A Framework for Private Matrix Analysis in Sliding Window Model

Jalaj Upadhyay [*1] **Sarvagya Upadhyay** [*2]

Abstract

We perform a rigorous study of private matrix analysis when only the last W updates to matrices are considered useful for analysis. We show the existing framework in the non-private setting is not robust to noise required for privacy. We then propose a framework robust to noise and use it to give first efficient $o(W)$ space differentially private algorithms for spectral approximation, principal component analysis (PCA), multi-response linear regression, sparse PCA, and non-negative PCA. Prior to our work, no such result was known for sparse and non-negative differentially private PCA even in the static data setting. We also give a lower bound to demonstrate the cost of privacy.

1. Introduction

Matrix analysis manifests itself in many walks of life such as financial transactions, recommendation system, social networks, machine learning, and learning kernels. In the recent past, there has been a paradigm shift in matrix analysis in the era of big data. Two aspects that have become increasingly important are (i) protecting sensitive information and (ii) the increasing frequency with which data is being continuously updated. An example that illustrates the importance of these two aspects arises in several investment strategies in a financial firm. The strategies rely on matrix analysis (such as principal component analysis) of financial data that get continuously updated. Most of these strategies make use of "recent data" as opposed to the entire history. This heuristic is rooted in the empirical observation that recent data are better predictors of the future behavior of assets than older data (Moore et al., 2013; Tsay, 2005), a theme also found in many other applications of matrix analysis (Campos et al., 2014; Quadrana et al., 2018).

Moreover, the strategies are sensitive and have to be kept private. It is well documented that performing statistical analysis, including matrix analysis, accurately can leak private information (Narayanan & Shmatikov, 2006). As a result, privacy preserving algorithms for matrix analysis with robust privacy guarantees such as *differential privacy* are known (Amin et al., 2019; Blum et al., 2005; Dwork et al., 2014; Kapralov & Talwar, 2013; McSherry & Mironov, 2009; Hardt & Price, 2014; Hardt & Roth, 2012; Upadhyay, 2018)). However, these algorithms are not amenable to the scenario where a collection of the most recent updates on data is pertinent for analysis. In contrast, the current practical deployment of private algorithms (Erlingsson et al., 2014; Thakurta et al., 2017) favors using only recent data for a variety of reasons.

In view of this, we focus on a rigorous and comprehensive study of privacy-preserving matrix analysis in the *sliding window model of privacy* (Bolot et al., 2013; Chan et al., 2012; Upadhyay, 2019). The model is parameterized by the window size W, and assumes that the data arrive in the form of (possibly infinite) stream over time. An analyst is required to perform the analysis only on the W most recent streams of data (usually referred to as a *sliding window*) using $o(W)$ space. On the other hand, privacy is guaranteed for the entire historical data, i.e., even if the data is not in the current window, its privacy should not be compromised.

We give $o(W)$ space differentially private algorithms for several matrix analysis problems in the sliding window model (see, Table 1). Here and henceforth, $o(W)$ will ignore other factors such as matrix dimensions and privacy parameters.

A brief overview of our main contributions are as follows (and annotate each of the points below with the corresponding appendix in the supplementary material).

[*]Equal contribution [1]Apple, USA (work done when the author was between jobs). [2]Fujitsu Research of America, USA. Correspondence to: Jalaj Upadhyay <jalaj@apple.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

	Privacy	Additive error	Space required	Reference
η-spectral approximation	(ϵ, δ)-DP	$O\left(\frac{r^2 \log^2(1/\delta)}{\epsilon^2}\right) \mathbb{1}_d$	$O\left(\frac{r^2 d}{\eta} \log W\right)$	Theorem 13
Principal component analysis (PCA)	(ϵ, δ)-DP	$O\left(\frac{\sqrt{kd}\log(1/\delta)}{\epsilon}\right)$	$O\left(\frac{dk^2}{\eta^3} \log W\right)$	Theorem 16
Sparse and Non-negative PCA	(ϵ, δ)-DP	$O\left(\frac{\sqrt{kd}\log(1/\delta)}{\epsilon}\right)$	$O\left(\frac{dk^2}{\eta^3} \log W\right)$	Theorem 17
Multiple linear regression	(ϵ, δ)-DP	$O\left(d\left(d + \frac{\log(1/\delta)}{\epsilon}\right)\right)$	$O\left(\frac{d^3}{\eta} \log W\right)$	Theorem 18
Directional variance query	(ϵ, δ)-DP	$O\left(d\left(d + \frac{\log(1/\delta)}{\epsilon}\right)\right)$	$O\left(\frac{d^3}{\eta} \log W\right)$	Theorem 11

Table 1. Results presented in this paper (W: window size, k : target rank, d : dimension of streamed row, privacy parameters (ϵ, δ), $\mathbb{1}_d$ is a $d \times d$ identity matrix, r: rank of streamed matrix).

1. **(Limitations of known framework and algorithm).** We show that existing framework of *spectral histogram* used in the non-private setting (Braverman et al., 2020) is too stringent for privacy and algorithms in that framework are not robust to perturbation required for privacy. We show rigorously that the strict constraint imposed by spectral histogram only permits sub-optimal accurate private algorithms (Appendix B). That is, adding appropriately scaled noise to the algorithm of (Braverman et al., 2020) does not suffice. This warrants a robust framework for private matrix analysis.

2. **(New framework and data structure).** We introduce a relaxation of spectral histogram property on a set of positive semidefinite (PSD) matrices that is more robust to noise and call it *approximate spectral histogram property*. We also design an update time efficient data structure that maintains the approximate spectral histogram property on a set of PSD matrices while preserving differential privacy (Appendix C).

3. **(Optimal algorithms for matrix analysis).** We use approximate spectral histogram property to efficiently compute private spectral approximation. Using this, we solve several matrix analysis problems in the sliding window model while preserving privacy and optimal accuracy in Appendix D: (i) principal component analysis (PCA); (ii) directional variance queries; and (iii) multi-response linear regression. We also give algorithm for private *constrained PCA* (Cohen et al., 2015). This generalizes many variants of PCA studied in statistical machine learning such as sparse PCA and non-negative PCA.

4. **(Limitation of private sliding window algorithms).** Finally, to complete the picture, we exhibit limitations of private matrix analysis by giving a lower bound on differentially private algorithm for low-rank approximation in the sliding window model (Appendix E).

There is a known separation between what is achievable with privacy and without privacy for real-valued functions in the sliding window model (Upadhyay, 2019). Our work can be seen as extending this study to matrix-valued functions in a unified manner. Conceptually, approximate spectral histogram property can be viewed as a generalization of *subspace embedding property* (Sarlós, 2006). This allows us to use approximate spectral histogram property in the sliding window model in the same way as subspace embedding is employed in the streaming model of privacy (Upadhyay, 2018). Given the wide application of subspace embedding in streaming algorithms, we believe that the notion of approximate spectral histogram will have further applications in the sliding window model of privacy.

A natural question one may ask is why we need to introduce approximate spectral histogram property in the sliding window model of privacy. We end this section with a discussion on this (more details in Section 2). Let us consider the spectral approximation of matrices. There is one private algorithm (Blocki et al., 2012) which relies on subspace embedding. They explicitly compute the singular value decomposition of the matrix making it suitable only for static data matrix. Furthermore, we cannot just take off-the-shelf algorithm and add noise matrix to preserve privacy as well as guarantee non-trivial utility and efficiency. To begin with, standard noise mechanisms would result in a matrix that is not positive semidefinite. This is, for example, the mechanism in Dwork et al. (2014). If we instead use the projection trick of Arora & Upadhyay (2019) on top of Dwork et al. (2014), it would incur noise that scales with the dimension and have an inefficient update time. Moreover, the existing randomized space-efficient algorithm of Braverman et al. (2020) performs sampling proportional to its *leverage score*. As a result, the effect of a single row in the matrix formed by this sampling procedure can be arbitrarily large, and consequently, the sensitivity is high[1].

[1]There are counterexamples where leverage score for a row can change arbitrarily depending on whether it is in the span of the current

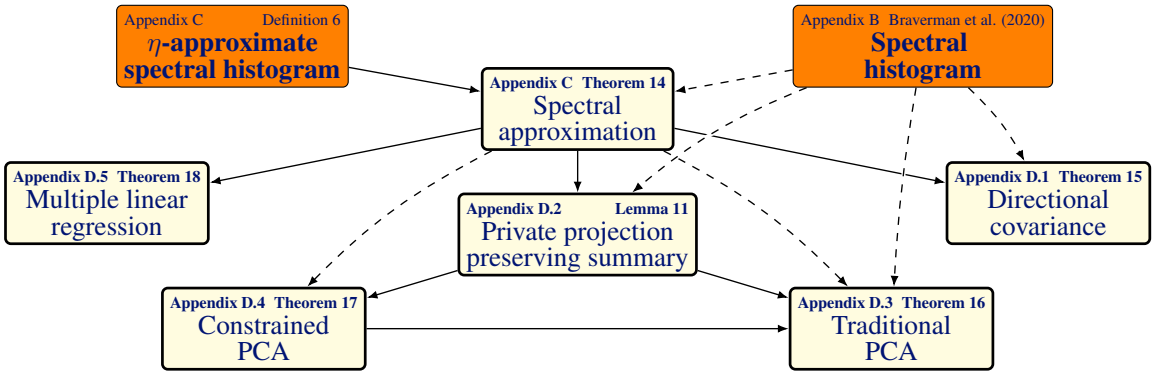

Figure 1. Dependency graph of various results (bold lines shows optimal results and dashed lines shows suboptimal results, orange boxes are datastructure). For example, a datastructure satisfying η-approximate spectral histogram implies an algorithm for spectral approximation, and so on. All our algorithms extend to the streaming model as well by setting $W = T$.

Notations. For a natural number n, the notation $[n]$ denotes the set $\{1, \ldots, n\}$. The Euclidean norm of a vector $v \in \mathbb{R}^d$ is denoted by $||v||_2$. For a rank-r matrix $A \in \mathbb{R}^{n\times d}$, we let the tuple $(s_1(A), s_2(A), \ldots, s_r(A))$ denote the non-zero singular values of A arranged in decreasing order, $A^\top$ to denote transpose of A, and $\|A\|_F$ to denote its Frobenius norm. The i-th row vector and the j-th column vector of a matrix A are denoted by $A[i:]$ and $A[:j]$, respectively. We use $\|A[:j]\|_2$ and $\|A[i:]\|_2$ to denote their Euclidean norms. We use $\mathbb{1}_d$ to denote identity matrix of dimension d. If all the eigenvalues of a symmetrix matrix $S \in \mathbb{R}^{d\times d}$ are non-negative, then the matrix is known as *positive semidefinite* (PSD for short) and is denoted by $S \succeq 0$. For symmetric matrices $A, B \in \mathbb{R}^{d\times d}$, the notations $A \preceq B$ implies that $B - A$ is PSD and $A \not\preceq B$ implies that $B - A$ is not a PSD. For any $T, d > 0$, we use $\mathsf{N}_{T,d}$ to denote the following set of $T \times d$ matrices:

$$\mathsf{N}_{T,d} := \big\{B \in \mathbb{R}^{T\times d} : \exists i \in [T] \text{ such that } \|B[i:]\|_2 \le 1 \text{ and } \|B[j:]\|_2 = 0 \text{ for all } j \neq i\big\}.$$

A comprehensive overview of preliminaries and notations is presented in Appendix A.

1.1. Sliding window, privacy, and matrix analysis

We start by defining some additional notations pertinent to studying matrix analysis in the sliding window model. The matrix formed by d-dimensional row vectors streamed between time stamps t_1 and t_2 is denoted $A_{[t_1,t_2]}$. We define $A_W(T) := A_{[T-W+1,t]}$ for any current timestamp T where W is used to denote the window size and $A_T := A_{[0,T]}$. The matrix A_T can be obtained by setting $W = T$ and gives us the insertion only streaming model (Muthukrishnan, 2005). The matrix $A_W(T) \in \mathbb{R}^{W\times d}$ is formed incrementally through a stream of d-dimensional row vectors $\{a_i : T-W+1 \le i \le T\}$ as follows:

$$A_W(T) := \begin{pmatrix} a_{T-W+1} \\ \vdots \\ a_{T-1} \\ a_T \end{pmatrix} \in \mathbb{R}^{W\times d}. \tag{1}$$

At start, the matrix $A_W(0)$ is an all zero matrix (with $a_i = 0^d$ if $i \le 0$). At any time T, we are interested in performing various analysis on the matrix $A_W(T)$. Our results are independent of the current time stamp T, and we will slightly abuse the notation by letting $A_W = A_W(T)$ as the matrix formed by rows streamed in the last W updates.

We now formalize the privacy model. We adhere to the neighboring relation employed in existing literature studying matrix analysis in static setting (Blocki et al., 2012; Hardt & Roth, 2012; Dwork et al., 2014; Sheffet, 2019) and streaming setting (Upadhyay, 2018).

matrix or not (see for example, (Arora & Upadhyay, 2019) in the context of graph sparsification). In fact, it is not clear if we can even use the exponential mechanism because for most natural score functions, one can construct counterexamples where the sensitivity of the score function is also large.

In privacy literature, there are two well-studied levels of granularity when the data arrives in an online manner (Bolot et al., 2013; Chan et al., 2011; 2012; Dwork et al., 2010; Dwork & Roth, 2014; Upadhyay, 2018; 2019): (i) *user-level privacy*, where two streams are neighboring if they differ in a single user's data; and (ii) *event-level privacy*, where two streams are neighboring if they differ in one-time epoch. We follow previous works on private analysis in the sliding window model (Bolot et al., 2013; Chan et al., 2012; Huang et al., 2021) and consider event-level privacy. We say that two streams are *neighboring* if, at any time $T > 0$, they form matrices A_T and A'_T such that $A_T - A'_T \in \mathsf{N}_{T,d}$. We now define the privacy notion that extends the privacy notion of Bolot et al. (2013); Chan et al. (2012); Huang et al. (2021); Upadhyay et al. (2021) and Upadhyay (2019) to general matrices.

Definition 1 (Differential privacy under sliding window model)**.** *For $\epsilon \geq 0, \delta \in [0,1]$, we say a randomized algorithm $\mathcal{M}$ with range Y is (ϵ, δ)-differentially private in the sliding window model if for all $T > 0$, for every two matrices A_T and A'_T formed by neighboring streams, and for all $\mathsf{S} \subseteq \mathsf{Y}$, $\Pr[\mathcal{M}(A_T) \in \mathsf{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(A'_T) \in \mathsf{S}] + \delta$, where the probability is over the private coin tosses of M.*

Note that the privacy guarantee is for the entire stream, i.e., even if the data has expired, its privacy is not lost. However, accuracy is required only for the last W updates. This is in accordance with previous problem formulation (Bolot et al., 2013; Chan et al., 2012; Upadhyay, 2019).

The central algebraic concept underlying all analysis of interest in this paper is the spectrum of a matrix (see Figure 1). Therefore, we focus on privately computing (η, ζ)-spectral approximation, i.e., given parameters $\eta, \zeta \geq 0$ and a matrix $A_W \in \mathbb{R}^{W \times d}$, find a matrix $C \in \mathbb{R}^{d \times d}$, such that

$$(1-\eta)A_W^\top A_W - \nu \mathbb{1}_d \preceq C \preceq (1+\eta)A_W^\top A_W + \nu \mathbb{1}_d.$$

Here the parameter $\nu \geq 0$ is the cost of privacy in the terms of distortion in the spectrum. Our goal is to keep η as small as possible so that they are useful in subsequent tasks, like PCA, multiple regression, etc. We show the following:

Theorem 1 (Informal version of Theorem 14)**.** *Let $A_W \in \mathbb{R}^{W \times d}$ be a rank-r matrix formed by the current window. Then for $\nu = \xi \log \xi$ where $\xi = O\left(\frac{r \log^2(W/\delta)}{\epsilon^2 \eta}\right)$, there is an efficient (ϵ, δ)-differentially private algorithm under sliding window model that uses $O\left(\frac{dr^2}{\eta^2} \log W\right)$ space and outputs a matrix C at the end of the stream such that*

$$(1-\eta)A_W^\top A_W - \nu \mathbb{1}_d \preceq C \preceq (1+\eta)A_W^\top A_W + \nu \mathbb{1}_d.$$

A special case when the matrix is the edge-adjacency matrices was considered by Upadhyay et al. (2021). In the static setting, using the result of Sarlós (2006) and Blocki et al. (2012), we get an $O(d^2)$ space private algorithm which guarantees (η, ν, ν)-spectral approximation for $\nu = O\left(\frac{d \log(1/\delta)}{\epsilon^2 \eta}\right)$. Non-privately, there is an algorithm in the sliding window model that uses $O\left(\frac{rd}{\eta} \log W\right)$ space if the matrix has a bounded condition number (Braverman et al., 2020). In many practical scenarios, the rank is constant. In this scenario, the privacy overhead is only a constant factor. Our algorithm is also flexible in the sense that we can also guarantee that the output is a PSD matrix.

