SAT + N_3$.

6: **SVD:** Compute $[U_c, \Sigma_c, V_c] = \mathsf{SVD}(Z_c)$, $[U_r, \Sigma_r, V_r] = \mathsf{SVD}(Z_r)$.

7: ℓ_2-**LRA:** Compute $\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_k \Sigma_r^\dagger U_r^T$, where $[B]_k = \mathrm{argmin}_{r(X)\le k} \|B - X\|_F$.

8: **Output:** $M = Y_c \widehat{X} Y_r$.

Table 2: Values of different variables.

C_ϕ, C_s	C_ψ, C_t	ϕ, ψ, s, t
$O(\log d)$	$O(\log n)$	$O(k \log k \log(1/\delta))$

3 Differentially private robust LRA

In this section, we give an (ε, δ)-differentially private polynomial-time algorithm for robust low-rank approximation. We first discuss algorithmic challenges in extending known techniques and analyses to our problem. We present the proposed algorithm and main results in Section 3.1, and discuss extensions to the general turnstile model and the continual release model in Section 3.2. Proofs of all results are deferred to the supplementary material of this paper.

Two common approaches to preserve privacy are output perturbation [11] and input perturbation [3, 30] of the objective function. In output perturbation, we first compute the output (e.g. rank-k approximation of a given matrix) non-privately and then add appropriately scaled noise to preserve privacy. In input perturbation, we add noise to the private matrix and then compute the output on the noisy matrix. Both these approaches require adding noise to every entry of the given input matrix or to every entry of the non-private output matrix. Consequently, both of these methods would incur an additive error of $O(nd)$. On the other hand, most existing non-private algorithms for robust low-rank approximation either use heuristics and do not have provable guarantees, or they make additional assumptions on the input matrix; the only exception is the work of Song et al. [29]. Again, a naive mechanism to make the algorithm of Song et al. [29] private would incur an additive error of $O(nd)$.

3.1 Proposed Algorithm

It is somewhat tantalizing, from a computational perspective, to attempt approximating a solution to the robust LRA problem using a low-rank approximation with respect to ℓ_2-norm; however, it is well understood that the latter is quite sensitive to even a single outlier. A key idea behind the proposed solution then is based on the following key observation. We can approximate the output of robust low rank approximation using low rank approximation with respect to ℓ_2-norm after sketching the matrix using $S \sim \mathcal{D}_p^{(r,n)}$ and $T \sim \mathcal{D}_p^{(c,d)}$ for some choice of r and s. In particular, p-stable distribution imparts robustness, and the effect of outliers is reduced in the lower dimensional space.

In summary, the proposed algorithms are based on the following three algorithmic primitives: (a) sketching the row-space and column-space of the input matrix, (b) formulating the low-rank matrix approximation problem as a regression problem, and (c) approximating the solution to ℓ_p regression problem by corresponding ℓ_2 regression problem. The analysis, then, carefully bounds the error in approximation for each of the steps above as well as error resulting from the privacy mechanism.

The pseudo-code of the proposed algorithm (ROBUST-LRA) is presented as Algorithm 1. We present values of various variables used in the algorithm in Table 2. Our main result is as follows.

Theorem 10. *Algorithm* ROBUST-LRA *(see Algorithm 1) is* (ε, δ)*-differentially private. Furthermore, given a matrix* $A \in \mathbb{R}^{n \times d}$*, it runs in* $\mathsf{poly}(k, n, d)$ *time,* $\widetilde{O}(k(n + d))$ *space, and outputs a rank* k *matrix* M *such that, with probability* $9/10$ *over the randomness of the algorithm,*

$$\|A - M\|_p \leq O((k \log k \log(1/\delta))^{2(2-p)/p} \log d \log n) \mathsf{OPT}_k(A) + \widetilde{O}(k^2(n+kd) \log^2(1/\delta)/\varepsilon),$$

where $\mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p$.

In particular, for $p = 1$, we get

$$\|A - M\|_p \leq O(k^2 \log^2 k \log^2(1/\delta) \log d \log n) \mathsf{OPT}_k(A) + \widetilde{O}(k^2(n+kd) \log^2(1/\delta)/\varepsilon).$$

Remark 1. Algorithm ROBUST-LRA (Figure 1) outputs a low-rank matrix. However, it is possible to output a low-rank factorization without any loss in efficiency. It can be done by computing the SVD $[U_{\widehat{X}}, \Sigma_{\widehat{X}}, V_{\widehat{X}}]$ of $\widehat{X}$, the QR decomposition of Y_c and Y_r to get orthonormal bases U of column space of Y_c and V of the row space of Y_r. The algorithm then outputs $[UU_{\widehat{X}}, \Sigma_{\widehat{X}}, VV_{\widehat{X}}]$ as a low-rank factorization. The extra running time of this algorithm is $O(\phi^2 d + \psi^2 n + \phi\psi^2) = \widetilde{O}(k^2(n + d))$. This is smaller than $O(nd^2)$ time if one naively factorizes M.

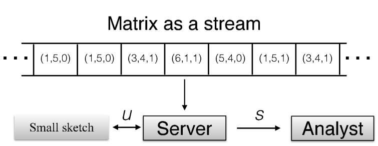
Remark 2 (Additive Error)**.** The additive error in Theorem 10 has a quadratic dependence on k. There is an implicit tradeoff between the additive and multiplicative error as k increases. When k is small, then error due to $\mathsf{OPT}_k(A)$ is higher, and when k is larger, then the additive error is high. For instance when k equals to the rank of the matrix, then we have zero multiplicative error, but additive error is of order $O(k^2 n)$. Note that $O(kn)$ error is unavoidable because we are trying to hide every single entry of the matrix A. Without making additional strong assumptions such as (a) stochastic data, and/or (b) incoherence, and/or (c) bounded norms, $O(kn)$ additive error is perhaps the best we can hope for. Intuitively, we have to privatize a k-dimensional latent representation of our data and therefore at least add noise proportional to kn.

3.2 Extension to Other Models of Differential Privacy

ROBUST LRA can be easily extended to the streaming model of computation [32] and the continual release model [12]. We first define the basic streaming model of computation that we study.

Definition 11 (General turnstile update model)**.** In the *general turnstile update model*, a matrix $A \in \mathbb{R}^{n \times d}$ is streamed in the form of tuple $(\Delta_t, i_t, j_t,)$, where $1 \leq i_t \leq n, 1 \leq j_t \leq d$ and $\Delta_t \in \mathbb{R}$. An update is of the form $A_{i_t, j_t} \leftarrow A_{i_{t-1}, j_{t-1}} + \Delta_t$. The curator is required to output a robust PCA or robust subspace for the matrix at the end of the stream.

For example, in the figure, the server receives an update of 6 to $A_{1,1}$ and it updates the small sketch using an update function, U.



At the end of the stream, the server uses the small sketch and runs an algorithm S to compute the function (low-rank approximation in our context).

We call two streams neighboring if they are formed by neighboring matrices. Note that the private matrix is stored only in the form of linear sketches, therefore, to get an algorithm in the general turnstile streaming model, we initialize $Y_r = N_1, Y_c = N_2$, and $Z = N_3$. Then when we receive $(\Delta_t, i_t, j_t) \in \mathbb{R} \times [n] \times [d]$, we construct a matrix $A^{(t)}$ with all entries zero except for $A^{(t)}_{i_t, j_t} = \Delta_t$. We then update the sketches as follows: $Y_c = Y_c + \Phi A^{(t)}, Y_r = Y_c + A^{(t)}\Psi$, and $Z = Z + SA^{(t)}T$. Once all the updates are made, we simply run the last three steps of ROBUST-LRA. As a result, we get the following corollary.

Corollary 12 (Informal)**.** *Algorithm* ROBUST-LRA *is an* (ε, δ)*-differentially private that on input a private matrix* A *in a general turnstile update model, outputs a rank* k *matrix* M *with the same accuracy guarantee as in Theorem 10.*

ROBUST-LRA can also be extended to the following continual release setting [12].

Definition 13 (Continual release model)**.** In a *continual release model*, a matrix $A \in \mathbb{R}^{n \times d}$ is streamed in the form of tuple (Δ_t, i_t, j_t), where $1 \leq i_t \leq n, 1 \leq j \leq d$ and $\Delta_t \in \mathbb{R}$. An update is of the form $A_{i_t, j_t} \leftarrow A_{i_{t-1}, j_{t-1}} + \Delta_t$. The curator is required to output a robust PCA or robust subspace for the matrix streamed up until any time $t \leq T$.

Algorithm 2 ROBUST-PCA

Input: Input data matrix $A \in \mathbb{R}^{d \times n}$, target rank k

Output: Rank-k projection matrix $\Pi \in \mathbb{R}^{d \times d}$

1: **Initialization:** Set the variables $\phi, \psi, t, C_\phi, C_\psi, C_t$ as in Table 2.