Before giving a technical overview of our private algorithm, we begin by arguing why the existing private algorithms in the static setting fail in the sliding window model. Blocki et al. (2012) gave the first privacy preserving approximation of matrices. Their approach is to first compute the singular value decomposition of the given matrix $A = USV^\top$, and then output $C_{\mathsf{BBDS}} = \widehat{A}^\top \Phi^\top \Phi \widehat{A}$, where $\widehat{A} := U\sqrt{S^2 + \sigma^2 \mathbb{1}_d} V^\top$ for a perturbation parameter σ chosen appropriately, and Φ is a random Gaussian matrix. Since, the algorithm requires computing the SVD, one cannot extend this approach in the sliding window model. Another approach, due to Dwork et al. (2014), computes $C_{\mathsf{DTTZ}} = A^\top A + N$, where N is a symmetric Gaussian matrix with appropriate variance. In this case, we cannot revert the effect of the rows outside of the window.

Private principal component analysis has been extensively studied (Amin et al., 2019; Blum et al., 2005; Dwork et al., 2014; Hardt & Roth, 2012; Upadhyay, 2018; Dwork et al., 2014; Hardt & Price, 2014; Kapralov & Talwar, 2013; Singhal & Steinke, 2021), and matching lower and upper bounds are known on achievable accuracy in the static setting. With the exception of Arora et al. (2018); Upadhyay (2018), these algorithms perform at least two passes over the matrix. Dwork et al. (2014) gave an online algorithm for PCA using regularized follow-the-leader framework; however, online model is very different from the sliding window model[2]. Finally, the algorithm of Arora et al. (2018) and Upadhyay (2018) does not

[2]The online learning model is a game between a decision-maker and adversary. The decision-maker makes decisions iteratively. After committing to a decision, it suffers a (possible adversarially) loss. The goal is to minimize the total loss in retrospect to the best decision the decision-maker should have taken.

extend to the sliding window model because we cannot revert the effect of the rows that are outside of the current window.

2. Main lemma and overview of techniques

One-shot vs Continual release. In this section, we focus only on the case when the output is produced just once at the end of the stream for the ease of presentation. Such algorithms are known as *one-shot algorithm* in the literature of differential privacy and used as a building block for algorithms that continually release statistics. We cover the case of *continual release* (Dwork et al., 2010) in Appendix F, where we propose two data structures that allow continual release depending on whether space is more important or accuracy. The first approach uses the binary tree method introduced by Bentley & Saxe (1980) and used in Dwork et al. (2010) and Chan et al. (2011). However, unlike them, we build the binary tree only over the current window. This uses space linear in W but incur error that only grows polylogarithmically. In the second approach, we reduce the space requirement to be sublinear in W at the cost of increasing the error. We subdivide each window in to $\sqrt{W}$ sub-windows, each of size $\sqrt{W}$. We then run an instance of our algorithm for each of these sub-windows.

We now focus our attention to design a one-shot algorithm. Algorithmically, our approach is closest to Smith et al. (2020). They present a one-shot space-optimal algorithm for *distinct element count* in a data-stream by showing that the celebrated Flajolet-Martin sketch initiated with some random "phantom" elements (guaranteed to be not in the data set) is differentially private. Similar approaches has been used for computing low-rank approximation of a matrix formed in a streaming manner (Upadhyay, 2018).

One-shot algorithm. Our one-shot algorithms (on which the continual release algorithms is based) can be seen as a generalization of the technique of Smith et al. (2020) from real-valued functions to matrix-valued functions. We inject an appropriate random matrix to the data stream. However, this would only allows us to perform the analysis on the entire data stream and not just on the current window. That is, we need to resolve the following two related questions:

1. **(Question 1).** How to account only for only the relevant part of the streamed data, i.e., one in the window?
2. **(Question 2).** What distribution of random matrices is to be used to inject phantom random matrices?

One naive candidate algorithm, $\mathsf{A}_{\mathsf{priv}}$, for private spectral approximation is as follows: store a set of $w = \min\{W, T\}$ positive semidefinite matrices at any time T, where the i-th matrix in this set is a sanitized version of the matrix formed by the last i updates. In this case, question 1 is answered by just removing any matrix that is out of the window, and question 2 is answered by using Wishart matrix of appropriate scale. However, $\mathsf{A}_{\mathsf{priv}}$ requires prohibitively large $O(Wd^2)$ space.

To answer question 1, while using significantly less space (as in Smith et al. (2020)) requires a conceptual contribution. To this end, we introduce *η-approximate spectral histogram property* for a set of PSD matrices and timestamps. We will occasionally refer to such a set as a data structure.

η-approximate spectral histogram property. For a matrix $S \succeq 0$, denote by $\widetilde{S}$ a matrix such that

$$\left(1 - \frac{\eta}{4}\right)\widetilde{S} \preceq S \preceq \left(1 + \frac{\eta}{4}\right)\widetilde{S}.$$

Let the current window of our matrix analysis be from timestamp $T - W + 1$ to T and $S(i)$ be the covariance matrix of the matrix formed by rows streamed between timestamps t_i and T. In other words,

$$S(i) = A_{[t_i,T]}^{\top} A_{[t_i,T]}.$$

Let $\mathfrak{D}$ be a data structure comprised of a collection of ℓ timestamps and PSD matrices $\{(t_1, \widetilde{S}(1)), \ldots, (t_\ell, \widetilde{S}(\ell))\}$ for some $\ell \in \mathbb{N}$. For all $i \in [\ell]$, $\widetilde{S}_i$ is an $(\eta/4, 0)$-spectral approximation of the matrix S_i. Roughly speaking, such a data structure $\mathfrak{D}$ satisfies *η-approximate spectral histogram property* if following two listed properties are satisfied..

1. The timestamps satisfy the following two requirements:

$$t_1 < \cdots < t_\ell = T \quad \text{and} \quad t_1 \leq T - W + 1 \leq t_2.$$

Algorithm 1 PHASE 2($\mathfrak{M}'_{T+1} = \{\bar{A}(1), \cdots, \bar{A}(\ell+1)\}$)

1: **If** $t_2 < T - W + 1$, set $\bar{A}(i) = \bar{A}(i+1), t_i = t_{i+1}$ for all $i \in [\ell - 1]$. Set $\ell = \ell - 1$
2: **Define** $\bar{S}(i) = \bar{A}(i)^\top \bar{A}(i)$ for all $1 \le i \le \ell + 1$.
3: **For** $i = 1, \cdots \ell - 2$
4: Find $j = \max\{u > i : (1 - \frac{\eta}{2})\bar{S}(i) \preceq \bar{S}(u)\}$.
5: Set $\mathfrak{M}'_{T+1} \leftarrow \mathfrak{M}'_{T+1} \backslash \{\bar{A}(i+1), \cdots, \bar{A}(j-1)\}$.
6: Reorder the indices of remaining matrices.
7: Update $\ell := \ell + i - j + 1$.
8: **Output** $\mathfrak{M}_{T+1} := \mathfrak{M}'_{T+1}$.

2. These two sets of matrices $\{S(i)\}_{i\in[\ell]}$ and $\{\widetilde{S}(i)\}_{i\in[\ell]}$ satisfy the following three conditions:

$$\begin{aligned} &\forall i \in [\ell - 1], S(i+1) \preceq S(i); \\ &\forall i \in [\ell - 1], (1-\eta)S(i) \preceq S(i+1); \text{ and} \\ &\forall i \in [\ell - 2], \left(1 - \frac{\eta}{2}\right)\widetilde{S}(i) \not\preceq \widetilde{S}(i+2). \end{aligned} \tag{2}$$

When it is clear from context, we call a set of matrices $\left\{\widetilde{S}_1, \cdots, \widetilde{S}_\ell\right\}$ as the one satisfying the η-approximate spectral histogram property. In contrast, spectral histogram in Braverman et al. (2020) requires $\widetilde{S}(i) = S(i)$ and uses the condition $(1-\eta)S(i) \not\preceq S(i+2)$ instead of $\left(1 - \frac{\eta}{2}\right)\widetilde{S}(i) \not\preceq \widetilde{S}(i+2)$.

The properties in Equation 2 are required to get the desirable space bound. Likewise, the second condition in Equation 2 and the restriction $t_1 \le T - W + 1 \le t_2$ are required to demonstrate the accuracy guarantee (see proof sketch of Theorem 1). Before proving the accuracy guarantee, we answer how to maintain such a set of matrices. For brevity, we introduce the following notation for matrices in the rest of this section: for any time T, we write $A(i)$ to denote the i-th matrix stored in the current data-structure.

Lemma 1. *Let $\mathfrak{M}_T := \{A(1), \ldots, A(\ell)\}$ be the set of matrices such that $\{A(1)^\top A(1), \ldots, A(\ell)^\top A(\ell)\}$ satisfies η-approximate spectral histogram property at time T. Then there is an efficient algorithm,* UPDATE*, that takes $\mathfrak{M}_T$ and a row $a_{T+1} \in \mathbb{R}^d$ as input and outputs a set of matrices $\mathfrak{M}_{T+1} = \{B(1), \ldots, B(m)\}$, such that $\{B(1)^\top B(1), \ldots, B(m)^\top B(m)\}$ satisfy the η-approximate spectral histogram property for some $m \le \ell + 1$.*

When a new row $a_{T+1} \in \mathbb{R}^d$ is streamed, an algorithm is invoked that updates the data structure. It works in two phases: *privatization* and *maintenance*. Privatization is accomplished by (i) adding a linear sketch of a_{T+1} to all ℓ matrices in $\mathfrak{M}_T$ to obtain a new set $\mathfrak{M}'_T$, (ii) privatizing a_{T+1} to get a matrix $A(\ell+1)$, and (iii) defining $\mathfrak{M}'_{T+1} := \mathfrak{M}'_T \cup A(\ell+1)$. For privacy (or answering Question 2), adding a noise matrix that is a PSD matrix would incur additive error linear in dimension. Moreover, it will not maintain structural properties of matrices such as low-rank, which are one of the reasons why matrix analysis have such a wide array of applications. Therefore, *just adding appropriately scaled noise is not an option* (see Appendix B for details). As it turns out, a variant of Johnson-Lindenstrauss mechanism (Blocki et al., 2012) used in Upadhyay (2018) suffices for our purpose.

Now the set $\{\bar{A}^\top \bar{A} : \bar{A} \in \mathfrak{M}'_{T+1}\}$ may not satisfy η-approximate spectral histogram property. The *maintenance* phase (high-level description of this phase is provided in Algorithm 1) ensures that the final set of matrices satisfies η-approximate spectral histogram property. In this phase, we greedily remove matrices if they do not satisfy any of the desired properties of η-approximate spectral histogram property (Algorithm 7 in supplementary material). The computationally expensive part in Algorithm 1 is Step 3. For this step, we can use known PSD testing algorithms (Bakshi et al., 2020).

Our greedy approach is reminiscent of the *potential barrier method* to compute spectral sparsification of a $W \times d$ matrix (Batson et al., 2012). In the potential barrier method, we remove a large subset of rank-one matrices and show that only storing $\Theta\left(d\eta^{-2}\right)$ rank-one matrices suffices for $(\eta, 0)$-spectral sparsification. This approach does not extend over to streaming matrices. In fact, two key technical features distinguish our method from theirs. In their setting, all PSD matrices are rank-one matrices corresponding to a row of the matrix; whereas we have W positive semidefinite matrices that may have different ranks (not necessarily rank-one). The second crucial point is that we aim to significantly reduce the number of matrices stored for our application. This makes maintaining our data structure much more complicated than the potential barrier method.

The proof of Lemma 1 is subtle. While it is tempting to use the analysis of the deterministic algorithm by Braverman et al. (2020) in our setting, their analysis is highly susceptible to noise. Their proof relies heavily on the fact that for all $i \in [\ell], \widetilde{S}(i) = S(i)$, i.e., matrices are exact covariance matrices corresponding to the streamed rows. In contrast, our analysis deals with the spectral approximation of the streamed matrix along with the perturbation required to preserve privacy. That is, each of the matrices $\widetilde{S}(1), \cdots, \widetilde{S}(\ell)$ is an approximation of the input matrix and has both multiplicative approximation as well as additive term. We give an arguably simpler analysis than Braverman et al. (2020) and crucially use the slack of $\left(1 - \frac{\eta}{2}\right)$ factor in the third condition of approximation spectral histogram property (Equation 2). A detail proof of Lemma 1 is presented in Appendix C.

Spectral approximation. Now that we have an algorithm to maintain η-approximate spectral histogram property, we show how to use it to compute an (η, ν)-spectral approximation of A_W. Let $\widetilde{S}(1), \ldots, \widetilde{S}(\ell)$ be the set of matrices satisfying η-approximate spectral histogram property. The algorithm outputs $S = \widetilde{S}(1) - \sigma^2 \mathbb{1}_d$, where σ^2 is the perturbation posit in the mechanism of Upadhyay (2018). Using the first condition of Equation 2 and that $t_1 < T - W + 1 < t_2$, $S(2) \preceq A_W^\top A_W \preceq S(1)$. The second condition of Equation 2 implies that $(1-\eta)S(1) \preceq S(2)$. Since $\widetilde{S}(1)$ and $\widetilde{S}(2)$ are a $(\eta/4, 0)$-spectral approximation of $S(1)$ and $S(2)$, respectively, this allows us to prove that $\widetilde{S}(1)$ is a spectral approximation of A_W.

Proof sketch of Theorem 1. For space bound, properties in equation (2) imply that there is at least one singular value that decreases by a factor of $(1 - \frac{\eta}{2})$ in every successive timestamp. We will see later that our privacy mechanism ensures that the spectrum of any matrix $\widetilde{S}_i$ is lower bounded by a constant. Since updates have bounded entries, there can be at most $\ell := O\left(r \log_{1-\frac{\eta}{2}}(W)\right) = O\left(\frac{r}{\eta}\log(W)\right)$ matrices satisfying η-approximate spectral histogram. For privacy, we use the Johnson-Lindenstrauss mechanism (Blocki et al., 2012). In this mechanism, we first perturb the matrix to raise its singular value and then multiply it with a random Gaussian matrix. The choice of perturbation used here is the one described in Upadhyay (2018) because it can account for the streamed data.

Now we give a proof sketch of the accuracy guarantee. At any time T, let $A(i)$ be the matrix formed between the time interval $[t_i, T]$. Let $\{\widetilde{A}(1), \cdots, \widetilde{A}(\ell)\}$ be the set of matrices obtained by applying Johnson Lindenstrauss mechanism on the streamed matrices $\{A(1), \cdots, A(\ell)\}$ and $\{\widehat{A}(1), \cdots, \widehat{A}(\ell)\}$ be the set of perturbed matrices before applying the Johnson-Lindenstrauss transform. Fix the following notations for covariance matrices:

$$C(j) := A(j)^\top A(j), \quad \widetilde{S}(j) := \widetilde{A}(j)^\top \widetilde{A}(j)$$
$$S(j) := \widehat{A}(j)^\top \widehat{A}(j) = C(j) + \sigma^2 \mathbb{1}_d.$$

The perturbation parameter σ is as chosen in Sheffet (2019). Since $t_1 \leq T - W + 1 \leq t_2$, we have $C(2) \preceq A_W^\top A_W \preceq C(1)$. By design of our algorithm and the second property of η-approximate spectral histogram property, we have $(1-\eta)S(1) \preceq S(2)$. We pick the dimension of the Johnson-Lindenstrauss transform so that $\widetilde{S}(j)$ is an $(\eta/4, 0)$-spectral approximation of $S(j)$ for all $j \in [\ell]$ using Sarlós (2006)'s result. Therefore, for $i \in \{1, 2\}$,

$$\left(1 - \frac{\eta}{4}\right) S(i) \preceq \widetilde{S}(i) \preceq \left(1 + \frac{\eta}{4}\right) S(i).$$

This implies that $\left(1 - \frac{\eta}{4}\right)(C(1) + \sigma^2 \mathbb{1}_d) \preceq \widetilde{S}(1)$. Since adding positive semidefinite matrices preserves the Loewner ordering and $A_W^\top A_W \preceq C(1)$, we get the following:

$$\left(1 - \frac{\eta}{4}\right)(A_W^\top A_W + \sigma^2 \mathbb{1}_d) \preceq \left(1 - \frac{\eta}{4}\right)(C(1) + \sigma^2 \mathbb{1}_d)$$
$$\preceq \widetilde{S}(1).$$

Similarly, for the upper bound, we have from the definition,

$$\widetilde{S}(1) \preceq \left(1 + \frac{\eta}{4}\right) S(1) \preceq \frac{\left(1 + \frac{\eta}{4}\right)}{(1 - \eta)} S(2)$$
$$= \frac{\left(1 + \frac{\eta}{4}\right)}{(1 - \eta)} (C(2) + \sigma^2 \mathbb{1}_d).$$

Using the fact that $C(2) \preceq A_W^\top A_W$, scaling η and setting the value of σ completes the proof. □

	Additive Error	Multiplicative	Space Required	Comments
Hardt & Roth (2012)	$\widetilde{O}(k\sqrt{d}/\epsilon^2)$	$O(1)$	$O(d^2)$	rank-$2k$, static data
Dwork et al. (2014)	$\widetilde{O}\left(\epsilon^{-1}k\sqrt{d}\right)$	–	$\widetilde{O}\left(d^2\right)$	Static data
Upadhyay (2018)	$\widetilde{O}\left(\epsilon^{-1}\sqrt{kd}\right)$	$(1+\eta)$	$\widetilde{O}\left(\eta^{-1}dk\right)$	Streaming data
Lower Bound	$\Omega\left(\sqrt{kd}\right)$	$(1+\eta)$	$\Omega\left(\eta^{-1}dk\log W\right)$	Sliding window
This Paper	$\widetilde{O}\left(\epsilon^{-1}\sqrt{kd}\right)$	$(1+\eta)$	$\widetilde{O}\left(\eta^{-3}dk^2\log W\right)$	Sliding window

Table 2. Comparison of $\left(\epsilon, \Theta\left(d^{-\log d}\right)\right)$-Differentially private PCA results (our results are in red).

3. Applications

We present three main applications of η-approximate spectral histogram property for matrix analysis.

Applications I: Principal component analysis. Principal component analysis is an extensively used subroutine in many applications like clustering (Cohen et al., 2015), recommendation systems (Drineas et al., 2002), and learning distributions (Achlioptas & McSherry, 2005). In these applications, given a matrix $A \in \mathbb{R}^{n\times d}$ and a target rank k, the goal is to output a rank-k orthonormal projection matrix $P \in \mathbb{R}^{d\times d}$ such that

$$\|A - AP\|_F \leq (1+\eta) \min_{\mathsf{rank}(X)\leq k} \|A - X\|_F + \zeta.$$

The goal here is to minimize ζ for a given k, d, and privacy parameters ϵ and δ. In many applications, instead of optimizing over all rank-k projection matrices, we are required to optimize over a smaller set of projection matrices, such as one with only non-negative entries. In particular, let Π be any set of rank-k projection matrices (not necessarily set of all rank-k projection matrices). Then the *constrained principal component analysis* is to find $P^* = \operatorname{argmin}_{P\in\Pi} \|A - AP\|_F^2$.

A naive application of approximate spectral histogram property to solve PCA leads to an additive error that depends linearly on the rank of the streamed matrix. To solve these problems with optimal accuracy, we introduce an intermediate problem that we call *private projection preserving summary* (Definition 8). This problem can be seen as a private analogue of PCP sketches (Cohen et al., 2015). Solving this problem ensures that the additive error scales with the parameter k and not with the rank of the matrix.

To remove the dependency on the rank of the streamed matrix, we consider the first k/η spectrum of the streamed matrix and show that it suffices for our purpose. That is, let $\widetilde{A}_1, \cdots, \widetilde{A}_\ell$ be matrices such that their covariance matrices $\widetilde{S}_1, \cdots, \widetilde{S}_\ell$ satisfy η-approximate spectral histogram property. We show that random projections of $\widetilde{A}_1, \cdots, \widetilde{A}_\ell$ to a k/η dimensional linear subspace suffice. Let $\pi_{k/\eta}(\widetilde{A}_1), \cdots, \pi_{k/\eta}(\widetilde{A}_\ell)$ be the projected matrices. We show that the set of covariance matrices corresponding to $\pi_{k/\eta}(\widetilde{A}_1), \cdots, \pi_{k/\eta}(\widetilde{A}_\ell)$ satisfy the approximate spectral histogram property. Using this, we show that the first matrix in this set, $\widetilde{A} := \pi_{k/\eta}(\widetilde{A}_1)$, is a *private projection preserving summary* for A_W with a small additive error. For this, we make use of the private version of one of the characterizations of projection preserving summary due to (Cohen et al., 2015). This characterization is crucial as it defines the multiplicative approximation as well as additive error.

Lemma 2 (Informal version of Lemma 11)**.** *Let k be the desired rank, η be the approximation parameter, and (ϵ, δ) be the privacy parameter. Let Π be the set of all rank-k projection matrices. Then there is an efficient (ϵ, δ)-differentially private algorithm under sliding window model that for a given matrix A_W formed by the current window, outputs a matrix $\widetilde{A}$ such that for any $P \in \Pi$,*

$$\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F \leq (1+\eta)\|A_W(\mathbb{1}_d - P)\|_F + O\left(\frac{1}{\alpha\epsilon}\sqrt{kd\log(d)\log^2\left(\frac{W}{\delta}\right)}\right).$$

This lemma allows us to show the first result to solve constrained PCA.

Theorem 2 (Informal version of Theorem 17)**.** *Let A_W be the matrix formed by last W updates and Π be a given set of rank-k projection matrices. Then there is an (ϵ, δ)-differentially private algorithm that outputs a matrix $\widetilde{A}$ at the end of the stream, such that if $\|\widetilde{A}(\mathbb{1}_d - P)\|_F \leq \gamma \cdot \min_{X\in\Pi} \|\widetilde{A}(\mathbb{1}_d - X)\|_F$ for some $\gamma > 0$ and $P \in \Pi$, then*

$$\|A_W(\mathbb{1}_d - P)\|_F \leq (1+\eta)\,\gamma \cdot \min_{X\in\Pi} \|A_W(\mathbb{1}_d - X)\|_F + O\left(\frac{1}{\alpha\epsilon}\sqrt{kd\log(d)\log^2\left(\frac{W}{\delta}\right)}\right).$$

The matrix P in the above result can be computed by running any known non-private algorithm on $\widetilde{A}$. There are existing results for structured projection matrices, such as Asteris et al. (2014); Yuan & Zhang (2013). In particular, if Π is a set of sparse or non-negative projection matrices, then Theorem 17 gives a way to solve these problems privately. Moreover, Theorem 17 also implies a private algorithm for PCA by using any algorithm for PCA that achieves $\gamma = 1$ (Eckart & Young, 1936).

For traditional PCA, Corollary 4.5 in Hardt & Roth (2012) gives a rank-p projection matrix for $p > 2k$ with a large constant multiplicative approximation and $O(\frac{k\sqrt{d}}{\epsilon^2})$ additive error. The underlying reason for this large constant factor is because they use Markov inequality after using the expectation bound of Halko et al. (2011). We avoid this by appealing to the results that use the concentration property of random Gaussian matrices (Kane & Nelson, 2014).

We finally remark that we do not violate the lower bound of Dwork et al. (2014). Their lower bound holds when there is no multiplicative approximation. They show similar upper bound as Theorem 17 when matrices has a singular value gap of $\Omega(\sqrt{d})$. In contrast to their $O(d^2)$ space algorithm, we make use of $O\left(\frac{dk^2}{\eta^3}\log W\right)$ space in the sliding window setting, which is an improvement whenever $k\log(W) = o(\eta^3 d)$. We also note that Dwork et al. (2014) studied PCA in the *online learning model* (Hazan, 2019), which is incomparable to the sliding window model.

Application II: Multi-response linear regression. Another application of Theorem 1 is solving multi-response linear regression (also known as *generalized linear regression*) in the sliding window model. It is a widely studied generalization of the standard ℓ_2-regression (Woodruff, 2014). Formally, given two matrices $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times p}$ as input, the multi-response linear regression is defined as the minimization problem, $\min_{X\in\mathbb{R}^{d\times p}} \|AX - B\|_F^2$.

Theorem 3 (Informal version of Theorem 18)**.** *Let $A_W \in \mathbb{R}^{W\times d}$ and $B \in \mathbb{R}^{W\times p}$ be the matrix streamed during the window of size W formed as defined in equation (1), ϵ, δ, η be as before. Then there exists an $\tau = \left(d + \frac{14}{\epsilon^2}\log(\frac{4}{\delta})\right)\log^2(W)$ and (ϵ,δ)-differentially private algorithm in the sliding window model that output a matrix $\widetilde{X} \in \mathbb{R}^{d\times p}$ such that*

$$\left\|A_W\widetilde{X} - B_W\right\|_F^2 \leq (1+\eta)\min_{X\in\mathbb{R}^{d\times p}} \|A_W X - B\|_F^2 + O\left(\frac{(\tau+p)^2\log(\tau+p)}{\epsilon}\right).$$

This is the first result for multiple-response regression and matches the bound achieved in Sheffet (2019) when $p = 1$ even though we are in a more restrictive setting.

Application III: Directional variance queries. The directional variance queries has the following form: the analyst gives a unit-length vector $x \in \mathbb{R}^d$ and wish to know the variance of A_W along x. Theorem 1 gives an algorithm to answer directional covariance queries (and cut queries when the matrix is the edge-adjacency matrix of a graph).

Theorem 4 (Informal version of Theorem 15)**.** *Let A_W be the matrix formed by last W updates as defined in equation (1) and ϵ, δ, η be as before. Given a bound q on the number of queries that can be made, there is an efficient (ϵ,δ)-differentially private algorithm that outputs a matrix C such that for any set of q unit vector queries $x_1, \cdots, x_q \in \mathbb{R}^d$, we have for all $i \in [q]$*

$$x_i^\top A_W^\top A_W x_i - \frac{c\log q\log d}{\epsilon} \leq x_i^\top C x_i \leq (1+\eta)x_i^\top A_W^\top A_W x_i + \frac{c\log q\log d}{\epsilon}.$$

Even though we are in a more restrictive setting of sliding window, this matches the bound achieved in Blocki et al. (2012) after we apply the improvement in Sheffet (2019).

4. Concluding remarks

We believe that our approach will find applications beyond what is covered in this paper and will pave way for further research in the intersection of differential privacy and sliding window model. We focus on the model where every data in the current window is considered equally useful to explain the heuristics used in recent deployments. However, one can consider other variants of the sliding window model as far as privacy is concerned. As an example, one can consider a model where the privacy of a data decays as a monotonic function of time lapse. More so, there are more concrete questions to be asked and answered even in the model studied in this paper.

As we mentioned earlier, one can see η-approximate spectral histogram property as a generalization of subspace embedding property. We believe that any improvement in designing a more efficient data structure for maintaining a set of matrices satisfying η-approximate spectral histogram property will have a profound impact on large-scale deployment of privacy-preserving algorithms in the sliding window model. For example, we believe that space requirements can be reduced using randomization. This randomization can be either oblivious or may depend on the current set of positive semidefinite matrices. Since our set of positive semidefinite matrices are generated using a privacy mechanism, any such sampling can be viewed as post-processing and hence privacy preserving. Hence, our main conjectures are concerning the space required by any privacy-preserving algorithm. We elaborate them next.

The lower bound of $\Omega\left(d^2\right)$ space for spectral approximation is required even in the static setting. We conjecture that there should be $\frac{1}{\eta}\log W$ factor due to the sliding window requirement. This is because, if the spectrum of a matrix is polynomially bounded, then one can construct a sequence of updates that requires at least $\frac{1}{\eta}\log W$ matrices such that successive matrices are $(1-\eta)$ apart in terms of their spectrum. For an upper bound, we believe randomization can help reduce a factor of d. This is achieved in the non-private setting using online row sampling. It was shown by Upadhyay (2018) that one can design private algorithms with space-bound comparable to a non-private algorithm in the streaming model of computation. The situation in the sliding window model is more complicated, but we believe it is possible to achieve a matching upper bound. In view of this, we conjecture the following.

Conjecture 1. *The space required for differentially private spectral approximation is* $\Theta\left(\frac{d^2}{\eta}\log W\right)$.

We believe that the bound on the additive error is optimal. A positive resolution to this conjecture would imply that the price of privacy is only in terms of the additive error.

Our second conjecture is for principal component analysis. We believe that our space-bound for principal component analysis is tight up to a factor of $\frac{k}{\eta}$. A lower bound of $\Omega(dk)$ is trivial as one requires $O(dk)$ space just to store the orthonormal matrix corresponding to the rank-k projection matrix. As before, a factor of $\frac{1}{\eta}\log W$ would be incurred due to the sliding window model. The factor of $\frac{1}{\eta}$ comes from the fact that to extract the top-k subspace, we need $\frac{k}{\eta}$ dimensional subspace.

Conjecture 2. *The space required for differentially private PCA is* $\Omega\left(\frac{dk}{\eta^2}\log W\right)$.

We believe that proving such a lower bound would require new techniques. This is because, in PCA, we only have access to an orthonormal projection matrix, while in the case of low-rank approximation, we have far more information to solve the underlying communication complexity problem.

Our work identifies another application of the Johnson-Lindenstrauss and Wishart mechanisms. Before our results, it was not even clear whether the JL mechanism can be used to compute PCA (see Section V in Blocki et al. (2012))! They consider their output matrix $\widetilde{C}$ as a "test" matrix to test if the input matrix has high directional variance along some direction $x \in \mathbb{R}^d$. However, they do not give any guarantee as to how the spectrum of C relates to that of the input covariance matrix.

5. Acknowledgement

JU's research was supported, in part, by NSF BIGDATA awards IIS-1838139 and IIS 1546482. JU would like to acknowledge Petros Drineas for useful discussion on the technique of Batson et al. (2012).

References

Achlioptas, D. and McSherry, F. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, pp. 458–469. Springer, 2005.

Amin, K., Dick, T., Kulesza, A., Munoz, A., and Vassilvitskii, S. Differentially private covariance estimation. In *Proceedings of the 32nd Advances in Neural Information Processing Systems*, pp. 14190–14199, 2019.

Arora, R. and Upadhyay, J. On differentially private graph sparsification and applications. In *Proceedings of the 32nd Advances in Neural Information Processing Systems*, pp. 13378–13389, 2019.

Arora, R., Braverman, V., and Upadhyay, J. Differentially private robust low-rank approximation. In *Advances in Neural Information Processing Systems*, pp. 4137–4145, 2018.

Asteris, M., Papailiopoulos, D., and Dimakis, A. Nonnegative sparse PCA with provable guarantees. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1728–1736, 2014.

Bakshi, A., Chepurko, N., and Jayaram, R. Testing positive semi-definiteness via random submatrices. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science*, pp. 1191–1202, 2020.