2: **Initialization:** Sample $\Phi \in \mathbb{R}^{\phi \times d}, \Psi \in \mathbb{R}^{n \times \psi}, S \in \mathbb{R}^{s \times d}$, and $T \in \mathbb{R}^{n \times t}$ from distributions $\mathcal{D}_p^{(\phi,d)}, \mathcal{D}_p^{(n,\psi)}, \mathcal{D}_p^{(s,d)}$, and $\mathcal{D}_p^{(n,t)}$, respectively. All these matrices are made public.

3: **Sample:** $N_1 \in \mathbb{R}^{\phi \times t}, N_2 \in \mathbb{R}^{d \times \psi}$ such that $N_1 \sim \mathsf{Lap}(0, C_\phi C_t/\varepsilon)^{\phi \times t}, N_2 \sim \mathsf{Lap}(0, C_\psi/\varepsilon)^{d \times \psi}$. Keep N_1, N_2 private.

4: **Sketch:** Compute $Y_r = \Phi A^T T + N_1$ and $Y_c = A^T \Psi + N_2$. $Z_c = \Phi Y_c$ and $Z = Y_r$

5: **SVD:** Compute $[U_c, \Sigma_c, V_c] = \mathsf{SVD}(Z_c)$,

6: $[U_r, \Sigma_r, V_r] = \mathsf{SVD}(Y_r)$.

7: ℓ_2-**LRA:** Compute $\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_k \Sigma_r^\dagger U_r^T$, where $[B]_k = \operatorname{argmin}_{r(X) \leq k} \|B - X\|_F$.

8: **Pick:** a permutation matrix $Q \in R^{\phi \times \phi}$.

9: **Compute:** the full SVD of $Y_c \widehat{X}$, $[U', \Sigma', V']$. Set $U = U'Q$, $\Sigma = \Sigma' Q$, and $P = \Phi^\dagger (U\Sigma)^\dagger$.

10: **Output:** $\Pi = PU\Sigma(\Phi PU\Sigma)^\dagger \Phi$.

For outputting a low-rank approximation in the continual release model, we can use the generic transformation to store a binary tree that is constructed over the privatized sketches of the updates as its leaves [12]. When a new query for a range of updates is made, we accumulate the sketches of the dyadic partition of the range to compute the sketches for that range. We then compute the last three steps of ROBUST-LRA. We have the following result.

Corollary 14. *Algorithm* ROBUST-LRA *is an* (ε, δ)*-differentially private algorithm that on input matrix* A *in a streaming manner, runs in time* $\mathsf{poly}(k, n, d, \log T)$ *and outputs a rank* k *matrix* $M^{(t)}$ *in the continual release model over* T *time epochs, such that, with probability at least* $9/10$, $\|A^{(t)} - M^{(t)}\|_p \leq O((k \log k \log(1/\delta))^{2(2-p)/p} \log d \log n) \mathsf{OPT}_k(A^{(t)}) + \widetilde{O}(k^2(n + kd) \log T)$, *where* $\mathsf{OPT}_k(A)$ *is as in Theorem 10, and* $A^{(t)}$ *is the matrix up to* t *time epochs.*

4 Differentially Private Robust Principal Component Analysis

In this section, we focus on the problem of robust PCA under the constraints of differential privacy. We first present the proposed algorithm and then discuss extensions to the general turnstile model continual release model. Proofs of all results are deferred to the supplementary material of this paper.

The key ideas underlying the proposed algorithm, ROBUST-PCA (see Algorithm 2 for the pseudocode), and its analysis, essentially follow the techniques developed in the previous section for ROBUST-LRA, but with a couple of small modifications to get a better additive error. First, we only generate two sketches, $Y_r = \Phi A^T T + N_1$ and $Y_c = A^T \Psi + N_2$, where Ψ, Φ, T are random sketching matrices and N_1, N_2 are noise matrices as defined in Algorithm 2. Second, we solve a slightly different optimization problem:

$$\min_{\mathsf{rank}(Y) \leq k} \left\| A^T - (PU\Sigma) Y (\Phi A^T) \right\|_F,$$

where P, U, Σ are as formed in Algorithm 2. We show that $(\Phi U \Sigma P)^\dagger$ is an approximate solution to $\min_X \left\| \Phi(A^T - PU\Sigma X \Phi A^T) T \right\|_p$. The rest of the proof then follows the same steps as in the proof of Theorem 10. In addition, we also show that Π is an orthonormal rank-k projection matrix. The above exposition focuses on the non-private setting for the sake of simplicity. The proof is more involved due to noise matrices added for privacy.

We show the following guarantee for the proposed algorithm.

Theorem 15. *Algorithm* ROBUST-PCA, *(see Algorithm 2), is* (ε, δ)*-differentially private. Further, given a matrix* $A \in \mathbb{R}^{n \times d}$ *with* $\mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p$, *it runs in time* $\mathsf{poly}(k, n, d)$, *space* $\widetilde{O}(k(n + d))$, *and outputs a rank* k *orthonormal projection matrix* Π *such that, with probability* $9/10$ *over the random coin tosses of the algorithm,*

$$\|A - A\Pi\|_p \leq O((k \log k \log(1/\delta))^{2(2-p)/p} \log n \log^3 d) \mathsf{OPT}_k(A) + \widetilde{O}(k^2 d \log n/\varepsilon).$$

In particular, when $p = 1$, we have the following guarantee:

$$\|A - A\Pi\|_p \leq O(k^2 \log n \log^3 d \log^2 k \log^2(1/\delta))\mathsf{OPT}_k(A) + \widetilde{O}(k^2 d \log n/\varepsilon).$$

We note that ROBUST-PCA yields a smaller additive error than ROBUST-LRA by a factor of n/d, but at the expense of an additional multiplicative factor of $\log^2(d)$. Therefore, in settings where $\mathsf{OPT}_k(A)$ is small (e.g. when A is nearly low rank), ROBUST-PCA enjoys a much better accuracy guarantee.

Extension to Other Models of Differential Privacy. We can extend ROBUST-PCA to the streaming model of computation [32] and the continual release model [12] as in Section 3.2. We can also extend ROBUST-PCA to the local model of differential privacy. Local differential privacy has gained a lot of attention recently [1, 15]. In the local privacy model, there is no central database of private data. Instead, each individual has its own data element (a database of size one), and sends a report based on its own datum in a differentially private manner.

Formally, we consider the database $X = [x_1, \cdots, x_n]^\top$ as a collection of n elements (rows) from some domain $\mathcal{X} \subseteq \mathbb{R}^d$, with each row held by a different individual. The i^{th} individual has access to ε_i-*local randomizer*, $R_i : \mathcal{X} \to W$, which is an ε_i-differentially private algorithm that takes as input a database of size $n = 1$. We assume that the algorithms may interact with the database only through local randomizers. We can then define local differential privacy as follows [13]. An algorithm is ε-locally differentially private if it accesses the database X via the local randomizers, $R_1(x_1), \ldots, R_n(x_n)$, where $\max\{\varepsilon_1, \ldots, \varepsilon_n\} \leq \varepsilon$.

We note that what we have defined above is a non-interactive local differential privacy algorithm where an individual only sends a single message to the server. It was argued in Smith et al. [27] that it is more desirable to have as few rounds of interactions as possible from an implementation point of view. In fact, existing large-scale deployments are limited to one that are non-interactive. Therefore, we limit our study to what is possible in the non-interactive variant of local differential privacy.

We extend Algorithm 2 to an ε-locally-differentially private protocol, LOCAL-ROBUST-PCA, where every user $1 \leq i \leq n$ has a row $A_{i:}$ of the data matrix and sends only one message to the server. We show that the output produced by the server after a run of LOCAL-ROBUST-PCA is a rank-k orthonormal projection matrix $\Pi \in \mathbb{R}^{d \times d}$ such that

$$\|A - A\Pi\|_p \leq O(\log n \log^3 d \, (k \log k \log(1/\delta))^{2(2-p)/p})\mathsf{OPT}_k(A) + \widetilde{O}(k^2 nd/\varepsilon).$$

The above guarantee is non-trivial when $\|A\|_p \gg nd$. Such an assumption is often valid in practical settings with large corruption to data matrices.

5 Discussion

In this paper, we present differentially private algorithms for robust low-rank approximation and for robust principal component analysis. In addition, we study extensions of our algorithms to a continual release model, the streaming model of computation, and the local model of differential privacy.

The bounds we provide involve a multiplicative factor that depends on the target rank k. Such a dependence was deemed necessary in non-private settings. In particular, Song et al. [29] show that if the exponential time hypothesis is true, then any linear-sketch based polynomial time algorithm for robust rank-k factorization incurs an $\Omega(k^{1/2-\gamma})$ multiplicative approximation for some $\gamma \in (0, 0.5)$ that can be arbitrarily small. It is not clear immediately if such a result still holds when we allow an additive error in the approximation, as is the case here.

Acknowledgements

This research was supported in part by NSF BIGDATA grant IIS-1546482, NSF BIGDATA grant IIS-1838139, NSF Career CCF-1652257, and ONR Award N00014-18-1-2364.

References

[1] Apple tries to peek at user habits without violating privacy. *The Wall Street Journal*, 2016.

[2] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *Learning Theory*, pages 458–469. Springer, 2005.

[3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy. In *FOCS*, pages 410–419, 2012.

[4] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[5] J Paul Brooks, José H Dulá, and Edward L Boone. A pure l1-norm principal component analysis. *Computational statistics & data analysis*, 61:83–98, 2013.

[6] Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. *arXiv preprint arXiv:1711.04740*, 2017.

[7] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[8] John M Chambers, Colin L Mallows, and BW Stuck. A method for simulating stable random variables. *Journal of the american statistical association*, 71(354):340–344, 1976.

[9] Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *STOC*, pages 163–172. ACM, 2015.

[10] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *STOC*, pages 82–90. ACM, 2002.

[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.

[12] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In Leonard J. Schulman, editor, *STOC*, pages 715–724. ACM, 2010.

[13] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[14] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis. In *STOC*, pages 11–20, 2014.

[15] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM, 2014.

[16] Nicolas Gillis and Stephen A Vavasis. On the complexity of robust pca and ℓ_1-norm low-rank matrix approximation. *arXiv preprint arXiv:1509.09236*, 2015.

[17] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2861–2869. Curran Associates, Inc., 2014.

[18] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 1255–1268. ACM, 2012.

[19] Moritz Hardt and Aaron Roth. Beyond worst-case analysis in private singular vector computation. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 331–340. ACM, 2013.

[20] Wuxuan Jiang, Cong Xie, and Zhihua Zhang. Wishart mechanism for differentially private principal components analysis. *arXiv preprint arXiv:1511.05680*, 2015.

[21] Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In *SODA*, volume 5, page 1. SIAM, 2013.

[22] Qifa Ke and Takeo Kanade. Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.

[23] Eunwoo Kim, Minsik Lee, Chong-Ho Choi, Nojun Kwak, and Songhwai Oh. Efficient l_{1}-norm-based low-rank matrix approximations for large-scale problems using alternating rectified gradient method. *IEEE transactions on neural networks and learning systems*, 26(2):237–251, 2015.

[24] Panos P Markopoulos, Sandipan Kundu, Shubham Chamadia, and Dimitrios Pados. Efficient l1-norm principal-component analysis via bit flipping. *IEEE Transactions on Signal Processing*, 2017.

[25] Frank McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537. IEEE, 2001.

[26] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE, 2007.

[27] A. Smith, A. Thakurata, and J. Upadhyay. Is Interaction Necessary for Distributed Private Learning? *To Appear in IEEE Symposium for Security & Privacy*, 2017.

[28] Christian Sohler and David P Woodruff. Subspace embeddings for the l 1-norm with applications. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 755–764. ACM, 2011.

[29] Zhao Song, David P. Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 688–701, 2017.

[30] Jalaj Upadhyay. Random Projections, Graph Sparsification, and Differential Privacy. In *ASIACRYPT (1)*, pages 276–295, 2013.

[31] Jalaj Upadhyay. Differentially private linear algebra in the streaming model. *arXiv preprint arXiv:1409.5414*, 2014.

[32] Jalaj Upadhyay. The price of privacy for low-rank factorization. In *Advances in Neural Information Processing Systems*, pages 4180–4191, 2018.

[33] Vladimir M Zolotarev. *One-dimensional stable distributions*, volume 65. American Mathematical Soc., 1986.

Supplementary Material to "Differentially Private Robust Low-Rank Approximation"

A Auxiliary Lemma

We need the following results about product of pseudo-inverse.

Fact 16. *If A has a left-inverse, then $A^\dagger = (A^\mathsf{T} A)^{-1} A^\mathsf{T}$ and if A has right-inverse, then $A^\dagger = A^\mathsf{T}(AA^\mathsf{T})^{-1}$.*

Theorem 17 (Product of pseudoinverse)**.** *Let A and B be conforming matrices and either,*

1. *A has orthonormal columns (i.e., $A^\mathsf{T} A$ is an identity matrix) or,*
2. *B has orthonormal rows (i.e., BB^T is an identity matrix),*
3. *A has all columns linearly independent (full column rank) and B has all rows linearly independent (full row rank) or,*
4. *$B = A^\mathsf{T}$ (i.e., B is the conjugate transpose of A),*

then $(AB)^\dagger = B^\dagger A^\dagger$.

The following lemma follows from Holder's inequality and minimality of $\hat{x}$.

Lemma 18 (ℓ_2 relaxation of ℓ_p regression)**.** *Let $p \in [1, 2)$. For any $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$, define $x^* = \operatorname{argmin} \|Ax - b\|_p$ and $\hat{x} = \operatorname{argmin} \|Ax - b\|_2$. Then, $\|Ax^* - b\|_p \leq \|A\hat{x} - b\|_p \leq n^{1/p-1/2} \|Ax^* - b\|_p$.*

Lemma 19 (Subsampling and rescaling lemma)**.** *Let k be a parameter and $s = O(k \log k)$. Let $S \in \mathbb{R}^{s \times d}$ be a random matrix with every entries sampled i.i.d. from $C(0, 1)$, Cauchy distribution with variance 1, and scaled by $1/s$. Let $A, B \in \mathbb{R}^{d \times p}$ be real-valued matrices such that $\mathsf{rank}(A) \leq k$. Let $S' \in \mathbb{R}^{s \times d}$ be a matrix with i-th row $S'_{i:}$ defined using the following probability distribution*

$$S'_{i:} = \begin{cases} \frac{s}{k} S_{i:} & \text{with probability } k/s \\ 0^d & \text{otherwise} \end{cases}$$

Then with probability $24/25$, $\|S'A - SB\|_1 \leq O(\log d) \|A - B\|_1$.

Proof. Let $\tilde{p} = sp$. We use the notation $C(0, 1)$ to define Cauchy random variable with variance 1. We call a random variable c *half clipped Cauchy random variable* if $c \sim |C(0, 1)|$. Define G_u to be event when half clipped random Cauchy variable c_u, such that $c_u < 100\tilde{p}$. Let $\mathsf{Good} = \cap \mathsf{G}_u$. We can easily compute the probability that G_u and Good happens. Using the pdf of Cauchy, we have

$$\mathsf{Pr}[\mathsf{G}_u] = 1 - \frac{2}{\pi} \tan^{-1}(1/100sp) \geq 1 - \frac{1}{50\pi\tilde{p}}.$$

Union bound then implies that $\mathsf{Pr}[\neg\mathsf{Good}] \leq \frac{1}{50\pi}$. Using total probability theorem and Markov's inequality, we have

$$\begin{aligned} \mathsf{Pr}[\|S'A - SB\|_1 \geq (100 \log p) \|A - B\|_1] \leq \mathsf{Pr}[\|S'A - SB\|_1 \geq (100 \log p) \|A - B\|_1 |\mathsf{Good}] \\ + \mathsf{Pr}[\neg\mathsf{Good}] \leq \frac{\mathsf{E}[\|S'A - SB\|_1 |\mathsf{Good}]}{100 \log p \|A - B\|_1} + \frac{1}{50\pi} \end{aligned} \tag{3}$$

Let $A_{:i}$ denote the i-th column of the matrix A. Then

$$\begin{aligned}
\mathsf{E}[\|S'A - SB\|_1 |\mathsf{Good}] &= \sum_{i=1}^{p} \mathsf{E}[\|S'A_{:i} - SB_{:i}\|_1 |\mathsf{Good}] \\
&= \sum_{i=1}^{p} \mathsf{E}\left[\sum_{j=1}^{s} \left|\sum_{\ell=1}^{d} \frac{1}{s} S'_{j,\ell} A_{\ell,i} - S_{j,\ell} B_{\ell,i}\right| |\mathsf{Good}\right] \\
&= \frac{1}{s} \sum_{i=1}^{p} \sum_{j=1}^{s} \mathsf{E}[\bar{s}_{i,j} |\mathsf{Good}] \\
&= \frac{1}{s} \sum_{i=1}^{p} \sum_{j=1}^{s} \|A_{:i} - B_{:i}\| \, \mathsf{E}[c_{i+p(j-1)} |\mathsf{Good}], \qquad (4)
\end{aligned}$$

where $\bar{s}_{i,j} \sim |C(0, \|A_{:i} - B_{:i}\|_1)|$ and $c_{i+p(j-1)}$ is half clipped Cauchy random variable, i.e., $c_{i+p(j-1)} \sim |C(0,1)|$. We next compute $\mathsf{E}[c_{i+p(j-1)}|\mathsf{Good}]$. Let $u = i + p(j-1)$.