Bar-Yossef, Z. *The complexity of massive data set computations*. PhD thesis, University of California, Berkeley, 2002.

Batson, J., Spielman, D. A., and Srivastava, N. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

Bentley, J. L. and Saxe, J. B. Decomposable searching problems i. static-to-dynamic transformation. *Journal of Algorithms*, 1(4):301–358, 1980.

Blocki, J., Blum, A., Datta, A., and Sheffet, O. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *Proceeding of the 53rd Annual IEEE Sympsium on Foundations of Computer Science*, pp. 410–419, 2012.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: the SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 128–138, 2005.

Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., and Taft, N. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory*, pp. 284–295, 2013.

Braverman, V., Drineas, P., Musco, C., Musco, C., Upadhyay, J., Woodruff, D. P., and Zhou, S. Near optimal linear algebra in the online and sliding window models. In *Proceeding of the 61st Annual Symposium on Foundations of Computer Science*, pp. 517–528, 2020.

Campos, P. G., Díez, F., and Cantador, I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119, 2014.

Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14(3):26:1–26:24, 2011.

Chan, T.-H. H., Li, M., Shi, E., and Xu, W. Differentially private continual monitoring of heavy hitters from distributed streams. In *Proceedings of the 12th International Privacy Enhancing Technologies Symposium*, pp. 140–159, 2012.

Clarkson, K. L. and Woodruff, D. P. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63 (6):54, 2017.

Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM symposium on Theory of computing*, pp. 163–172, 2015.

Drineas, P., Kerenidis, I., and Raghavan, P. Competitive recommendation systems. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pp. 82–90, 2002.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 715–724, 2010.

Dwork, C., Talwar, K., Thakurta, A., and Zhang, L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pp. 11–20, 2014.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Erlingsson, Ú., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067, 2014.

Friedland, S. and Torokhti, A. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

Halko, N., Martinsson, P.-G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

Hardt, M. and Price, E. The noisy power method: A meta algorithm with applications. In *Proceedings of the 27th Advances in Neural Information Processing Systems*, pp. 2861–2869, 2014.

Hardt, M. and Roth, A. Beating randomized response on incoherent matrices. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pp. 1255–1268, 2012.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Huang, Z., Qiu, Y., Yi, K., and Cormode, G. Frequency estimation under multiparty differential privacy: One-shot and streaming. *arXiv preprint arXiv:2104.01808*, 2021.

Kane, D. M. and Nelson, J. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1):4, 2014.

Kapralov, M. and Talwar, K. On differentially private low rank approximation. In *Proceedings of the 44th Annual ACM-SIAM Symposium on Discrete algorithms*, pp. 1395–1414, 2013.

McSherry, F. and Mironov, I. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 627–636, 2009.

Miltersen, P. B., Nisan, N., Safra, S., and Wigderson, A. On data structures and asymmetric communication complexity. In *STOC*, pp. 103–111. ACM, 1995.

Moore, J. L., Chen, S., Turnbull, D., and Joachims, T. Taste over time: The temporal dynamics of user preferences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 401–406, 2013.

Muthukrishnan, S. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, 1(2), 2005.

Narayanan, A. and Shmatikov, V. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

Quadrana, M., Cremonesi, P., and Jannach, D. Sequence-aware recommender systems. *ACM Computing Surveys*, 51(4):66, 2018.

Sarlós, T. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 143–152, 2006.

Sheffet, O. Old techniques in differentially private linear regression. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pp. 789–827, 2019.

Singhal, V. and Steinke, T. Privately learning subspaces. *arXiv preprint arXiv:2106.00001*, 2021.

Smith, A., Song, S., and Thakurta, A. The Flajolet-Martin sketch itself preserves differential privacy: Private counting with minimal space. In *Proceedings of the 33rd Advances in Neural Information Processing Systems*, 2020.

Thakurta, A. G., Vyrros, A. H., Vaishampayan, U. S., Kapoor, G., Freudiger, J., Sridhar, V. R., and Davidson, D. Learning new words, March 14 2017. US Patent 9,594,741.

Tsay, R. *Analysis of Financial Time Series*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2005.

Upadhyay, J. The price of privacy for low-rank factorization. In *Proceedings of the 31st Advances in Neural Information Processing Systems*, pp. 4180–4191, 2018.

Upadhyay, J. Sublinear space private algorithms under the sliding window model. In *Proceedings of the 36th International Conference of Machine Learning*, pp. 6363–6372, 2019.

Upadhyay, J., Upadhyay, S., and Arora, R. Differentially private analysis on graph streams. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 1171–1179, 2021.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, 2014.

Yuan, X.-T. and Zhang, T. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14:899–925, 2013.

Zass, R. and Shashua, A. Nonnegative sparse PCA. In *Proceedings of the 19th Advances in Neural Information Processing Systems*, pp. 1561–1568, 2006.

A. Notation and Preliminaries

We use the notation $\mathbb{R}$ to denote the space of real numbers and $\mathbb{N}$ to denote the set of natural numbers. For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$.

Linear algebra. The space of n-dimensional vectors over reals is denoted $\mathbb{R}^n$. The set of non-negative vectors (also known as non-negative orthant) and the set of strictly positive vectors in $\mathbb{R}^n$ are denoted $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$, respectively. For a vector x, we let $x^\top$ denote the transpose of the vector. We reserve the letters x, y, z to denote real vectors. The entries of a vector $x \in \mathbb{R}^n$ is denoted as follows:

$$x = \big(x[1], x[2], \cdots, x[n]\big)^\top.$$

We let $\{\bar{e}_i : i \in [n]\}$ (where $[n] := \{1, 2, \cdots, n\}$) denote the set of standard basis vectors of $\mathbb{R}^n$. That is,

$$\bar{e}_i[j] = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

We let the vector of all 1's denoted by $\bar{e}$, i.e., $\bar{e} = \bar{e}_1 + \bar{e}_2 + \cdots + \bar{e}_n$. For two vectors $x, y \in \mathbb{R}^n$, their inner product is denoted $\langle x, y \rangle$. The set of real $n \times m$ matrices is denoted by $\mathbb{R}^{n \times m}$. For a real matrix A, its (i, j) entry is denoted by $A[i, j]$ and its transpose is denoted $A^\top$. The following special classes of matrices are relevant to this paper.

1. A real matrix $A \in \mathbb{R}^{n \times n}$ is *symmetric* if $A = A^\top$. The set of symmetric matrices is denoted $\mathbb{S}^n$ and forms a vector space over $\mathbb{R}$. The eigenvalues of symmetric matrices are real.
2. A symmetric matrix $A \in \mathbb{S}^n$ is *positive semidefinite* (PSD) if all of its eigenvalues are non-negative. The set of such matrices is denoted $\mathbb{S}^n_+$. The notation $A \succeq 0$ indicates that A is positive semidefinite and the notations $A \succeq B$ and $B \preceq A$ indicate that $A - B \succeq 0$ for symmetric matrices A and B. We also use the notation $A \not\succeq B$ and $B \not\preceq A$ for $A, B \in \mathbb{S}^n$ to say that $A - B \notin \mathbb{S}^n_+$.
3. A PSD matrix $A \in \mathbb{S}^n_+$ is *positive definite* if all of its eigenvalues are strictly positive. The set of such matrices is denoted $\mathbb{S}^n_{++}$. The notation $A \succ 0$ indicates that A is positive definite and the notations $A \succ B$ and $B \prec A$ indicate that $A - B \succ 0$ for symmetric matrices A and B.
4. A matrix $U \in \mathbb{R}^{n \times n}$ is *orthonormal* if $UU^\top = U^\top U = \mathbb{1}_n$, where $\mathbb{1}_n$ is the identity matrix.
5. A symmetric matrix $P \in \mathbb{S}^n$ is a rank-k *orthogonal projection matrix* if it satisfies $P^2 = P$ and it's rank is k. Such matrices have eigenvalues 0 and 1.

The eigenvalues of any symmetric matrix $A \in \mathbb{S}^n$ are denoted by $(\lambda_1(A), \ldots, \lambda_n(A))$ sorted from largest to smallest: $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$. When discussing the largest and smallest eigenvalues, we alternately use the notation $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ to denote $\lambda_1(A)$ and $\lambda_n(A)$, respectively. Similarly, the singular values of A is denoted by the tuple $(s_1(A), \ldots, s_n(A))$ sorted from largest to smallest: $s_1(A) \geq s_2(A) \geq \cdots \geq s_n(A)$. We use the notation $s_{\max}(A)$ and $s_{\min}(A)$ to denote the largest and smallest singular values of A, respectively. It is a well known fact that for any symmetric matrix A $s_{\max}(A) = \max\{|\lambda_{\max}(A)|, |\lambda_{\min}(A)|\}$ and $s_{\min}(A) = \min\{|\lambda_{\max}(A)|, |\lambda_{\min}(A)|\}$. The maximum number of non-zero singular values of $A \in \mathbb{R}^{n \times m}$ is $\min\{n, m\}$. The *spectral norm* of a matrix $A \in \mathbb{R}^{n \times m}$ is defined as

$$\|A\|_2 = \max\{\|Ax\|_2 \,:\, x \in \mathbb{R}^m,\ \|x\|_2 = 1\}.$$

The spectral norm of A is equal to the largest singular value of A. The trace norm of a rank-r symmetric matrix A is defined as the sum of its singular values. The Frobenius norm of a matrix A is defined as

$$\|A\|_F := \left(\sum_{ij} |A[i,j]|^2\right)^{1/2} = \left(\sum_{i=1}^{r} |s_i(A)|^2\right)^{1/2}.$$

This directly implies that $\|A\|_F^2 = \mathsf{Tr}\left(A^\top A\right)$ for any $n \times d$ matrix A.

We use two types of matrix decomposition. The first matrix decomposition is *spectral decomposition* (or *eigenvalue decomposition*), i.e., a symmetric matrix $A \in \mathbb{S}^n$ can be written as

$$A = U \Lambda U^\top = \sum_{i=1}^{n} \lambda_i(A) x_i x_i^\top$$

where U is an orthonormal matrix, Λ is a diagonal matrix with eigenvalues of A on its diagonal, and the set $\{x_i \in \mathbb{R}^n : i \in [n]\}$ are set of orthonormal vectors known as eigenvectors of A. We note that orthonormal matrices can also be decomposed in above form. The second matrix decomposition that is relevant to this paper is *singular value decomposition* (or SVD for short). Any real matrix $A \in \mathbb{R}^{n\times d}$ can be decomposed as follows:

$$A = USV^\top = \sum_{i=1}^{\min\{n,d\}} s_i(A) x_i y_i^\top.$$

Here, $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{d\times d}$ are orthonormal matrices, S is a diagonal matrix with diagonal entries singular values of A, and the sets $\{x_i \in \mathbb{R}^n : i \in \min\{n,d\}\}$ and $\{y_j \in \mathbb{R}^d : j \in \min\{n,d\}\}$ are orthonormal sets of vectors. Associated with any real matrix $A \in \mathbb{R}^{n\times d}$ is a matrix $A^\dagger$ called the *Moore-Penrose pseudoinverse* (or, *pseudoinverse*) and is defined as

$$A^\dagger = \sum_{i=1}^{\min\{n,d\}} \frac{1}{s_i(A)} x_i y_i^\top \quad \text{where} \quad A = \sum_{i=1}^{\min\{n,d\}} s_i(A) x_i y_i^\top.$$

We use the notation

$$[A]_k := \sum_{i=1}^{\min\{k,n,d\}} s_i(A) x_i y_i^\top$$

to denote the best rank-k approximation of matrix A under any unitary invariant norm.

We consider matrices formed by a streams of d-dimensional row vectors over $\mathbb{R}$. For a row vector $a_t \in \mathbb{R}^d$ streamed at time t, we use the notation $\bar{A}(a_t) \in \mathbb{R}^{W\times d}$ to denote an all zero matrix except row t which is a_t if $t \le W$ and row W to be a_t if $t > W$. For time epochs, t_1 and t_2, we define the matrix $A_{[t_1,t_2]} \in \mathbb{R}^{(t_2-t_1)\times W}$ to denote the matrix formed by stacking the row vectors $a_{t_1}, \cdots, a_{t_2} \in \mathbb{R}^d$ streamed from time epoch t_1 to t_2:

$$A_{[t_1,t_2]} := \begin{pmatrix} a_{t_1} & a_{t_1+1} & \dots & a_{t_2} \end{pmatrix}^\top.$$

Probability distributions. For a random variable $X \in \mathbb{R}$, we denote the mean, μ, and variance, σ^2, of X by $\mathbb{E}[X]$ and $\mathsf{Var}(X)$, respectively. We say that a random variable $X \in \mathbb{R}$ has Gaussian (or normal) distribution, denoted by $X \sim \mathcal{N}(\mu, \sigma^2)$, if its probability density function is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

We denote Gaussian distribution by $\mathcal{N}(\mu, \sigma^2)$.

Throughout this paper, we discuss and work with *random matrices*. They are simply matrices with matrix entries drawn from random variables that may or may not be independent. A special class of random matrices are Wishart matrices. They are defined as below. Let C is a $d \times d$ positive definite matrix and $m > d-1$. A $d \times d$ random symmetric positive definite matrix R is said to have a Wishart distribution $R \sim \mathsf{Wis}_d(m, C)$, if its probability density function is

$$p(W) := \frac{|R|^{\frac{m-d-1}{2}}}{2^{md}|C|^{\frac{m}{2}}\Gamma_d(\frac{m}{2})} \exp\left(\frac{1}{2}\mathsf{Tr}\left(C^{-1}R\right)\right),$$

We also consider the *constrained PCA* (Cohen et al., 2015). This is a generalization of many variants of principal component analysis such as sparse and non-negative PCA.

Definition 2 (Constrained principal component analysis (Cohen et al., 2015))**.** *Given a matrix A_W formed by a window of size W, a rank parameter k, a given set of rank-k projection matrices Π and an accuracy parameter $0 < \eta < 1$, design a differentially private algorithm under sliding window model which outputs a projection matrix $P \in \Pi$ such that*

$$\|A - PA\|_F \le (1+\eta) \min_{X\in\Pi} \|A - XP\|_F + \tau$$

with probability at least $1-\beta$. Furthermore, the algorithm should satisfies (ϵ, δ)-differential privacy. When Π is a set of all rank-k projection matrices, then we get the traditional PCA.

Definition 3. *An* embedding *for a set of points $P \subseteq \mathbb{R}^n$ with distortion α is a matrix E such that*

$$\forall x \in P, (1-\eta)\|x\|_2^2 \leq \|Ex\|_2^2 \leq (1+\eta)\|x\|_2^2.$$

A subspace embedding *is an embedding for a set K, where K is a k-dimensional linear subspace.*

A key concept in randomized numerical linear algebra and low rank approximation is that of rank-k projection cost preserving sketch. It was defined and characterized by Cohen et al. (2015).

Definition 4 (Rank-k projection-cost preserving sketch (Cohen et al., 2015))**.** *A matrix $\widetilde{A} \in \mathbb{R}^{n\times d}$ is a rank k projection-cost preserving sketch of $A \in \mathbb{R}^{n\times d}$ with error $0 \leq \eta < 1$ if, for all rank k orthogonal projection matrices $P \in \mathbb{R}^{n\times n}$,*

$$(1-\eta)\|A - PA\|_F \leq \left\|\widetilde{A} - P\widetilde{A}\right\|_F + c \leq (1+\eta)\|A - PA\|_F$$

for some fixed non-negative constant c that may depend on A and $\widetilde{A}$ but is independent of P.

Differential privacy. An often easy to handle way to define (ϵ, δ)-differential privacy is in the terms of privacy loss function. Let S be the support of the output of an algorithm M. Let $\mathcal{P}$ be the output distribution of M when its input is $A_W(t)$ and $\mathcal{Q}$ be the output distribution of M when its input is $A'_W(t)$. Then for $v \in S$, we define the privacy loss function as follows:

$$L(v) := \log\left(\frac{\mathcal{P}(v)}{\mathcal{Q}(v)}\right).$$

An algorithm M is (ϵ, δ) differentially private if $\mathsf{Pr}_{v\sim\mathcal{P}}\left[L(v; M) \leq \epsilon\right] \geq 1 - \delta$.