Since for any random variable X and Y, $\mathsf{E}[X] = \sum_y \mathsf{Pr}[Y = y]\mathsf{E}[X|Y = y]$), we have

$$\begin{aligned}
\mathsf{E}[c_u|\mathsf{G}_u] &= \mathsf{Pr}[\mathsf{Good}|\mathsf{G}_u]\mathsf{E}[c_u|\mathsf{G}_u \cap \mathsf{Good}] + \mathsf{Pr}[\neg\mathsf{Good}|\mathsf{G}_u]\mathsf{E}[c_u|\mathsf{G}_u \cap \neg\mathsf{Good}] \\
&\geq \mathsf{E}[c_u|\mathsf{G}_u \cap \mathsf{Good}]\mathsf{Pr}[\mathsf{Good}|\mathsf{G}_u] = \mathsf{E}[c_u|\mathsf{Good}]\mathsf{Pr}[\mathsf{Good}|\mathsf{G}_u],
\end{aligned}$$

In other words,

$$\mathsf{E}[c_u|\mathsf{Good}] \leq \frac{\mathsf{E}[c_u|\mathsf{G}_u]\mathsf{Pr}[\mathsf{G}_u]}{\mathsf{Pr}[\mathsf{Good}|\mathsf{G}_u]}.$$

Now $\mathsf{E}[c_u|\mathsf{G}_u] = \frac{\log(1+(sp)^2)}{\pi \mathsf{Pr}[\mathsf{G}_u]}$. Using Bayes theorem and the fact that $\mathsf{Pr}[\mathsf{Good}] = \mathsf{Pr}[\mathsf{Good} \cap \mathsf{G}_u]$, this implies that

$$\begin{aligned}
\mathsf{E}[c_u|\mathsf{Good}] &\leq \frac{\mathsf{E}[c_u|\mathsf{G}_u]\mathsf{Pr}[\mathsf{G}_u]}{\mathsf{Pr}[\mathsf{Good}]} = \frac{\log(1+(sp)^2)}{\pi \mathsf{Pr}[\mathsf{G}_u]} \frac{\mathsf{Pr}[\mathsf{G}_u]}{\mathsf{Pr}[\mathsf{Good}]} \\
&\leq \frac{\log(1+(sp)^2)}{\pi(1 - 1/(50\pi))} \leq 2\log(sp).
\end{aligned}$$

We can now bound equation (4) as below:

$$\mathsf{E}\left[\|S'A - SB\|_1 |\mathsf{Good}\right] = \frac{1}{s} \sum_{i=1}^{p} \sum_{j=1}^{s} \|A_{:i} - B_{:i}\|_1 \, \mathsf{E}[c_{i+p(j-1)}|\mathsf{Good}] = 2\log p \|A - B\|_1.$$

Plugging this in equation (3), we have

$$\begin{aligned}
\mathsf{Pr}[\|S'A - SB\|_1 \geq 100 \log p \, \|A - B\|_1] &\leq \frac{\mathsf{E}[\|S'A - SB\|_1 |\mathsf{Good}]}{50 \log p \, \|A - B\|_1} + \frac{1}{5\pi} \\
&\leq \frac{2 \log sp}{100 \log p} + \frac{1}{50\pi} \leq \frac{1}{25},
\end{aligned}$$

where the last inequality holds because $s \leq p$. This completes the proof. □

B Missing Proofs

B.1 Proof of Theorem 10

Reminder of Theorem 10. Algorithm ROBUST-LRA, (see Algorithm 1), is (ε, δ)-differentially private. Furthermore, given a matrix $A \in \mathbb{R}^{n \times d}$, it runs in $\mathsf{poly}(k, n, d)$ time, $\widetilde{O}(k(n+d))$ space, and outputs a rank k matrix M such that, with probability $9/10$ over the randomness of the algorithm,

$$\|A - M\|_p \leq O((k \log k \log(1/\delta))^{2(2-p)/p} \log d \log n)\mathsf{OPT}_k(A) + \widetilde{O}(k^2(n+kd)\log^2(1/\delta)/\varepsilon),$$

Proof of Theorem 10. We first give the privacy proof of Theorem 10. Let A and A' be neighboring matrices, i.e., $\|A - A'\| 1 \leq 1$. We argue the privacy result for $p = 1$ (the case for $p \in (1, 2)$ follows from invoking Holder's inequality. The private matrix is used to generate three sketches: Y_c, Y_r, and Z. Since Φ, Ψ, S, and T are sampled from distribution of random matrices that preserves the ℓ_1-norm, we have that $\|\Phi(A - A')\| 1 \leq C_\phi \|A - A'\| 1 = C_\phi$ with probabilty at least $1 - \delta$. The privacy proof now follows from Laplace mechanism. Note that for $p \in (1, 2)$, we have $\|\Phi(A - A')\|_p \leq C_\phi \|(A - A')\|_p \leq C_\phi \|(A - A')\|_1$.

We now give the utility proof of Theorem 10. Let

$$U^*, V^* := \underset{\substack{U \in \mathbb{R}^{n \times k} \\ V \in \mathbb{R}^{k \times d}}}{\operatorname{argmin}} \|UV - A)\|_p$$

Our proof relies on three fundamental techniques.

Two fundamental techniques. The first fundamental technique is to use the fact that solving generalized linear regression problem in the projected space gives an approximate solution to the original generalized regression problem. The second main idea is the reduction from low-rank approximation to a generalized linear regression problem.

Let $B = A + S^\dagger N_3 T^\dagger$, then $SBT = Z$. Also let $C = A + \Phi^\dagger N_1$, then $\Phi C = Y_r$. Let

$$\begin{aligned} \widetilde{V} &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|\Phi(U^* V - C)\|_p, \\ \widehat{V} &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|\Phi(U^* V - C)\|_F, \\ V' &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|U^* V - C)\|_p \end{aligned}$$

Then using Lemma 8 and the fact that $\|U^* V' - B)\|_p \leq \|U^* V - B)\|_p$ for all V (and in particular, V^*), we have

$$\begin{aligned} \left\|U^* \widetilde{V} - C)\right\|_p &\leq O(C_\phi) \|U^* V' - C)\|_p \\ &\leq O(C_\phi) \left(\|U^* V^* - A)\|_p + \left\|\Phi^\dagger N_1\right\|_p\right). \end{aligned} \tag{5}$$

Since $\widehat{V}_{:i} = (\Phi U^*)^\dagger \Phi C_{:i} = \operatorname{argmin}_x \|\Phi(U^* x - C_{:i})\|_F$, using Holder's inequality, we have

$$\begin{aligned} \left\|(U^* \widehat{V} - C)\right\|_p &= \sum_{i=1}^d \left\|(U^* \widehat{V}'_{:i} - C_{:i})\right\|_p \\ &\leq \sqrt{\phi} \sum_{i=1}^d \left\|(U^* \widetilde{V}_{:i} - C)\right\|_p \qquad \text{(Lemma 18)} \\ &= \sqrt{\phi} \left\|U^* \widetilde{V} - C)\right\|_p. \end{aligned}$$

Combining this with equation (5), we have

$$\left\|(U^* \widehat{V} - C)\right\|_p \leq O(C_\phi \sqrt{\phi}) \|U^* V^* - A)\|_p + O(C_\phi \sqrt{\phi}) \left\|\Phi^\dagger N_1\right\|_p.$$

Moreover,

$$\left\|(U^* \widehat{V} - C)\right\|_p \geq \left\|U^* \widehat{V} - A)\right\|_p - \left\|\Phi^\dagger N_1\right\|_p.$$

Combining the last two inequalities gives us

$$\left\|(U^* \widehat{V} - A)\right\|_p \leq O(C_\phi \sqrt{\phi}) \|U^* V^* - A)\|_p + O(C_\phi \sqrt{\phi}) \left\|\Phi^\dagger N_1\right\|_p. \tag{6}$$

Further let,

$$\widetilde{U} := \underset{U \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \left\| (U\widehat{V} - A)\Psi \right\|_p,$$
$$\widehat{U} := \underset{U \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \left\| (U\widehat{V} - A)\Psi \right\|_F,$$
$$U' := \underset{U \in \mathbb{R}^{n \times k}}{\operatorname{argmin}} \left\| U\widehat{V} - A) \right\|_p$$

Then using Lemma 8 and the fact that $\left\| U'\widehat{V} - B) \right\|_p \leq \left\| U\widehat{V} - B) \right\|_p$ for all U (and in particular, U^*), we have

$$\begin{aligned} \left\| \widetilde{U}\widehat{V} - A) \right\|_p &\leq O(C_\psi) \left\| U'\widehat{V} - A) \right\|_p \\ &\leq O(C_\psi) \left\| (U^*\widehat{V} - A) \right\|_p. \end{aligned} \tag{7}$$

We know that $\widehat{U}_{i:} = A_{:i}\Psi(\widehat{V}\Psi)^\dagger = \operatorname{argmin}_x \left\| (x\widehat{V} - A_{i:})\Psi \right\|_F$. Equation (7) then gives us

$$\begin{aligned} \left\| (\widehat{U}\widehat{V} - A) \right\|_p &= \sum_{i=1}^n \left\| (\widehat{U}_{i:}\widehat{V} - A_{i:}) \right\|_p \\ &\leq \sqrt{\psi} \sum_{i=1}^d \left\| (\widetilde{U}_{i:}\widehat{V} - A_{i:}) \right\|_p && \text{(Lemma 18)} \\ &= \sqrt{\psi} \left\| \widetilde{U}\widehat{V} - A) \right\|_p \\ &\leq O(C_\psi \sqrt{\psi}) \left\| (U^*\widehat{V} - A) \right\|_p. \end{aligned} \tag{8}$$

Substituting the value of $\widehat{U} = A\Psi(\widehat{V}\Psi)^\dagger$,

$$\left\| A\Psi(\widehat{V}\Psi)^\dagger\widehat{V} - A \right\|_p \leq O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| U^*V^* - A \right\|_p + O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| \Phi^\dagger N_1 \right\|_p \tag{9}$$

Recall that $Y_c = A\Psi + N$ by the construction in the algorithm. Using subadditivity of norms and substituting $\widehat{V} = (\Phi U^*)^\dagger Y_r$, we have

$$\begin{aligned} \left\| Y_c(\widehat{V}\Psi)^\dagger\widehat{V} - A \right\|_p &\leq \left\| A\Psi(\widehat{V}\Psi)^\dagger\widehat{V} - A \right\|_p + \left\| N(\widehat{V}\Psi)^\dagger\widehat{V} \right\|_p && \text{(subadditivity)} \\ &\leq O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| U^*V^* - A) \right\|_p + \left\| N_2(\widehat{V}\Psi)^\dagger\widehat{V} \right\|_p && \text{(equation (9))} \\ &\quad + O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| \Phi^\dagger N_1 \right\|_p \end{aligned} \tag{10}$$

Now again from subadditivity, we have

$$\left\| Y_c(\widehat{V}\Psi)^\dagger\widehat{V} - B \right\|_p \leq \left\| Y_c(\widehat{V}\Psi)^\dagger\widehat{V} - A \right\|_p + \left\| S^\dagger N_3 T^\dagger \right\|_p$$

Combining equation (10) with the above inequality, we get

$$\begin{aligned} \left\| Y_c(\widehat{V}\Psi)^\dagger\widehat{V} - B \right\|_p &\leq O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| U^*V^* - A) \right\|_p \\ &\quad + \left\| N_2(\widehat{V}\Psi)^\dagger\widehat{V} \right\|_p + \left\| S^\dagger N_3 T^\dagger \right\|_p \\ &\quad + O(C_\phi C_\psi \sqrt{\psi\phi}) \left\| \Phi^\dagger N_1 \right\|_p. \end{aligned}$$

Further, since U^* has rank at most k and $\widehat{V} = (\Phi U^*)^\dagger Y_r$, $(\widehat{V}\Psi)^\dagger(\Phi U^*)^\dagger \Phi$ has rank at most k. This implies that

$$\begin{aligned}
\min_{r(X)\leq k} \|Y_c X Y_r - B\|_p &\leq \left\|Y_c(\widehat{V}\Psi)^\dagger(\Phi U^*)^\dagger Y_r - B\right\|_p && \text{(minimality)} \\
&\leq O(C_\phi C_\psi \sqrt{\psi\phi}) \left\|U^* V^* - A)\right\|_p + \left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\
&\quad + \left\|S^\dagger N_3 T^\dagger\right\|_p + O(C_\phi C_\psi \sqrt{\psi\phi}) \left\|\Phi^\dagger N_1\right\|_p && (11)
\end{aligned}$$

Third fundamental technique. The last fundamental technique that we use is that an approximate solution of low-rank problem in the projected space also gives an approximate solution of the original low-rank problem. Let $Q = SAT$ and

$$\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_k \Sigma_r^\dagger U_r^T.$$

Let $\widetilde{X} := \operatorname{argmin}_{\mathsf{rank}(Y)\leq k} \|S Y_c X Y_r T - Z\|_p$. To show that we can achieve an approximate solution of a low-rank problem in the projected space, we use Holder's inequality. More precisely, we have the following set of inequalities:

$$\begin{aligned}
\left\|S Y_c \widehat{X} Y_r T - Z\right\|_p &\leq \sqrt{st}\left\|S Y_c \widehat{X} Y_r T - Z\right\|_F \\
&= \sqrt{st} \min_{\mathsf{rank}(Y)\leq k} \left\|S Y_c \widehat{X} Y_r T - Z\right\|_F && \text{(by definition)} \\
&\leq \sqrt{st}\left\|S Y_c \widetilde{X} Y_r T - Z\right\|_F && \text{(by minimality)} \\
&\leq \sqrt{st}\left\|S Y_c \widetilde{X} Y_r T - Z\right\|_p \\
&= \sqrt{st} \min_{\mathsf{rank}(Y)\leq k} \|S Y_c X Y_r T - Z\|_p, && (12)
\end{aligned}$$

where the first and last inequalites follow from Holder's inequality, second inequality from the minimality, the first equality is due to [32], and the last equality is by definition. Using Lemma 9, we have

$$\left\|Y_c \widehat{X} Y_r - B\right\|_p \leq \widetilde{O}(\sqrt{st}) \min_{\mathsf{rank}(X)\leq k} \|Y_c X Y_r - B\|_p.$$

Now, we have from subadditivity of norm,

$$\left\|Y_c \widehat{X} Y_r - A\right\|_p - \|S^\dagger N_3 T^\dagger\|_p \leq \left\|Y_c \widehat{X} Y_r - B\right\|_p.$$

Combining this with equation (11), we have

$$\begin{aligned}
\left\|Y_c \widehat{X} Y_r - B\right\|_p &\leq \widetilde{O}(C_\phi C_\psi \sqrt{\psi\phi st}) \left\|U^* V^* - A)\right\|_p + \sqrt{st}\left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\
&\quad + 2\sqrt{st}\left\|S^\dagger N_3 T^\dagger\right\|_p + O(C_\phi C_\psi \sqrt{\psi\phi st}) \left\|\Phi^\dagger N_1\right\|_p && (13)
\end{aligned}$$

Note that $Y_c \widehat{X} Y_r$ is the output of the algorithm. Therefore, all that remain is to bound each of the above additive term. The following claim does this.

Claim 20. *With probability at least* $24/25$,

$$\begin{aligned}
\left\|\Phi^\dagger N_1\right\|_p &\leq \widetilde{O}(C_\phi d\phi/\varepsilon), \\
\left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p &\leq \widetilde{O}(C_\psi kn/\varepsilon), \\
\left\|S^\dagger N_3 T^\dagger\right\|_p &\leq \widetilde{O}(C_s C_t st/\varepsilon).
\end{aligned}$$

Proof. Now $N_2(\widehat{V}\Psi)^\dagger \widehat{V}\Psi = \widehat{N}_2$. Using no dilation property of Φ, we have $\left\| N_2(\widehat{V}\Psi)^\dagger \widehat{V} \right\|_p \leq O(C_\psi) \left\| \widehat{N}_2 \right\|_p$. This can be bound using the standard tail inequality for Laplace mechanism, i.e, with probability at least $99/100$, $\left\| \widehat{N}_1 \right\|_p = \widetilde{O}(kn)$. Similarly, $\Phi\Phi^\dagger N_1 = \widehat{N}_1$ and $SS^\dagger N_3 T^\dagger T = N_3$. Using dilation and contraction properties of Φ, Ψ, S, and T completes the proof of claim. □

Using Claim 20 in equation (13) completes the proof of Theorem 10. □

B.2 Proof of Theorem 15

Restatement of Theorem 15. Algorithm ROBUST-PCA, (see Algorithm 2), is (ε, δ)-differentially private. Further, given a matrix $A \in \mathbb{R}^{n \times d}$ with $\mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p$, it runs in time $\mathsf{poly}(k, n, d)$, space $\widetilde{O}(k(n+d))$, and outputs a rank k orthonormal projection matrix Π such that, with probability $9/10$ over the random coin tosses of the algorithm,

$$\|A - A\Pi\|_p \leq O((k \log k \log(1/\delta))^{2(2-p)/p} \log n \log^3 d)\mathsf{OPT}_k(A) + \widetilde{O}(k^2 d \log n/\varepsilon).$$

Proof of Theorem 15. We start by giving the privacy proof. Let A and A' be neighboring matrices, i.e., $\|A - A'\|_1 \leq 1$. We argue the privacy result for $p = 1$ (the case for $p \in (1,2)$ follows from invoking Holder's inequality. The private matrix is used to generate two sketches: Y_c, Y_r. Since Φ, Ψ, and T are sampled from distribution of random matrices that preserves the ℓ_p-norm, we have that $\|\Phi(A - A')\|_1 \leq C_\phi \|A - A'\|_1 = C_\phi$ with probabilty at least $1-\delta$. The privacy proof now follows from Laplace mechanism. Note that for $p \in (1,2)$, we have $\|\Phi(A - A')\|_p \leq C_\phi \|(A - A')\|_p \leq C_\phi \|(A - A')\|_1$.