One concept that would be useful in our algorithm for continual release is that of partial sum.

Definition 5 (Partial sum)**.** *A P-sum is a partial sum of consecutive items,*

$$S_{[i,j]} = \sum_{\ell=i}^{j} A_i^\top A_i.$$

Fact 1. *Let $\mathfrak{M}$ be an algorithm that releases a collection of P-sums such that a single entry in the stream can appear in at most k of the P-sums. Then the sensitivity of the output is k. Further, if each of the answers are the result of at most these ℓ P-sums, then the error would scale as a factor of $k\sqrt{\ell}$.*

A.1. Prior results used in this paper

Theorem 5 (Clarkson & Woodruff (2017))**.** *Let $A \in \mathbb{R}^{n\times d}$ be a rank r matrix and $\Phi \in \mathbb{R}^{n\times \frac{4r}{\eta}}$ be random matrix with i.i.d. copies of $\mathcal{N}(0, \frac{\eta}{4r})$. Then we have*

$$\mathsf{Pr}\left[\left(1-\frac{\eta}{4}\right)A^\top A \preceq A^\top\Phi^\top\Phi A \preceq \left(1+\frac{\eta}{4}\right)A^\top A\right] \geq 1 - \frac{1}{\mathrm{poly}(d)}.$$

Lemma 3 (Sarlós (2006))**.** *Let $\mathcal{R} \in \mathbb{R}^{t\times n}$ be a distribution of random Gaussian matrix, i.e., for $R \sim \mathcal{R}$, such that $R[i,j] \sim \mathcal{N}(0, 1/t)$. Then R satisfies subspace embedding for some $t = O(\eta^{-2}\log(1/\beta))$.*

Lemma 4 (Sufficient condition for rank-k PCP (Cohen et al., 2015; Sarlós, 2006))**.** *A matrix $\widetilde{A}$ is a rank-k projection-cost preserving sketch with $c = \min\{0, \eta'_2\|A - [A]_k\|_F\}$ as long as we can write*

$$\widetilde{A}^\top\widetilde{A} - A^\top A = E_1 + E_2 + E_3 + E_4$$

with $\eta = \eta_1 + \eta_2 + \eta'_2 + \eta_3 + \eta_4$, such that

1. *E_1 is symmetric and $-\eta_1 A^\top A \preceq E_1 \preceq \eta_1 A^\top A$.*
2. *E_2 is symmetric, $\mathsf{Tr}(E_2) \leq \eta'_2\|A - [A]_k\|_F^2$, and*

$$\sum_{i=1}^{k}|\lambda_i(E_2)| \leq \eta_2\|A - [A]_k\|_F^2.$$

3. *The span of columns of E_3 is a subspace of span of columns of $A^\top A$ and*

$$\mathsf{Tr}\left(E_3^\top (A^\top A)^\dagger E_3\right) \leq \eta_3^2 \left\|A - [A]_k\right\|_F^2.$$

4. *The span of rows of E_4 is a subspace of span of rows of $A^\top A$ and*

$$\mathsf{Tr}\left(E_4^\top (A^\top A)^\dagger E_4\right) \leq \eta_4^2 \left\|A - [A]_k\right\|_F^2.$$

Proposition 6. *For $\eta \in (0,1)$, we have $(1-\eta) < \left(1-\frac{\eta}{2}\right)\left(\frac{1-\frac{\eta}{4}}{1+\frac{\eta}{4}}\right)$*

Proof. For $\eta \in (0,1)$, we have the following:

$$(1-\eta)\left(1+\frac{\eta}{4}\right) = 1 - \frac{3\eta}{4} - \frac{\eta^2}{4} \leq 1 - \frac{3\eta}{4} + \frac{\eta^2}{8} = \left(1-\frac{\eta}{2}\right)\left(1-\frac{\eta}{4}\right).$$

Since $\eta^2 > 0$, the result follows. □

Lemma 5 (Friedland & Torokhti (2007))**.** *Let R be a matrix with orthonormal rows and C have orthonormal columns. Then for a given matrix F of conforming dimensions, we have*

$$\min_{X:\mathsf{rank}(X)=k} \|CXR - F\|_F = \|C[C^\top F R^\top]_k R - F\|_F.$$

We use the following two results by Sheffet (2019), the latter first shown by (Blocki et al., 2012).

Theorem 7 (Wishart mechanism (Sheffet, 2019))**.** *Draw a sample $R \sim \mathsf{Wis}_d(\tau, \mathbb{1}_d)$, where $\tau \geq d + \frac{28\ln(4/\delta)}{\epsilon^2}$. Then for a matrix $X \in \mathbb{R}^{n\times d}$, $X^\top X + R$ is (ϵ,δ)-differentially private.*

Theorem 8 (Johnson-Lindenstrauss mechanism (Sheffet, 2019))**.** *Fix a positive integer r and $w = \frac{4\sqrt{r\log(4/\delta)}+\log(4/\delta)}{\epsilon}$. Let $A \in \mathbb{R}^{n\times d}$ such that $d < r$. Given that $s_d(A) \geq w$, then publishing RA is (ϵ,δ)-differentially private if the entries of $R \in \mathbb{R}^{r\times n}$ are i.i.d samples from $\mathcal{N}(0,1)$.*

To prove our lower bound, we give a reduction to the augmented indexing problem, AIND: In this problem, Alice is given an N-bit string x and Bob is given an index $\mathsf{ind} \in [N]$ together with $x_{\mathsf{ind}+1}, \cdots, x_N$. The goal of Bob is to output x_{ind}. The complexity for solving AIND is well known.

Theorem 9 (Miltersen et al. (1995))**.** *The minimum number of bits of communication required to solve* AIND *with probability $2/3$ in one way communication model (the messages are sent either from Alice to Bob or from Bob to Alice), is $\Omega(N)$.*

B. Limitation of spectral histogram framework in private analysis

The goal of this section is to show that the sufficient condition used in Braverman et al. (2020) that allows us to perform all the matrix analysis tasks mentioned in Section 1 allows works in the private case, but making the deterministic algorithm of Braverman et al. (2020) differentially private leads to a sub-optimal accuracy. In other words, we need a new framework and algorithm.

Definition 6 (Spectral histogram property (Braverman et al., 2020))**.** *A data structure $\mathfrak{D}$ satisfy the* spectral histogram *property if there exists an $\ell = \mathrm{poly}(n, \log W)$ such that*

1. *$\mathfrak{D}$ consists of ℓ timestamps $\mathfrak{T} := \{t_1, \cdots, t_\ell\}$ and PSD matrices $\mathfrak{M} := \{K(1), \cdots, K(\ell)\}$.*
2. *For $1 \leq i \leq \ell - 1$, at least one of the following holds:*
   (a) If $t_{i+1} = t_i + 1$, then $(1-\eta)\,K(i) \not\preceq K(i+1)$.
   (b) For all $1 \leq i \leq \ell - 2$: (i) $(1-\eta)K(i) \preceq K(i+1)$, and (ii) $(1-\eta)\,K(i) \not\preceq K(i+2)$.
3. *Let T be the current time stamp, then $t_1 \leq T - W + 1 \leq t_2$.*

Algorithm 2 UPDATE-EXACT(DS)

Require: A data structure DS a set of positive semidefinite matrices $\{K(1), \cdots, K(\ell)\}$ such that $K(1) \succeq K(2) \succeq \ldots \succeq K(\ell)$ and corresponding a set of timestamps $t_1, \cdots, t_\ell$.
Ensure: Updated set of positive semidefinite matrices $K(1), \cdots, K(\ell)$ and timestamps $t_1, \cdots, t_\ell$.
1: **for** $i = 1, \cdots \ell - 2$
2: Find $j := \max\{p : (1-\eta)K(i) \preceq K(p) \wedge (i < p \leq \ell - 1)\}$. {Find spectrally close checkpoints.}
3: **Delete** $K(i+1), \cdots, K(j-1)$. {It is important that we delete only up to index $j-1$.}
4: **Set** k=1
5: **while** $i + k \leq \ell$
6: **Update** the checkpoints as follows: $K(i+k) = K(j+k-1), t_{i+k} = t_{j+k-1}$.
7: **end**
8: Update $\ell := \ell + i - j + 1$.
9: **end**
10: **Return** $\mathsf{DS}_{\mathsf{spectral}} := \{(K(1), t_1), \ldots, (K(\ell), t_\ell)\}$.

We later show that even a relative error approximation to these covariance matrices suffices for matrix analysis and gives tighter bounds. The relative strength comes from the fact that K_i are exact covariance matrices (and not approximation). The main purpose of this section is as follows: (i) demonstrate rigorously that using spectral histogram property gives sub-optimal bounds and (ii) simplify the presentation of of our main algorithm by first presenting a special case of exact computation at every timestamps stored in the spectral histogram data structure.

We first show existence of an algorithm (UPDATE-EXACT) that on takes a set of positive semidefinite matrices not necessarily satisfying spectral histogram property as input and outputs a set of positive semidefinite matrices satisfying spectral histogram property. The algorithm UPDATE-EXACT performs a sequential check over the current set of positive semidefinite matrices and removes all the matrices that do not satisfy the spectral histogram property. Since we make no assumption on the input except that they satisfy Loewner ordering, it is possible that, in the worst case, all but one matrix can be deleted in step 2. However, as we will see later, UPDATE-EXACT will form a subroutine of our differentially private algorithms such that the input to UPDATE-EXACT will have a particular form on top of satisfying the Loewner ordering. This will help us utilize UPDATE-EXACT in a much better way. We being with showing the following for UPDATE-EXACT algorithm

Lemma 6 (Spectral histogram property)**.** *Let* $\mathsf{DS} = \{(K(1), t_1), \ldots, (K(\ell), t_\ell)\}$ *be a set consisting of timestamps and positive semidefinite matrices such that* $K(1) \succeq K(2) \succeq \ldots \succeq K(\ell) \succeq 0$. *Let* $\mathsf{DS}_{\mathsf{spectral}} \leftarrow$UPDATE-EXACT$(\mathsf{DS})$ *be the output of the algorithm defined in Algorithm 2. Then* UPDATE$(\cdot)$ *is an efficient algorithm and* $\mathsf{DS}_{\mathsf{spectral}}$ *satisfy spectral histogram property.*

Proof Sketch of Lemma 6. The proof of the above lemma can be derived from Lemma 7 which states a more general case of approximation. We give a short proof sketch below.

Consider a time epoch T and a succeeding time epoch $T' = T + 1$. Let the data structure at time T be $\mathsf{DS}_{\mathsf{priv}}(T)$ and at time T' be $\mathsf{DS}_{\mathsf{priv}}(T')$. Let t_i be a timestamp in $\mathsf{DS}_{\mathsf{priv}}(T)$ where $i < \ell$. We can have two cases: (i) There is no $1 \leq j \leq s$ such that $t'_j = t_i$ and $t'_{j+1} = t_{i+1}$, and (ii) There is a $1 \leq j \leq s$ such that $t'_j = t_i$ and $t'_{j+1} = t_{i+1}$. In both cases, spectral histogram property follows from the update rules. This is because we delete indices up to $j-1$ in Step 2 and the maximality of the index j in Step 2 of Algorithm 2. □

We next give an intuition why we need this lemma. Lemma 6 gives the guarantee that, if we are given a set of PSD matrices in Loewner ordering, then we can efficiently maintain a small set of PSD matrices that satisfy spectral histogram property. The idea of our algorithm is to ensure that UPDATE-EXACT always receives a set of PSD matrices. This is attained by our algorithm PRIV-INITIALIZE, described in Algorithm 3. We use both these subroutines in our main algorithm, SLIDING-PRIV. SLIDING-PRIV receives a stream of rows and call these two subroutines on every new update. We show that SLIDING-PRIV, described in Algorithm 4, provides the following guarantee.

Theorem 10 (Private spectral approximation under sliding window)**.** *Given the privacy parameter* ϵ*, window size* W*, approximation parameter* β*, let* $S = (a_t)_{t>0}$ *be the stream such that* $a_t \in \mathbb{R}^d$*. Further define* A_W *to be the matrix formed at time* T *by the last* W *updates. Then* SLIDING-PRIV$(S; (\epsilon, \delta); W)$*, described in Algorithm 4, uses* $O\left(\frac{n^3}{\eta} \log W\right)$ *space and*

Algorithm 3 PRIV-INITIALIZE($\mathsf{DS}_{\mathsf{priv}}; a_t; t; (\epsilon, \delta); W; r$)

Require: A new row $a_t \in \mathbb{R}^d$, a data structure $\mathsf{DS}_{\mathsf{priv}}$ storing a set of timestamps $t_1, \cdots, t_\ell$, current time t, privacy parameters (ϵ, δ), window size W, and set of matrices $\widetilde{K}(1), \cdots, \widetilde{K}(\ell+1)$.

Ensure: Updated matrices $\widetilde{K}(1), \cdots, \widetilde{K}(\ell+1)$ and timestamps $t_1, \cdots, t_\ell$.

if $t_2 < t - W + 1$
  Set $t_j = t_{j+1}, \widetilde{K}(j) := \widetilde{K}(j+1)$ for $1 \le j \le s-1$ {Delete the expired timestamp.}
end
Set $t_{\ell+1} = t, \epsilon_0 = \frac{\alpha\epsilon}{\log(W)}$. Sample $R \sim \mathsf{Wis}_d(\tau, \mathbb{1}_d)$, where $\tau = \left\lfloor d + \frac{14}{\epsilon_0^2}\log(1/\delta) \right\rfloor$.
Define $\widetilde{K}(\ell+1) = a_t^\top a_t + R$.
Include $\mathsf{DS}_{\mathsf{priv}} \leftarrow \mathsf{DS}_{\mathsf{priv}} \cup (\widetilde{K}(\ell+1), t)$.
for $i = 2, \ldots, \ell$
  Compute $\widetilde{K}(i) \leftarrow \widetilde{K}(i) + a_t^\top a_t$. {Update the matrices.}
end
Find $j := \min\{p : K(p) \not\succeq K(\ell)\}$ and delete $K(p), \cdots, K(\ell-1)$
Update $K(p) = K(\ell), \ell = p$. {Mantain PSD ordering.}
Return $\mathsf{DS}_{\mathsf{priv}} := \left\{(\widetilde{K}(i), t_i)\right\}_{i=1}^{\ell}$.

Algorithm 4 SLIDING-PRIV($S; (\epsilon, \delta); W$)

Require: A stream, S, of row $\{a_t\}$, privacy parameters (ϵ, δ), and window size W.

Ensure: A positive semidefinite matrix $\widetilde{C}$ at the end of the stream.

Initialize $\mathsf{DS}_{\mathsf{priv}}$ to be an empty set, $r = d$.
while stream S has not ended
  Include new row, $\mathsf{DS}_{\mathsf{priv}} \leftarrow$ PRIV-INITIALIZE $(\mathsf{DS}_{\mathsf{priv}}; a_t; t; (\epsilon, \delta); W; r)$. {Algorithm 3}
  Update the data structure, $\mathsf{DS}_{\mathsf{priv}} \leftarrow$UPDATE-EXACT($\mathsf{DS}_{\mathsf{priv}}$). {Algorithm 2}
end
Let $\mathsf{DS}_{\mathsf{priv}} = \left\{(\widetilde{K}(1), t_1), \ldots, (\widetilde{K}(\ell), t_\ell)\right\}$ for some ℓ.
Output $\widetilde{C} = \widetilde{K}(1)$.

is (ϵ, δ)*-differential private. Further,* $\widetilde{C} \leftarrow$SLIDING-PRIV$(S; (\epsilon, \delta); W)$ *satisfies the following with probability* $1 - \frac{1}{\mathrm{poly}(d)}$

$$\left(A_W^\top A_W - (c\tau\log\tau)\mathbb{1}_d\right) \preceq \widetilde{C} \preceq \left(\frac{1}{(1-\eta)}A_W^\top A_W + (C\tau\log\tau)\mathbb{1}_d\right), \text{ where } \tau := d + \frac{14}{\epsilon^2}.$$

Proof. Consider an index $1 \le i \le \ell$ and the time epoch t when the stream of rows are different resulting in neighboring matrices $A_{[t_i,t]}$ and $A'_{[t_i,t]}$, we have $A_{[t_i,t]}^\top A_{[t_i,t]} - (A'_{[t_i,t]})^\top A'_{[t_i,t]} = u^\top u$, where u is a unit row vector. That is $A_{[t_i,t]}^\top A_{[t_i,t]} - (A'_{[t_i,t]})^\top A'_{[t_i,t]}$ is a rank-1 matrix.