We now move to prove the utility guarantee. For the ease of presentation, we just present the case for $p = 1$. The case for $p \in (1,2)$ follows similarly.

Let us define $\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_k \Sigma_r^\dagger U_r^T$. Further, let

$$U^*, V^* := \underset{\substack{U \in \mathbb{R}^{n \times k} \\ V \in \mathbb{R}^{k \times d}}}{\operatorname{argmin}} \|UV - A)\|_p$$

Two fundamental techniques. The first fundamental technique is to use the fact that solving generalized linear regression problem in the projected space gives an approximate solution to the original generalized regression problem. Then we use the reduction from low-rank approximation to a generalized linear regression problem.

Let $B = A^T + \Phi^\dagger N_1 T^\dagger$, then $\Phi B T = Z$. We now define the following optimization problems:

$$\begin{aligned}
\widetilde{V} &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|\Phi(U^* V - B)\|_p, \\
\widehat{V} &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|\Phi(U^* V - B)\|_F, \\
V' &:= \underset{V \in \mathbb{R}^{k \times d}}{\operatorname{argmin}} \|U^* V - B)\|_p
\end{aligned}$$

Then using Lemma 8 and the fact that $\|U^* V' - B)\|_p \leq \|U^* V - B)\|_p$ for all V (and in particular, V^*), we have

$$\begin{aligned}
\left\| U^* \widetilde{V} - B) \right\|_p &\leq O(\log d) \|U^* V' - B)\|_p && \text{(Lemma 8)} \\
&\leq O(\log d) \left\| U^* V^* - A^T) \right\|_p + O(\log d) \left\| \Phi^\dagger N_1 T^\dagger \right\|_p.
\end{aligned}$$

Since $\widehat{V}_{:i} = (\Phi U^*)^\dagger \Phi B_{:i} = \min_x \|\Phi(U^* x - B_{:i})\|_F$, using Holder's inequality, we have

$$\begin{aligned}
\left\|(U^* \widehat{V} - B)\right\|_p &= \sum_{i=1}^{d} \left\|(U^* \widehat{V}'_{:i} - B_{:i})\right\|_p \\
&\leq \sqrt{\phi} \sum_{i=1}^{d} \left\|(U^* \widetilde{V}_{:i} - B)\right\|_p && \text{(Lemma 18)} \\
&= \sqrt{\phi} \left\|U^* \widetilde{V} - B)\right\|_p . && (14)
\end{aligned}$$

In other words,

$$\left\|(U^* \widehat{V} - B)\right\|_p \leq O(\sqrt{\phi} \log d) \left\|U^* V^* - A^T)\right\|_p + O(\sqrt{\phi} \log d) \left\|\Phi^\dagger N_1 T^\dagger\right\|_p .$$

Moreover,

$$\left\|(U^* \widehat{V} - B)\right\|_p \geq \left\|U^* \widehat{V} - A^T)\right\|_p + \left\|\Phi^\dagger N_1 T^\dagger\right\|_p .$$

Combining the last two inequalities gives us

$$\left\|(U^* \widehat{V} - A^T)\right\|_p \leq O(\sqrt{\phi} \log d) \left\|U^* V^* - A^T)\right\|_p + O(\sqrt{\phi} \log d) \left\|\Phi^\dagger N_1 T^\dagger\right\|_p . \qquad (15)$$

Now define the following optimization problems:

$$\begin{aligned}
\widetilde{U} &:= \operatorname*{argmin}_{U \in \mathbb{R}^{n \times k}} \left\|(U \widehat{V} - A^T) \Psi\right\|_p , \\
\widehat{U} &:= \operatorname*{argmin}_{U \in \mathbb{R}^{n \times k}} \left\|(U \widehat{V} - A^T) \Psi\right\|_F , \\
U' &:= \operatorname*{argmin}_{U \in \mathbb{R}^{n \times k}} \left\|U \widehat{V} - A^T)\right\|_p
\end{aligned}$$

with solutions $\widetilde{U}, \widehat{U}$, and U'. Then using Lemma 8 and the fact that $\left\|U' \widehat{V} - A^T)\right\|_p \leq \left\|U \widehat{V} - A^T)\right\|_p$ for all U (and in particular, U^*), we have

$$\begin{aligned}
\left\|\widetilde{U} \widehat{V} - A^T)\right\|_p &\leq O(\log n) \left\|U' \widehat{V} - A^T)\right\|_p && \text{(Lemma 8)} \\
&\leq O(\log n) \left\|(U^* \widehat{V} - A^T)\right\|_p . && \text{(minimality)} \qquad (16)
\end{aligned}$$

We know that $\widehat{U}_{i:} = A^T_{:i} \Psi (\widehat{V} \Psi)^\dagger = \min_x \left\|(x \widehat{V} - A^T_{i:}) \Psi\right\|_F$. Using Holder's inequality and equation (16), we have

$$\begin{aligned}
\left\|(\widehat{U} \widehat{V} - A^T)\right\|_p &= \sum_{i=1}^{n} \left\|(\widehat{U}_{i:} \widehat{V} - A^T_{i:})\right\|_p \\
&\leq \sqrt{\psi} \sum_{i=1}^{d} \left\|(\widetilde{U}_{i:} \widehat{V} - A^T_{i:})\right\|_p && \text{(Lemma 18)} \\
&= \sqrt{\psi} \left\|\widetilde{U} \widehat{V} - A^T)\right\|_p \\
&\leq O(\sqrt{\psi} \log n) \left\|(U^* \widehat{V} - A^T)\right\|_p , && (17)
\end{aligned}$$

where the last inequality follows from equation (16). Substituting the value of $\widehat{U} = A^T \Psi (\widehat{V} \Psi)^\dagger$,

$$\begin{aligned}
\left\|A^T \Psi (\widehat{V} \Psi)^\dagger \widehat{V} - A^T\right\|_p &\leq O(\log d \log n \sqrt{\phi \psi}) \left\|U^* V^* - A^T)\right\|_p \\
&\quad + O(\sqrt{\phi \psi} \log d \log n) \left\|\Phi^\dagger N_1 T^\dagger\right\|_p
\end{aligned}$$

Recall that $Y_c = A^T\Psi + N_2$ by construction in the algorithm. Using subadditivity of norms and substituting $\widehat{V} = (\Phi U^*)^\dagger \Phi A^T$, we have

$$\begin{aligned}\left\|Y_c(\widehat{V}\Psi)^\dagger \widehat{V} - A^T\right\|_p &= \left\|Y_c(\widehat{V}\Psi)^\dagger (\Phi U^*)^\dagger \Phi B - A^T\right\|_p \\ &\leq \left\|A^T\Psi(\widehat{V}\Psi)^\dagger \widehat{V} - A^T\right\|_p + \left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\ &\leq O(\sqrt{\phi\psi}\log d \log n)\left\|U^*V^* - A^T)\right\|_p + \left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\ &\quad + O(\sqrt{\phi\psi}\log d \log n)\left\|\Phi^\dagger N_1 T^\dagger\right\|_p \end{aligned} \tag{18}$$

Now,

$$\left\|Y_c(\widehat{V}\Psi)^\dagger \widehat{V} - B\right\|_p \leq \left\|Y_c(\widehat{V}\Psi)^\dagger \widehat{V} - A^T\right\|_p + \left\|\Phi^\dagger N_1 T^\dagger\right\|_p$$

Combining the above two inequalities, we get

$$\begin{aligned}\left\|Y_c(\widehat{V}\Psi)^\dagger \widehat{V} - B\right\|_p &\leq O(\sqrt{\phi\psi}\log d \log n)\left\|U^*V^* - A^T)\right\|_p + \left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\ &\quad + O(\sqrt{\phi\psi}\log d \log n)\left\|\Phi^\dagger N_1 T^\dagger\right\|_p\end{aligned}$$

Further, since U^* has rank at most k, we have that $\widehat{V} = (\Phi U^*)^\dagger \Phi B$ has rank at most k. This implies that

$$\begin{aligned}\min_{\mathsf{rank}(X)\leq k}\|Y_c X B - B\|_p &\leq \left\|Y_c(\widehat{V}\Psi)^\dagger (\Phi U^*)^\dagger \Phi B - B\right\|_p \\ &\leq O(\sqrt{\phi\psi}\log d \log n)\left\|U^*V^* - A^T)\right\|_p + \left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\ &\quad + O(\sqrt{\phi\psi}\log d \log n)\left\|\Phi^\dagger N_1 T^\dagger\right\|_p \end{aligned} \tag{19}$$

Third fundamental technique. The last fundamental technique that we use is that an approximate solution of low-rank problem in the projected space also gives an approximate solution of the original low-rank problem. Let $R = PU\Sigma(\Phi PU\Sigma)^\dagger \Phi$, where $\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_s \Sigma_r^\dagger U_r^T$. Let $\bar{X} = \operatorname{argmin}_X \left\|\Phi(PY_c\widehat{X}X\Phi B - B)T\right\|_1$. We have the following:

$$\begin{aligned}\|\Phi(RB - B)T\|_1 &\leq \sqrt{st}\left\|\Phi(PU\Sigma(\Phi PU\Sigma)^\dagger \Phi B - B)T\right\|_F && \text{(definition of } P\text{)} \\ &= \sqrt{st}\min_X \|\Phi(PU\Sigma X\Phi B - B)T\|_F && \text{(by definition of normal form)} \\ &\leq \sqrt{st}\left\|\Phi(PU\Sigma\bar{X}\Phi B - B)T\right\|_F \\ &\leq \sqrt{st}\left\|\Phi(PU\Sigma\bar{X}\Phi B - B)T\right\|_1 \\ &= \sqrt{st}\min_X \|\Phi(PU\Sigma X\Phi B - B)T\|_1 . && \text{(definition of } \bar{X}\text{)}\end{aligned}$$

This implies that $(\Phi U\Sigma P)^\dagger$ is the approximate solution of $\min_X \|\Phi(PU\Sigma X\Phi B - B)T\|_1$. Using Lemma 9, we have

$$\left\|PU\Sigma(\Phi PY_c\widehat{X})^\dagger \Phi B - B\right\|_p \leq \sqrt{st}\min_X \|PU\Sigma X\Phi B - B\|_p$$

Let $\bar{\Phi} = \begin{pmatrix} \Phi_k \\ 0 \end{pmatrix}$ and Φ' be the matrix such that $\Phi'_{i:} = \bar{\Phi}_{\pi(i):}$. Then we have the following set of inequalities

$$\begin{aligned}
\min_X \|PU\Sigma X\Phi B - B\|_p &\leq \left\|\frac{\phi}{k}\Phi^\dagger (U\Sigma)^\dagger (U\Sigma)\Phi Y_c(\widehat{V}\Psi)^\dagger \Phi B - B\right\|_1 && \text{(by minimality)} \\
&\leq O(\log d)\left\|\frac{\phi}{k}\Phi\Phi^\dagger (U\Sigma)^\dagger (U\Sigma)\Phi Y_c(\widehat{X}\Psi)^\dagger \Phi B - \Phi B\right\|_1 && \text{(no-dilation property)} \\
&\leq O(\log d)\left\|\frac{\phi}{k}(U\Sigma)^\dagger (U\Sigma)\Phi Y_c(\widehat{X}\Psi)^\dagger \Phi B - \Phi B\right\|_1 && (\Phi\Phi^\dagger = \mathbb{I}) \\
&= O(\log d)\left\|\frac{\phi}{k}\Phi' Y_c(\widehat{V}\Psi)^\dagger \Phi B - \Phi B\right\|_1 && \text{(definition)} \\
&\leq O(\log^2 d)\left\|Y_c(\widehat{X}\Psi)^\dagger \Phi B - B\right\|_1 && \text{(Lemma 19).}
\end{aligned}$$

Combining this with equation (19) and using the value of Π gives

$$\begin{aligned}
\left\|\Pi A^T - A^T\right\|_p \leq O(\log^3 d\log n\sqrt{st\phi\psi})\left\|U^*V^* - A^T)\right\|_p + O(\sqrt{st}\log^2 d)\left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \\
+ O(\sqrt{st}\log^2 d)\left\|Y_c\widehat{X}(\Phi Y_c\widehat{X})^\dagger N_1 T^\dagger\right\|_p + O(\sqrt{st\phi\psi}\log^3 d\log n)\left\|\Phi^\dagger N_1 T^\dagger\right\|_p
\end{aligned} \tag{20}$$

All that remain is to bound each of the above additive term. The following claim does this.

Claim 21. *With probability at least* $97/100$*,*

$$\begin{aligned}
\left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p &\leq \widetilde{O}(kd\log n/\varepsilon), \\
\left\|\Phi^\dagger N_1 T^\dagger\right\|_p &\leq \widetilde{O}(C_s C_t st/\varepsilon), \\
\left\|Y_c\widehat{X}(\Phi Y_c\widehat{X})^\dagger N_1 T^\dagger\right\|_p &\leq \widetilde{O}(C_t st/\varepsilon).
\end{aligned}$$

Proof. Now $N_2(\widehat{V}\Psi)^\dagger \widehat{V}\Psi = \widehat{N}_2$. Using no dilation property of Ψ, we have $\left\|N_2(\widehat{V}\Psi)^\dagger \widehat{V}\right\|_p \leq \log n\left\|\widehat{N}_2\right\|_p$ This can be bound using the standard tail inequality for Laplace mechanism, i.e, with probability at least $99/100$, $\left\|\widehat{N}_2\right\|_p = \widetilde{O}(kd)$. Similarly, $\Phi Y_c\widehat{X}(\Phi Y_c\widehat{X})^\dagger N_1 T^\dagger T = \widehat{N}_1$ and $\Phi\Phi^\dagger N_1 T^\dagger T = N_1$. Using dilation and contraction properties of Φ, Ψ, and T completes the proof of claim. □

We finish the proof by proving that the projection matrix is an orthonormal projection matrix with high probability.

Claim 22. $PU\Sigma(\Phi PU\Sigma)^\dagger\Phi$ *is an orthonormal projection matrix with probability* $99/100$.

Proof. Since Φ is a Cauchy matrix with i.i.d. entries, Φ is a full row matrix with probability $99/100$. Therefore, it follows from the definition of P that

$$\begin{aligned}
\Pi &= PU\Sigma(\Phi PU\Sigma)^\dagger\Phi \\
&= \Phi^\dagger(U\Sigma)^\dagger U\Sigma(\Phi\Phi^\dagger(U\Sigma)^\dagger U\Sigma)^\dagger\Phi \\
&= \Phi^\dagger(U\Sigma)^\dagger U\Sigma((U\Sigma)^\dagger U\Sigma)^\dagger\Phi \\
&= \Phi^\dagger(U\Sigma)^\dagger U\Sigma(\Phi^\dagger(U\Sigma)^\dagger U\Sigma)^\dagger
\end{aligned}$$

with probability $99/100$. This completes the proof. □

The next proposition follows from the definition of $\widehat{X}$ and the fact that N, N_1, and N_2 are i.i.d. Laplace matrix.

Proposition 23. *$Y_c\widehat{X}$ has rank-k with probability at least $1-\delta$, where the probability is over the randomness of the algorithm.*

Using Claim 21 in equation (20) completes the proof of Theorem 15. □

C Local Learning

Local differential privacy, a stronger variant of privacy, has gained a lot of attention recently. For e.g., it is the privacy guarantee employed by Apple in their new iOS [1] and has been used by Google for various data analysis [15]. In the local privacy model, there is no central database of private data. Instead, each individual has its own data element (a database of size one), and sends a report based on its own datum in a differentially private manner. The local model allows individuals to retain control of their data since privacy guarantees are enforced directly by their devices. However, it entails a different set of algorithmic techniques from the central model. In principle, one could also use cryptographic techniques to simulate central model algorithms in a local model, but such algorithms currently impose bandwidth and liveness constraints that make them impractical for large deployments.

Formally, we consider the database $X = [x_1, \cdots, x_n]^\top$ as a collection of n elements (rows) from some domain $\mathcal{X} \subseteq \mathbb{R}^d$, with each row held by a different individual. The i^{th} individual has access to ε_i-*local randomizer*, $R_i : \mathcal{X} \to W$ which is an ε_i-differentially private algorithm that takes as input a database of size $n = 1$. We assume that the algorithms may interact with the database only through local randomizers. We can then define local differential privacy as follows [13]. An algorithm is ε-locally differentially private if it accesses the database X via the local randomizers, $R_1(x_1), \cdots, R_n(x_n)$, where R_i is an ε_i-local randomizer, and $\max\{\varepsilon_1, \cdots, \varepsilon_n\} \leq \varepsilon$.