Now $W \sim \mathsf{Wis}_d(\tau, \mathbb{1}_d)$. Let $\mathcal{P}$ denote the output distribution of our mechanism when run on the input matrix $A_{[t_i,t]}$ and $\mathcal{Q}$ denote the output of our algorithm on input matrix $A'_{[t_i,t]}$. Both distribution are supported on $\mathbf{S} := \mathbb{R}^{d\times d}$ matrices. For $M \in \mathbf{S}$, consider the privacy loss function $L(M) := \log\left(\frac{\mathcal{P}(M)}{\mathcal{Q}(M)}\right)$. When $T < t$, the output distribution of $\mathcal{P}$ and $\mathcal{Q}$ are identical, i.e., $L(M) = 0$. When $T = t$, the privacy proof follows by the choice of τ and Theorem 7. That is, $\mathsf{Pr}[L(M) \le \epsilon] \ge 1 - \delta$. For any time $T \ge t$, we have differential privacy because of the post-processing property.

For the space bound, note that the number of checkpoints stored by $\mathsf{DS}_{\mathsf{priv}}$ is $O\left(\frac{d}{\eta}\log W\right)$. This is because there are exactly d singular values and the matrix has polynomially bounded spectrum. Since at each checkpoints defined by t_i for $i \ge 1$ stores an $d \times d$ matrix, the total space used by the data structure $\mathsf{DS}_{\mathsf{priv}}$, and hence the algorithm SLIDING-PRIV, is $O\left(\frac{d^3}{\eta}\log W\right)$.

For the accuracy guarantee, we first note that the output of SLIDING-PRIV$(S; (\epsilon, \delta); W)$ is $\widetilde{K}(1)$, the first positive semidefinite matrix in the data structure $\mathsf{DS}_{\mathsf{priv}}$. Let $K(1)$ and $K(2)$ denote the covariance matrix formed between time epochs

$[t_1, T]$ and $[t_2, T]$, respectively. Since the window is sandwiched between the first and second timestamp, we have

$$K(2) \preceq A_W^\top A_W \preceq K(1). \tag{3}$$

Let $R(1)$ and $R(2)$ be matrices sampled from the Wishart distribution such that $\widetilde{K}(1) := K(1) + R(1)$ and $\widetilde{K}(2) := K(2) + R(2)$. Note that $\mathsf{DS}_{\mathsf{priv}}$ stores the set $\left\{\widetilde{K}(1), \widetilde{K}(2), \cdots, \widetilde{K}(\ell)\right\}$. Recall that $\sigma := \tau \log(\tau) = \left(d + \frac{\log(1/\delta)}{\epsilon^2}\right) \log\left(d + \frac{\log(1/\delta)}{\epsilon^2}\right)$. Using the standard result on the eigenvalue bounds of matrices sampled from Wishart distribution, we have that $\lambda_1(R(1)) \leq c\sigma$ and $\lambda_1(R(2)) \leq c\sigma$ for some constant $c > 1$ with probability $1 - \frac{1}{\mathrm{poly}(d)}$. Further $R(1)$ and $R(2)$ are positive semidefinite. Therefore, for $i \in \{1, 2\}$ we have

$$\mathsf{Pr}\left[K(i) - c\sigma \mathbb{1}_d \preceq \widetilde{K}(i) \preceq K(i) + c\sigma \mathbb{1}_d\right] \geq 1 - \frac{1}{\mathrm{poly}(d)}, \tag{4}$$

where $K(1)$ is the underlying covariance matrix formed during the time epochs $[t_1, t]$, $K(2)$ is the underlying covariance matrix formed during the time epochs $[t_2, t]$, and $c > 0$ is a constant.

We now condition on the event that Equation (4) holds for the rest of the proof. From the smooth-PSD property of the matrices in $\mathsf{DS}_{\mathsf{priv}}$, we know that $(1-\eta)\widetilde{A}^\top \widetilde{A} = (1-\eta)\widetilde{K}(1) \preceq \widetilde{K}(2)$. Using this with Equations (3) and (4), we arrive at

$$(1-\eta)\left(K(1) - c_1\sigma \mathbb{1}_d\right) \preceq (1-\eta)\widetilde{K}(1) \preceq \widetilde{K}(2) \preceq K(2) + c_2\sigma \mathbb{1}_d \preceq A_W^\top A_W + c_2\sigma \mathbb{1}_d \tag{5}$$

Rearranging the terms in Equation (5) gives us

$$K(1) \preceq \frac{1}{(1-\eta)} A_W^\top A_W + c_3\sigma \mathbb{1}_d, \text{ where } c_3 = c_1 + \frac{c_2}{(1-\eta)}. \tag{6}$$

Using Equation (6) in the right side positive semidefinite inequality of Equation (4), we have

$$\left(K(1) - c_1\sigma \mathbb{1}_d\right) \preceq \widetilde{K}(1) \preceq \frac{1}{(1-\eta)} A_W^\top A_W + c_3\sigma \mathbb{1}_d. \tag{7}$$

Using the left hand semidefinite inequality of Equation (3) in Equation (7), we get

$$\left(A_W^\top A_W - c_1\sigma \mathbb{1}_d\right) \preceq \widetilde{K}(1) \preceq \frac{1}{(1-\eta)} A_W^\top A_W + c_3\sigma \mathbb{1}_d. \tag{8}$$

Since $\widetilde{C} = \widetilde{K}(1)$ by the output of the algorithm, we have the desired bound. □

B.1. Application of Algorithm 4: sub-optimal algorithms for private matrix analysis

Theorem 10 gives the guarantee that Algorithm 4 outputs a matrix that approximates the spectrum of $A_W^\top A_W$ up to a small additive error in the spectrum. This in particular means that Algorithm 4 can be used to solve directional variance and PCA; however, the accuracy guarantees are sub-optimal.

The directional variance queries has the following form: the analyst gives a unit-length vector $x \in \mathbb{R}^d$ and wish to know the variance of A_W along x. A special case of directional variance queries is cut queries when a_t is the edges of a weighted graph, $d = n$, and the query is of form $\{0, 1\}^n$. Using Theorem 7 and the fact that $\langle x, x\rangle = \|x\|_2 = 1$, we have the following two results.

Theorem 11 (Directional variance queries)**.** *Let A_W be the matrix formed by last W updates, η be the given approximation parameter, (ϵ, δ) be the privacy parameter. Then there is an efficient (ϵ, δ)-differentially private algorithm that outputs a matrix C such that for any unit vector $x \in \mathbb{R}^d$,*

$$x^\top A_W^\top A_W x - c_1\tau\log(\tau) \leq x^\top C x \leq (1+\eta) x^\top A_W^\top A_W x + c_3\tau\log(\tau), \textit{where } \tau = d + \frac{14}{\epsilon^2}\log(4/\delta).$$

Corollary 1 (Cut queries)**.** *Let $\mathcal{G}_W$ be the graph formed by last W updates. There is an efficient (ϵ, δ)-differentially private algorithm that outputs a matrix C such that for any cut query $S \subseteq [n]$,*

$$\Phi_S(\mathcal{G}_W) - \frac{c|S|\sqrt{\tau\log\tau}}{\epsilon} \leq \mathsf{Out}_S \leq (1+\eta)\Phi_S(\mathcal{G}_W) + \frac{c|S|\sqrt{\tau\log\tau}}{\epsilon}, \textit{ where } \tau = n + \frac{14}{\epsilon^2}\log(4/\delta).$$

Since Theorem 10 preserves the spectrum of the covariance matrix, it can be used for a variety of tasks involving spectrum. In particular, we can use it to compute the principal component of the matrix streamed in the window. Let Π be the set of all rank-k orthonormal projection matrices, i.e., every matrix $P \in \Pi$ has rank k and satisfy $P^2 = P$ and $P = P^\top$. Now consider the following algorithm, SLIDING-PCA: compute $\widetilde{A}^\top \widetilde{A} \leftarrow$SLIDING-PRIV$(S; (\epsilon, \delta); W)$ and then solve the rank constrained problem using Lemma 5

$$\widetilde{X} = \operatorname*{argmin}_{P \in \Pi} \left\| \widetilde{A}(\mathbb{1}_d - P) \right\|_F^2 .$$

Theorem 12. *Given privacy parameters (ϵ, δ) and approximation parameter $\eta \in (0, 1/2)$, let A_W be the matrix formed by the last W updates as defined in equation (1) and Π be the set of all rank-k orthonormal projection matrices. Then* SLIDING-PCA *is an efficient (ϵ, δ)-differentially private algorithm that outputs a rank-k projection matrix $\widetilde{X} \in \mathbb{R}^{d \times d}$ such that*

$$\mathsf{Pr}\left[\left\| A_W(\mathbb{1}_d - \widetilde{X}) \right\|_F^2 \le (1 + 2\eta) \min_{P \in \Pi} \|A_W(\mathbb{1}_d - P)\|_F^2 + O\left(d\tau \log(\tau)\right) \right] \ge 1 - \frac{1}{\mathrm{poly}(d)},$$

where $\tau := \left(d + \frac{14}{\epsilon^2} \log(4/\delta)\right)$.

Proof. Let

$$\widehat{X} := \operatorname*{argmin}_{X \in \Pi} \|A_W(\mathbb{1}_d - X)\|_F^2 \qquad \text{and} \qquad \widetilde{X} := \operatorname*{argmin}_{X \in \Pi} \left\| \widetilde{A}(\mathbb{1}_d - X) \right\|_F^2 . \tag{9}$$

Then from the optimality of $\widetilde{X}$ and the fact that $\|Y\|_F^2 = \mathsf{Tr}\left(Y^\top Y\right)$ for any matrix Y, we have

$$\left\| \widetilde{A}(\mathbb{1}_d - \widetilde{X}) \right\|_F^2 \le \left\| \widetilde{A}(\mathbb{1}_d - \widehat{X}) \right\|_F^2 = \mathsf{Tr}\left((\mathbb{1}_d - \widehat{X})^\top \widetilde{A}^\top \widetilde{A} (\mathbb{1}_d - \widehat{X}) \right).$$

Using the right hand side semidefinite inequality in Equation (8), we have

$$\begin{aligned} \left\| \widetilde{A}(\mathbb{1}_d - \widetilde{X}) \right\|_F^2 &\le \frac{1}{(1-\eta)} \mathsf{Tr}\left((\mathbb{1}_d - \widehat{X})^\top A_W^\top A_W (\mathbb{1}_d - \widehat{X}) \right) + c(\mathbb{1}_d - \widehat{X})^\top (\mathbb{1}_d - \widehat{X}) \tau \log(\tau) \\ &\le \frac{1}{(1-\eta)} \left\| A_W(\mathbb{1}_d - \widehat{X}) \right\|_F^2 + cd \left(d + \frac{14}{\epsilon^2} \log(4/\delta) \right) \log \left(d + \frac{14}{\epsilon^2} \log(4/\delta) \right), \end{aligned} \tag{10}$$

where the first inequality follows from the fact that $(\mathbb{1}_d - \widehat{X})$ is a rank $d - k$ projection matrix and second equality follows from equation (9). Similarly, the left hand side inequality of Equation (8) and the fact that $(\mathbb{1}_d - \widehat{X})$ is a rank $d - k$ projection matrix yields

$$\begin{aligned} \left\| \widetilde{A}(\mathbb{1}_d - \widetilde{X}) \right\|_F^2 &\ge \mathsf{Tr}\left((\mathbb{1}_d - \widetilde{X})^\top A_W^\top A_W (\mathbb{1}_d - \widetilde{X}) \right) - c(\mathbb{1}_d - \widetilde{X})^\top (\mathbb{1}_d - \widetilde{X}) \tau \log(\tau) \\ &\ge \left\| A_W(\mathbb{1}_d - \widetilde{X}) \right\|_F^2 - cd \left(d + \frac{14}{\epsilon^2} \log(4/\delta) \right) \log \left(d + \frac{14}{\epsilon^2} \log(4/\delta) \right). \end{aligned} \tag{11}$$

Combining equations (10) and (11), we have Theorem 12. □

C. Approximate spectral histogram and proof of Theorem 1

The only source of randomness in the data structure in Section B is the one that enables us to preserve differential privacy. However, the resulting matrix analysis are sub-optimal in terms of achievable accuracy. The main reason is that we did not use any low-rank structure of the underlying matrix – adding a Wishart matrix makes the resulting matrix full rank. The natural question is whether additional randomness can help improve the bound when the matrix has a low-rank structure.

In this section, we introduce the *η-approximate smooth-PSD property*. We later show in Section D that these properties are sufficient to perform all the matrix analysis mentioned in Section 1 efficiently with respect to time, efficiency, and accuracy. In particular, by maintaining this property, we can maintain an intrinsic rank dependent bound. To maintain this property, our data structure stores a set of random matrices that approximates the spectrum of the original matrices.

Algorithm 5 UPDATE-APPROX($\mathsf{DS}_{\mathsf{priv}}$)

Require: A data structure $\mathsf{DS}_{\mathsf{priv}}$ storing a set of matrices $\widetilde{A}_{[t_1,t]}, \cdots, \widetilde{A}_{[t_\ell,t]}$ and corresponding time stamps $t_1, \cdots, t_\ell$.
Ensure: Updated matrices $\widetilde{A}_{[t_1,t]}, \cdots, \widetilde{A}_{[t_\ell,t]}$ and timestamps $t_1, \cdots, t_\ell$.

Define for $1 \le i \le \ell$,

$$\widetilde{K}(i) := \widetilde{A}_{[t_i,t]}^\top \widetilde{A}_{[t_i,t]}.$$

For $i = 1, \cdots \ell - 2$

  Find spectrally close checkpoints

$$j := \max\left\{p : \left(1 - \frac{\eta}{2}\right)\widetilde{K}(i) \preceq \widetilde{K}(p) \wedge (i < p \le \ell - 1)\right\} \tag{12}$$

  Delete $\widetilde{A}_{[t_{i+1},t]}, \cdots, \widetilde{A}_{[t_{j-1},t]}$. {It is important that we delete only up to index $j-1$.}
  Set k=1
  While $i + k \le \ell$
    Update the checkpoints: $\widetilde{A}_{[t_{i+k},t]} = \widetilde{A}_{[t_{j+k-1},t]}$, $t_{i+k} = t_{j+k-1}$.
  end
  $\ell := \ell + i - j + 1$.

Define $\mathsf{DS}_{\mathsf{priv}} := \left\{(\widetilde{A}_{[t_1,t]}, t_1), \cdots, (\widetilde{A}_{[t_\ell,t]}, t_\ell)\right\}$.

Output $\mathsf{DS}_{\mathsf{priv}}$.