We note that what we have defined above is a non-interactive local differential privacy algorithm where an individual only sends a single message to the server. Another well studied variant is that of interactive local differential privacy where the server sends several query messages, each to a subset of users. Each such message, together with responses from users, counts as a *round* of interaction. In the end, the server aggregates and summarizes the messages it received from every user (over possibly multiple rounds), and uses it to answer queries about the data. It was argued in [27] that from an implementation point of view, it is more desirable to have as few rounds of interactions as possible because interaction introduces latency, synchronization, and bandwidth issues. In fact, existing large-scale deployments [1, 15] are limited to one that are noninteractive. Therefore, we limit our study to what is possible in the noninteractive variant of local differential privacy. We study robust principal component analysis in local model of differential privacy. We show that with high probability, we have that $\|A\Pi - A\|_p \leq \ell \cdot \mathsf{OPT}_k(A) + \widetilde{O}(\varepsilon^{-1}knd)$.

Our result is applicable in the setting when $\|A\|_p \gg O(nd)$. We note that, in practice, robust LRA is used on corrupted data matrix with a reasonable fraction of entries corrupted by large values. There are other scenarios, like network analysis, where private matrices have large entries. In such scenarios, typically $\|A\|_p \gg O(nd)$, and outputting an all zero matrix would incur an error far greater than what we incurred. If we wish to output a rank-k matrix with provable guarantees, the naive algorithm that works as follows: every user add Laplace vector to their data and send the report to the server, and the server runs a non-private algorithm leads to worse additive error. This is because the low-rank approximation is now done on $A + N$ for $N \sim \mathsf{Lap}(0, 1/\varepsilon)^{n \times d}$. We next show that we can convert ROBUST-PCA to the model of local differential privacy. See Figure 3 for details. Our algorithm is non-interactive; therefore, we can use the generic transformation of [6] to get an ε-local differentially private algorithm.

Theorem 24. *Algorithm* LOCAL-ROBUST-PCA *(see Figure 3) is an ε-local differentially private algorithm. Furthermore, given a matrix $A \in \mathbb{R}^{n \times d}$ with $\mathsf{OPT}_k(A) := \min_{\mathsf{rank}(X) \leq k} \|A - X\|_p$,* LOCAL-ROBUST-PCA *runs in time $\mathsf{poly}(k, n, d)$, space $\widetilde{O}(k(n+d))$, and outputs a rank k projection matrix Π such that, with probability $9/10$ over the randomoziation of the algorithm,*

$$\|A - A\Pi\|_p \leq O(\log n \log^3 d \, (k \log k \log(1/\delta))^{2(2-p)/p}) \mathsf{OPT}_k(A) + \widetilde{O}(k^2 nd/\varepsilon).$$

Algorithm 3 LOCAL-ROBUST-PCA

Input: Every user $i \in [n]$ having access to a row A_i, target rank k

Output: Rank-k projection matrix $\Pi \in \mathbb{R}^{d \times d}$

Initialization. Sample $\Phi \in \mathbb{R}^{\phi \times d}, \Psi \in \mathbb{R}^{n \times \psi}, S \in \mathbb{R}^{s \times d}$, and $T \in \mathbb{R}^{n \times t}$ with every entry sampled iid from $\mathcal{D}_p$. All these matrices are publicly available.

user side computation: every user i, **do**

  Sample private noise. Sample $N_{1,i} \sim \mathsf{Lap}(0, C_\phi C_t)^{\phi \times t}, N_{0,i} \sim \mathsf{Lap}(0, C_\psi)^{d \times \psi}, N_{2,i} \sim \mathsf{Lap}(0, C_s C_t)^{s \times t}$ using its private coin.

  Construct. $A^{(i)} \in \mathbb{R}^{d \times n}$ with all zero entries except the column i has entry A_i.

  Compute. $Y_{r,i} = \Phi A^{(i)} T + N_{1,i}, Y_{c,i} = A^{(i)} \Psi + N_{0,i}$ and $Z_i = Y_{r,i}$, where Φ, T and Ψ are sketching matrices with every entries sampled i.i.d. from a p-stable distribution.

  Send. $(Y_{r,i}, Y_{c,i}, Z_i)$ to the server.

end user side computation:

server's computation: get $\{Y_{r,i}, Y_{c,i}, Z_i\}_{i=1}^n$, **do**

  Compute. $Y_c = \sum Y_{c,i}, Y_r = \sum Y_{r,i}$, and $Z = \sum Z_i$.

  Compute. $\mathsf{SVD}(\Phi Y_c) = [U_c, \Sigma_c, V_c]$. and $\mathsf{SVD}(Y_r) = [U_r, \Sigma_r, V_r]$.

  Set. $\widehat{X} = V_c \Sigma_c^\dagger [U_c^T Z V_r^T]_k \Sigma_r^\dagger U_r^T$, where $[B]_k = \mathrm{argmin}_{r(X) \leq k} \|B - X\|_F$.

  Pick: a permutation matrix $Q \in R^{\phi \times \phi}$.

  Compute: the full SVD of $Y_c \widehat{X}$, $[U', \Sigma', V']$. Set $U = U'Q$, $\Sigma = \Sigma' Q$, and $P = \Phi^\dagger (U\Sigma)^\dagger$.

  Output: $\Pi = PU\Sigma(\Phi PU\Sigma)^\dagger \Phi$.

end server's computation:

Note that the naive algorithm that works as follows: every user add Laplace vector to their data and send the report to the server, and the server runs a non-private algorithm leads to a slight worse additive error. This is because the low-rank approximation is now done on $A + N$ for $N \sim \mathsf{Lap}(0, 1/\varepsilon)^{n \times d}$. Song et al. [29] would then imply that the additive error would be about $O(C_\phi C_\psi \sqrt{st\psi\phi} nd)$.

Proof of Theorem 24. Using the same arithmetic as in the proof of Theorem 15, the error incurred would be

$$\|A - A\Pi\|_p \leq O(C_\phi C_\psi \sqrt{st\phi\psi}) \mathsf{OPT}_{k,p}(A) + c_2 \left(2n \left\| \widetilde{N}_1 \right\|_p + 2n \left\| \widetilde{N}_2 \right\|_p \right),$$

where $\Pi := PU\Sigma(\Phi PU\Sigma)^\dagger \Phi$, $\mathsf{OPT}_{k,p}(A) := \|U^* V^* - A)\|_p$, N_1 is an n times a $d \times \psi$ random Laplace matrix, $\widetilde{N}_1$ and $\widetilde{N}_2$ are as formed in the proof of Theorem 15. Using the same calculation completes the proof of Theorem 24. □

D A Closer Look on Current Techniques

We first give the argument we made in the main text for the accuracy guarantees by using the ζ-net mechanism of Blum et al. [4]. To apply Blum et al. [4] in our setting, we need to compute the number of k-tuples of unit vectors in $\mathbb{R}^d$ and $\mathbb{R}^n$. The size of ζ-net of unit vectors in $\mathbb{R}^d$ (for row space) is $p = \zeta^{1-d}$. Hence, number of k-tuples of unit vectors is $\binom{p}{k}$. Similarly, for column space, it is $\binom{\zeta^{1-n}}{k}$. This gives us the error claimed earlier in the introduction.

There are two main approaches for efficient private algorithms – output perturbation and input perturbation. In output perturbation, we first compute the output (e.g. rank-k approximation of a given matrix) non-privately and then add appropriately scaled noise to preserve privacy. In input perturbation, we add noise to the private matrix and then compute the output on the noisy matrix. Both these approaches require adding noise to every entry of the given input matrix or to every entry of the non-private output matrix. Consequently, both of these methods would incur an additive error of $O(nd)$.

Alternatively, one could consider iterative approaches, such as noisy Krylov subspace iteration [19], for finding low-rank matrix approximation with respect to spectral norm. However, it is not immediately clear how to adapt such an algorithm for ℓ_p low-rank approximation. The methods used in known results for differentially private low-rank approximation with respect to entrywise ℓ_2-norm, say [18, 14, 32], also has some hurdles. The main problem here is that the given objective is not rotationally invariant. If we just use the output produced in any of the results for Frobenius norm and then use, say Holder's inequality, then the accuracy would depreciate proportional to $(nd)^{1/p-1/2}$.

One may then argue that we can solve robust low-rank approximation for constant dimension by using *exponential mechanism* [26]. For using exponential mechanism, we need to find a suitable *scoring function*, which is not clear in the case of entrywise ℓ_p-norm. Even if we are able to find a scoring function analogous to one used in [21], it is not clear whether we can iterate it for k rounds to get all the top-k subspace. More precisely, it is not clear whether a result analogous to the Deflation lemma of [21] holds in the case of entrywise ℓ_p approximation.