The difference between spectral histogram property and η-approximate smooth histogram property is that in η-approximate smooth histogram property, we allow the matrices in the data structure to be a spectral approximation of the corresponding original matrices and in properties defined in item 2. Let $\widetilde{A}$ denotes the η-spectral approximation of the matrix A. Then η-approximate smooth histogram property is formally defined as follows:

Definition 7 (η-approximate-Smooth-PSD property)**.** *A data structure $\mathfrak{D}$ satisfy* η-approximate smooth-PSD *property if there exists an $s = \mathrm{poly}(n, \log W)$ such that*

1. $\mathfrak{D}$ consists of ℓ timestamps $\mathfrak{T} := \{t_1, \cdots, t_\ell\}$ and the corresponding matrices $\mathfrak{M} := \{\widetilde{M}_1, \cdots, \widetilde{M}_\ell\}$.
2. For $1 \le i \le s - 1$, at least one of the following holds:
   - If $t_{i+1} = t_i + 1$, then $\left(1 - \frac{\eta}{2}\right)\widetilde{M}_i^\top \widetilde{M}_i \not\preceq \widetilde{M}_{i+1}^\top \widetilde{M}_{i+1}$.
   - For all $1 \le i \le \ell - 2$:
     (a) $(1 - \eta) M_i^\top \widetilde{M}_i \preceq M_{i+1}^\top \widetilde{M}_{i+1}$.
     (b) $\left(1 - \frac{\eta}{2}\right)\widetilde{M}_i^\top \widetilde{M}_i \not\preceq \widetilde{M}_{i+2}^\top \widetilde{M}_{i+2}$, where $\widetilde{M}_i$ is $\frac{\eta}{4}$-spectral approximation of M_i and $\widetilde{M}_{i+2}$ is $\frac{\eta}{4}$-spectral approximation of M_{i+2}.
3. Let A_W be the matrix formed by the window W. Then $t_1 \le T - W + 1 \le t_2$.

We next show that we can maintain a data structure that allows efficient updates and a sequence of matrices that satisfy the η-approximate smooth-PSD property using an algorithm UPDATE-APPROX. We will show in equation (16) that the input to the UPDATE-APPROX is constructed in a specific manner; therefore, the matrices in the sequence satisfy the Loewner ordering.

Lemma 7. *Let $\mathfrak{D}$ be a data structure that at time T consists of ℓ tuples $\left\{(t_i, \widetilde{A}_{[t_i,t]})\right\}_{i=1}^{\ell}$, where $0 < t_i \le T$ for all i and*

$$\widetilde{A}_{[t_\ell,t]}^\top \widetilde{A}_{[t_\ell,t]} \preceq \ldots \preceq \widetilde{A}_{[t_1,t]}^\top \widetilde{A}_{[t_1,t]}^\top.$$

Let $\mathfrak{D}_{\mathsf{smooth}} \leftarrow$UPDATE-APPROX($\mathfrak{D}$) be the output of the algorithm UPDATE-APPROX*, defined in Algorithm 5. Then $\mathfrak{D}_{\mathsf{smooth}}$ satisfy the η-approximate smooth histogram property (Definition 7). Moreover, the algorithm $\mathfrak{D}_{\mathsf{smooth}}$ runs in $\mathrm{poly}(d, \ell)$ time.*

Proof. The run-time of $\mathfrak{D}_{\mathsf{smooth}}$ is straightforward, so we only concentrate on the correctness part. Consider a time epoch t and a succeeding time epoch $t' = t + 1$. First notice that if we cannot find a j in equation (12) for all $i = 1, \cdots, \ell - 2$,

then $\ell = O\left(\frac{d \log W}{\eta}\right)$. This is because the data structure $\mathsf{DS}_{\mathsf{priv}}$ satisfied η-approximate smooth-PSD property at time t and if no such j exists, then none of the properties are violated due to an update. As a result, equation (15) gives us that $\ell = O\left(\frac{d \log W}{\eta}\right)$. Hence, we assume that this is not the case and the data structure got updated between time epochs t and $t+1$. Let the data structure at time t be $\mathsf{DS}_{\mathsf{priv}}(t)$ and at time $t' = t+1$ be $\mathsf{DS}_{\mathsf{priv}}(t')$. That is,

$$\mathsf{DS}_{\mathsf{priv}}(t) := \left\{(t_1, \widetilde{A}_{[t_1,t]}); \cdots, (t_\ell, \widetilde{A}_{[t_\ell,t]})\right\} \quad \text{and} \quad \mathsf{DS}_{\mathsf{priv}}(t') := \left\{(t'_1, \widetilde{B}_{[t'_1,t+1]}); \cdots, (t'_\ell, \widetilde{B}_{[t'_\ell,t+1]})\right\}.$$

Let t_i be a timestamp in $\mathsf{DS}_{\mathsf{priv}}(T)$ where $i < \ell$. We can have two possibilities:

1. There is no $c \in [\ell]$ such that $t'_c = t_i$ and $t'_{c+1} = t_{i+1}$.
2. There is a $c \in [\ell]$ such that $t'_c = t_i$ and $t'_{c+1} = t_{i+1}$.

Fix the following notations for all $1 \le j \le \ell$:

$$\widetilde{K}(j) := \widetilde{A}^\top_{[t_j,t]} \widetilde{A}_{[t_j,t]} \qquad \text{and} \qquad \widetilde{L}(j) := \widetilde{B}^\top_{[t_j,t+1]} \widetilde{B}_{[t_j,t+1]}.$$

Let $K(j)$ and $L(j)$ be such that

$$\left(1 - \frac{\eta}{4}\right) K(j) \preceq \widetilde{K}(j) \preceq \left(1 + \frac{\eta}{4}\right) K(j) \quad \text{and} \quad \left(1 - \frac{\eta}{4}\right) L(j) \preceq \widetilde{L}(j) \preceq \left(1 + \frac{\eta}{4}\right) L(j). \tag{13}$$

As we will see later (in equation (16)), the input to the UPDATE-APPROX are constructed in a specific manner that will ensure that $K(j)$ and $L(j)$ satisfy the Loewner ordering. Let us consider the first possibility, i.e., there is no $j \in [s]$ such that $t'_j = t_i$ and $t'_{j+1} = t_{i+1}$. By the update rule,

$$\left(1 - \frac{\eta}{2}\right) \widetilde{L}(j) \preceq \widetilde{L}(j+1).$$

Using Proposition 6, we have

$$(1-\eta) L(j) \prec \left(1 - \frac{\eta}{2}\right) \left(\frac{1 - \frac{\eta}{4}}{1 + \frac{\eta}{4}}\right) L(j).$$

Equation (13) gives us a relationship between $L(j)$ and $\widetilde{L}(j)$:

$$\left(1 - \frac{\eta}{2}\right) \left(\frac{1 - \frac{\eta}{4}}{1 + \frac{\eta}{4}}\right) L(j) \preceq \left(\frac{1 - \frac{\eta}{2}}{1 + \frac{\eta}{4}}\right) \widetilde{L}(j).$$

From the update rule, it follows that

$$\left(\frac{1 - \frac{\eta}{2}}{1 + \frac{\eta}{4}}\right) \widetilde{L}(j) \preceq \frac{1}{1 + \frac{\eta}{4}} \widetilde{L}(j+1)$$

Finally, another application of Equation (13) gives us

$$\frac{1}{1 + \frac{\eta}{4}} \widetilde{L}(j+1) \preceq L(j+1).$$

Therefore, we have the relation listed in item 2a in Definition 7. Furthermore, due to the maximality condition in the update rule (lines 2 and 2), there exists an $k \in [d]$ such that

$$s_k(\widetilde{L}(j+2)) < \left(1 - \frac{\eta}{2}\right) s_k(\widetilde{L}(j)),$$

Algorithm 6 SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$

Require: A stream, S, of row vectors $\{a_t\}$ and the desired rank of matrices, r, privacy parameters (ϵ, δ), and window size W.

Ensure: A matrix $\widetilde{A} \in \mathbb{R}^{\frac{4r}{\eta} \times d}$ at the end of the stream.

1: **Initialize** $\mathsf{DS_{priv}}$ to be an empty set and $\Phi \in \mathbb{R}^{\frac{4r}{\eta} \times (d+1)}$ such that every entry $\Phi[i,j] \sim \mathcal{N}(0, \frac{\eta}{4r})$.
2: **while** stream S has not ended
3: $\quad \mathsf{DS_{priv}} \leftarrow$ PRIV-INITIALIZE-APPROX $\left(\left(\mathsf{DS_{priv}}; a_t; t; \left(\frac{\eta}{\log(W)}\epsilon, \frac{\eta}{\log(W)}\delta\right); W; \Phi; r\right)\right.$. {Algorithm 7}
4: $\quad$ **Update** the data structure $\mathsf{DS_{priv}} \leftarrow$UPDATE-APPROX$(\mathsf{DS_{priv}})$. {Algorithm 5}
5: **end**
6: **Let** $\mathsf{DS_{priv}} = \left\{(t_i, \widetilde{A}_{[t_i,t]})\right\}_{i=1}^{\ell}$ for some ℓ.
7: **Output** $\widetilde{A} = \widetilde{A}_{[t_1,t]}$.

where $\{s_1(D), \cdots, s_d(D)\}$ denotes the singular values of an $d \times d$ matrix D. This proves the statement of the Lemma 7 for the first scenario.

Now let us consider the second possibility. Again by the update rule, we have

$$\left(1 - \frac{\eta}{2}\right)\widetilde{L}(j) \preceq \widetilde{L}(j+1).$$

It follows from Proposition 6 and Theorem 5 that

$$(1-\eta)L(j) \prec \left(1 - \frac{\eta}{2}\right)\left(\frac{1-\frac{\eta}{4}}{1+\frac{\eta}{4}}\right)L(j) \preceq \left(\frac{1-\frac{\eta}{2}}{1+\frac{\eta}{4}}\right)\widetilde{L}(j).$$

Another application of the update rule and Theorem 5 gives us

$$\left(\frac{1-\frac{\eta}{2}}{1+\frac{\eta}{4}}\right)\widetilde{L}(j) \preceq \frac{1}{1+\frac{\eta}{4}}\widetilde{L}(j+1) \preceq L(j+1),$$

where the last positive semidefinite inequality follows from Theorem 5. The second part of *η-approximate smooth-PSD property* follows similarly as in the case of the first case. This proves the statement of the Lemma 7 for the second sdenario. □

Lemma 7 gives the guarantee that if we are given a set of positive semidefinite matrices in the Loewner ordering, then we can efficiently maintain a small set of positive semidefinite matrices that satisfy η-approximate smooth histogram property. The idea of our algorithm for spectral approximation is to ensure that UPDATE-APPROX always receives a set of positive semidefinite matrices. This is attained by our algorithm PRIV-INITIALIZE-APPROX, which gets as input a new row and updates all the matrices in the current data structure. We use both these subroutines in our main algorithm, SLIDING-PRIV-APPROX, that receives a stream of rows and call these two subroutines on every new update. Equipped with Lemma 7, we show that SLIDING-PRIV-APPROX, described in Algorithm 6 provides the following guarantee.

Theorem 13 (Private spectral approximation under sliding window)**.** *Given the privacy parameter ϵ, δ, window size W, desired rank r approximation parameter η, and $S = (a_t)_{t>0}$ be the stream such that $a_t \in \mathbb{R}^d$. Let A_W be the matrix formed at time T using the last W updates as defined in equation (1). Then we have the following:*

1. SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ *is (ϵ, δ)-differential privacy.*
2. $\widetilde{A} \leftarrow$SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ *satisfies the following with probability* $9/10$*:*

$$\left(1 - \frac{\eta}{4}\right)\left(A_W^\top A_W + \frac{cr\log^2(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d\right) \preceq \widetilde{A}^\top\widetilde{A} \preceq \frac{(1+\frac{\eta}{4})^2}{(1-\eta)}A_W^\top A_W + \frac{cr\log^2(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d.$$

3. *The space required by* SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ *is* $O(\frac{dr^2}{\eta^2}\log W)$.

Algorithm 7 PRIV-INITIALIZE-APPROX($\mathsf{DS}_{\mathsf{priv}}; a_t; t; (\epsilon, \delta); W; \Phi; r$)

Require: A new row $a_t \in \mathbb{R}^d$, a data structure $\mathsf{DS}_{\mathsf{priv}}$ storing a set of timestamps $t_1, \cdots, t_\ell$ and set of matrices

$$\widetilde{A}_{[t_1,t]}, \cdots, \widetilde{A}_{[t_\ell,t]},$$

current time t, window size W, and random matrix $\Phi \in \mathbb{R}^{\frac{4r}{\eta} \times (d+1)}$.

Ensure: Updated matrices $\widetilde{A}_{[t_1,t]}, \cdots, \widetilde{A}_{[t_\ell,t]}$ and timestamps $t_1, \cdots, t_\ell$.

1: **if** $t_2 < t - W + 1$

2: **Set** $t_j = t_{j+1}, \widetilde{A}_{[t_j,t]} := \widetilde{A}_{[t_{j+1},t]}$ for $1 \le j \le \ell - 1$ {Delete the expired timestamp.}

3: **end**

4: **Set** $t_{\ell+1} = t, \sigma = \frac{16\sqrt{r\log(4/\delta_0)} + \log(4/\delta)}{\epsilon_0}$ for $\epsilon_0 = \frac{\log(W)}{\eta}\epsilon, \delta_0 = \frac{\log(W)}{\eta}\delta$, and

$$\widetilde{A}_{[t_{\ell+1},t]} := \Phi \begin{pmatrix} \sigma \mathbb{1}_d \\ a_t \end{pmatrix} \in \mathbb{R}^{\frac{4r}{\eta} \times d}, \quad \widetilde{A} := \Phi \begin{pmatrix} \mathbf{0}^{d \times d} \\ a_t \end{pmatrix} \in \mathbb{R}^{\frac{4r}{\eta} \times d}. \tag{14}$$

5: **Update** $\mathsf{DS}_{\mathsf{priv}} \leftarrow \mathsf{DS}_{\mathsf{priv}} \cup (\widetilde{A}_{[t_{\ell+1},t]}, t)$.

6: **for** $i = 2, \ldots, \ell$

7: **Compute** $\widetilde{A}_{[t_i,t]} \leftarrow \widetilde{A}_{[t_i,t]} + \widetilde{A}$. {Update the matrices.}

8: **end**

9: **Return** $\mathsf{DS}_{\mathsf{priv}} := \left\{ (\widetilde{A}_{[t_i,t]}, t_i) \right\}_{i=1}^{\ell}$.

Proof. We divide the proof of Theorem 13 in three parts: (i) the privacy proof (Lemma 9); (ii) accuracy proof (Lemma 10); and (iii) the space complexity proof (Lemma 8).

Lemma 8 (Space complexity)**.** *Let η, ϵ, δ be as in Theorem 13. The total space required to maintain the data structure $\mathsf{DS}_{priv}$is $O\left(\frac{r^2 d}{\eta} \log n \log W\right)$.*

Proof. Note that there are at most r singular values. Since each checkpoints are at least $(1 - \frac{\eta}{2})$ apart for at least one singular value and all the non-zero singular values are bounded by a polynomial, there can be at most

$$c \log_{(1-\frac{\eta}{2})}(W) = c\frac{\log W}{\log(1 - \frac{\eta}{2})} \le \frac{2c}{\eta} \log W \tag{15}$$

checkpoints that sees the jump in a specific singular value. Since each checkpoint defined by t_i (for $i \ge 2$) stores a covariance matrix, using Theorem 5, the total space used by the checkpoints $\left\{ \widetilde{A}_{[t_i,t]}, t_i \right\}_{i \ge 2}$ is $O\left(\frac{rd}{\eta} \log W\right)$. This finishes the proof of Lemma 8. □

Lemma 9 (Privacy guarantee)**.** *Let $S := \{a_t\}_{t=1}^T$ be the streams of row and A_W be the rank r matrix formed in the time epoch $[T - W + 1, W]$ as defined in equation (1). Then $\widetilde{A} \leftarrow$ SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ is (ϵ, δ)-differentially private.*

Proof. Let $\mathcal{P}$ denote the output distribution of our algorithm when run on the input matrix $A_{[t_i,t]}$ and similarly let $\mathcal{Q}$ denote the output of our algorithm on input matrix $A'_{[t_i,t]}$. Both distribution are supported on $\mathbf{S} := \mathbb{R}^{d \times d}$ matrices. For $M \in \mathbf{S}$, consider the privacy loss function

$$L(M) := \log\left(\frac{\mathcal{P}(M)}{\mathcal{Q}(M)}\right).$$

Our goal is to show that $\mathsf{Pr}_M[L(M) \le \epsilon] \ge 1 - \delta$ for all T. There are three cases:

1. When $T < t$, the output distributions of $\mathcal{P}$ and $\mathcal{Q}$ are identical. It follows that $L(M) = 0$.
2. When $T = t$: In this case, we note that all singular values of matrix is at least σ. Therefore, Theorem 8 implies that $\mathsf{Pr}_M[L(M) \le \epsilon_0] \ge 1 - \delta_0$.

3. When $T > t$, privacy follows due to the post processing property of differential privacy.

Now each of the streamed data appears in at most $\frac{\log(W)}{\eta}$ checkpoints (equation (15)) and the output is just one checkpoint at the end of the stream. Note that each of the checkpoints is a P-sum (Definition 5) The privacy guarantee now follows using Fact 1 and the composition theorem. □

Lemma 10 (Accuracy guarantee)**.** *Let $S := \{a_t\}_{t=1}^{T}$ be the streams of row and A_W be the rank r matrix formed in the time epoch $[T - W + 1, W]$ as defined in equation (1). Then $\widetilde{A} \leftarrow$ SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ satisfies the following*

$$\left(1 - \frac{\eta}{4}\right)\left(A_W^\top A_W + \frac{cr\log^2(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d\right) \preceq \widetilde{A}^\top\widetilde{A} \preceq \frac{(1+\frac{\eta}{4})}{(1-\eta)}A_W^\top A_W + \frac{cr\log^2(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d$$

for a constant $c > 0$ with probability at least $1 - \beta$, where the probability is taken over the internal coin tosses of PRIV-INITIALIZE-APPROX.

Proof. Note that the output of SLIDING-PRIV-APPROX $(S; r; (\epsilon, \delta); W)$ is $\widetilde{A}_{[t_1,t]}$, the first matrix in the data structure $\mathsf{DS}_{\mathsf{priv}}$. Fix the following notations for $j \in [\ell]$:

$$\begin{aligned} K(j) &:= A_{[t_j,t]}^\top A_{[t_j,t]}, \quad \widetilde{K}(j) := \widetilde{A}_{[t_j,t]}^\top \widetilde{A}_{[t_j,t]} \\ \widehat{K}(j) &:= \begin{pmatrix} \sigma\mathbb{1}_d \\ A_{[t_j,t]} \end{pmatrix}^\top \begin{pmatrix} \sigma\mathbb{1}_d \\ A_{[t_j,t]} \end{pmatrix} = A_{[t_j,t]}^\top A_{[t_j,t]} + \sigma^2\mathbb{1}_d = K(j) + \sigma^2\mathbb{1}_d. \end{aligned} \tag{16}$$

where

$$\sigma := \frac{16\sqrt{r\log(4/\delta)} + \log(4/\delta)}{\epsilon} \quad \text{and} \quad \widehat{A}_{[t_j,t]} := \begin{pmatrix} \sigma\mathbb{1}_d \\ A_{[t_j,t]} \end{pmatrix}.$$

Since the starting time of the window is sandwiched between the first and second timestamps, the matrix $A_W^\top A_W$ is approximated in the following manner:

$$K(2) \preceq A_W^\top A_W \preceq K(1) \tag{17}$$

Moreover, from the η-approximate smooth-PSD property, we have the following relation between $\widehat{K}(1)$ and $\widehat{K}(2)$ (equivalently $\widehat{A}_{[t_1,t]}$ and $\widehat{A}_{[t_2,t]}$):

$$(1 - \eta)\widehat{K}(1) \preceq \widehat{K}(2). \tag{18}$$

Since $\widetilde{K}(j)$ is $\frac{\eta}{4}$-spectral approximation of $\widehat{K}(j)$ for all $j \in [\ell]$, setting $\zeta = \frac{\eta}{4}$ and $A := \widehat{A}_{[t_i,t]}$ for $i = \{1, 2\}$ in Theorem 5, we have with probability $1 - \beta$,

$$\left(1 - \frac{\eta}{4}\right)\widehat{K}(1) \preceq \widetilde{K}(1) \preceq \left(1 + \frac{\eta}{4}\right)\widehat{K}(1), \quad \left(1 - \frac{\eta}{4}\right)\widehat{K}(2) \preceq \widetilde{K}(2) \preceq \left(1 + \frac{\eta}{4}\right)\widehat{K}(2). \tag{19}$$

Using equation (16) and (19), we have that

$$\left(1 - \frac{\eta}{4}\right)(K(1) + \sigma^2\mathbb{1}_d) \preceq \widetilde{K}(1). \tag{20}$$

Since adding positive semidefinite matrices preserves the Loewner ordering, using equation (17) we can deduce the following implication from equation (20), we get the following:

$$\left(1 - \frac{\eta}{4}\right)(A_W^\top A_W + \sigma^2\mathbb{1}_d) \preceq \left(1 - \frac{\eta}{4}\right)(K(1) + \sigma^2\mathbb{1}_d) \preceq \widetilde{K}(1). \tag{21}$$

This completes the proof of the lower bound on $\widetilde{K}(1)$. To upper bound $\widetilde{K}(1)$, equation (19) gives us

$$\widetilde{K}(1) \preceq \left(1 + \frac{\eta}{4}\right) \widehat{K}(1). \tag{22}$$

Combined with equation (18), equation (22) gives us

$$\widetilde{K}(1) \preceq \frac{\left(1 + \frac{\eta}{4}\right)}{(1-\eta)} \widehat{K}(2). \tag{23}$$

Combining equation (23) with equation (16) and equation (17) then gives us

$$\widetilde{K}(1) \preceq \frac{\left(1 + \frac{\eta}{4}\right)}{(1-\eta)} \left(A_W^\top A_W + \sigma^2 \mathbb{1}_d\right). \tag{24}$$

This completes the proof of the upper bound on $\widetilde{K}(1)$. The statement of the lemma follows by combining equations (21) and (24) and substituting the value of σ. □

Theorem 13 follows by combining Lemma 9, Lemma 10, and Lemma 8. □

Note that $(1 + \frac{\eta}{4}) \leq (1 + 2\eta)(1 - \eta)$ for $\eta \leq \frac{3}{8}$. Scaling the value of η, we have the following:

Corollary 2. *Let $\eta \in (0, \frac{3}{8})$. Then* SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ *outputs a matrix $\widetilde{A}$ such that, for some constant $c > 0$,*

$$(1-\eta)\left(A_W^\top A_W - \frac{cr\log(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d\right) \preceq \widetilde{A}^\top \widetilde{A} \preceq (1+\eta)\left(A_W^\top A_W + \frac{cr\log(1/\delta)\log(W)}{\eta\epsilon}\mathbb{1}_d\right).$$

If instead, we compute $\widetilde{A}^\top \widetilde{A} - \sigma^2 \mathbb{1}_d$, then we get the following theorem.

Theorem 14. *Given the privacy parameters (ϵ, δ), window size W, desired rank r, and approximation parameter β, let $S = \{a_t\}_{t>0}$ be the streams such that $a_t \in \mathbb{R}^d$. Let A_W be the matrix formed at time T using the last W updates as defined in equation (1). Let $\widetilde{A} \leftarrow$* SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$. *Then we can compute a positive semidefinite matrix C such that*

$$\mathsf{Pr}\left[(1-\eta)\, A_W^\top A_W - \frac{c\eta r\log(1/\delta)\log(W)}{4\eta\epsilon^2}\mathbb{1}_d \preceq C \preceq (1+\eta)\, A_W^\top A_W + \frac{c\eta r\log(1/\delta)\log(W)}{\eta\epsilon^2}\mathbb{1}_d.\right] \geq 1 - \beta.$$

Proof. Compute $\widetilde{A} \leftarrow$ SLIDING-PRIV-APPROX$(S; r; (\epsilon, \delta); W)$ and let $C = \widetilde{A}^\top \widetilde{A} - \sigma^2 \mathbb{1}_d$. Then using Corollary 2 gives us the required bound. □

D. Applications of Theorem 1

Algorithm 6 can be used in solving many matrix analysis problems with better additive error. As a warm up, we consider directional variance queries.

D.1. Directional covariance queries

Theorem 11 is true for any d-dimensional unit vector $x \in \mathbb{R}^d$. However, in many practical scenarios, it is infeasible to ask all possible questions, but only a bounded number of queries. If we are given an apriori bound q on the number of queries an analyst can make, we can apply SLIDING-PRIV-APPROX to get dimension independent bound.

Theorem 15 (Directional variance queries)**.** *Given privacy parameters (ϵ, δ) and approximation parameter $\eta \in (0, 1/2)$, let A_W be the matrix formed by last W updates as defined in equation (1). Given a bound q, on the number of queries that can be made, there is an efficient (ϵ, δ)-differentially private algorithm that outputs a matrix C such that for any set of q unit vector queries $x_1, \cdots, x_q \in \mathbb{R}^d$, we have for all $i \in [q]$*

$$\langle x_i A_W, A_W x_i\rangle - \frac{c\eta\log q\log d}{\epsilon} \leq \langle x_i, Cx_i\rangle \leq \frac{1}{(1-\eta)}\langle x_i A_W, A_W x_i\rangle + \frac{c\eta\log q\log d}{\epsilon}.$$

For the rest of this section, we will explore applications of Algorithm 6 in principal component analysis in both restricted and unrestricted form. Recall that Theorem 12 gives an accuracy bound on PCA that depends linearly on the dimension of the data. However, one would ideally like the dependencies to be sublinear in d. In this section, we give a method to achieve this.

For this, we introduce a new concept which we call *private projection preserving summary*, which can be seen as the private analogue of projection cost preserving sketches (Cohen et al., 2015). However, it is far from clear if the techniques used in non-private literature extends straightforwardly. We show that Algorithm 6 with proper choice of parameters provides us with one such summary. This can be later employed to solve restricted and unrestricted principal component analysis.

D.2. Private Projection Preserving Summary

In this section, we introduce a notion called *private projection preserving summary*. This notion would be useful in giving our bounds on the principal component analysis and in giving the first bound on restricted principal component analysis.

Definition 8 (Private Projection Preserving Summary)**.** *Let k be a desired rank. Given a set of rank-k projection matrices Π, a matrix $\widetilde{A} \in \mathbb{R}^{n\times d}$ is called a private projection preserving summary of $A \in \mathbb{R}^{n\times d}$ with error $0 \le \eta < 1$ if it is (ϵ, δ) differentially private and for all $P \in \Pi$,*

$$(1-\eta)\left\|A - PA\right\|_F^2 - \tau_1 \le \left\|\widetilde{A} - P\widetilde{A}\right\|_F^2 \le (1-\eta)\left\|A - PA\right\|_F^2 + \tau_2$$

for τ_1, τ_2 that depends on k, d, ϵ, δ.

Private projection preserving summary can be seen as the private analogue of projection-cost preserving sketches introduced by (Cohen et al., 2015); however, there are some key differences. First of all, we do not require a constant c that depends only on A and $\widetilde{A}$. Second of all, there is a difference in quantifier. We do not require the private projection preserving summary to be with respect to all rank-k projection matrices but a predefined set Π of projection matrices.

Lemma 11. *Let k be the desired rank, η be the approximation parameter, and (ϵ, δ) be the privacy parameter. Let Π be the set of all rank-k projection matrices. Then for a given matrix A_W formed by the last W updates as defined in equation (1), the output $\widetilde{A} \leftarrow$SLIDING-PRIV-PCP$\left(S; \frac{k+\log(1/\beta)}{\eta}; (\epsilon, \delta); W\right)$ is (ϵ, δ)-differentially private and satisfies the following for all $P \in \Pi$ with probability at least $1-\beta$,*

$$(1-6\eta)\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F - c_1 K \le \left\|A(\mathbb{1}_d - P)\right\|_F \le (1+6\eta)\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F + c_2 K,$$

where

$$K := \frac{\sqrt{k' d \log(1/\delta)\log(W)}}{\eta\epsilon} \qquad \text{for} \qquad k' := k + \log(1/\beta).$$

Proof. Recall that

$$\widehat{A} = \begin{pmatrix} \sigma \mathbb{1}_d \\ A \end{pmatrix} \qquad \text{where} \qquad \sigma = \frac{16\sqrt{r\log(4/\delta)} + \log(4/\delta)}{\epsilon}$$

is as defined in Algorithm 5. By subadditivity of Frobenius norm, we have

$$\left\|A(\mathbb{1}_d - P)\right\|_F - \frac{\sqrt{k' d \log(1/\delta)\log(W)}}{\eta\epsilon} \le \left\|\widehat{A}(\mathbb{1}_d - P)\right\|_F \le \left\|A(\mathbb{1}_d - P)\right\|_F + \frac{\sqrt{k' d \log(1/\delta)\log(W)}}{\eta\epsilon}. \tag{25}$$

We wish to use Lemma 4. For this, we design E_1, E_2, E_3 and E_4 that satisfy the conditions of Lemma 4. Let $\widehat{A} = \widehat{V}\widehat{S}\widehat{U}^\top$ be a singular value decomposition. Let $\widehat{U}_k$ be the top k-left singular vectors of $\widehat{A}$ and let $P_1 = \widehat{U}_k\widehat{U}_k^\top$ be a rank-k orthonormal projection matrix. Then we define

$$E_1 := P_1^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_1 - P_1^\top \widehat{A}^\top \widehat{A} P_1.$$

From the *subspace embedding property* of Φ (Clarkson & Woodruff, 2017), we have that

$$-\eta_1 \widehat{A}^\top \widehat{A} \preceq E_1 \preceq \eta_1 \widehat{A}^\top \widehat{A}.$$

Let $P_2 := (\mathbb{1}_d - P_1)$ and define $E_2 := P_2^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_2 - P_2^\top \widehat{A}^\top \widehat{A} P_2$. By construction E_2 is symmetric. Moreover, using Kane & Nelson (2014, Theorem 21) gives us

$$\begin{aligned}\mathsf{Tr}\left(E_2\right) &= \mathsf{Tr}\left(P_2^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_2 - P_2^\top \widehat{A}^\top \widehat{A} P_2\right) = \mathsf{Tr}\left(P_2^\top \widehat{A}^\top (\Phi^\top \Phi - \mathbb{1}_d) \widehat{A} P_2\right) \\ &\leq \left\|\widehat{A} P_2\right\|_F \left\|\Phi^\top \Phi - \mathbb{1}_d\right\|_2 \left\|\widehat{A} P_2\right\|_F = \left\|\widehat{A} - \widehat{A} P_1\right\|_F \left\|\Phi^\top \Phi - \mathbb{1}_d\right\|_2 \left\|\widehat{A} - \widehat{A} P_1\right\|_F \\ &= \left\|\widehat{A} - [\widehat{A}]_k\right\|_F^2 \left\|\Phi^\top \Phi - \mathbb{1}_d\right\|_2 \leq \frac{\eta}{k} \left\|\widehat{A} - [\widehat{A}]_k\right\|_F^2\end{aligned}$$

because P_2 is the orthogonal projection to the top-k singular space of $\widehat{A}$. Using the inequality relationship between trace norm and Frobenius norm, it follows that

$$\sum_{i=1}^{k} |\lambda_i(E_2)| \leq \sqrt{k} \left\|E_2\right\|_F.$$

Now we define the matrices E_3 and E_4 using the cross terms as below.

$$E_3 := P_1^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_2 - P_1^\top \widehat{A}^\top \widehat{A} P_2 \qquad \text{and} \qquad E_4 := P_2^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_1 - P_2^\top \widehat{A}^\top \widehat{A} P_1.$$

We show the desired bound for the case of E_3. The case for E_4 follows similarly. First of all, by definition of P_1 and the fact that P_1 and P_2 are orthogonal to each other, we simply have

$$E_3 = P_1^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_2 - P_1^\top \widehat{A}^\top \widehat{A} P_2 = P_1^\top \widehat{A}^\top \Phi^\top \Phi \widehat{A} P_2 \tag{26}$$

Using Kane & Nelson (2014, Theorem 21), if follows from the singular value decomposition of $\widehat{A}$ and $\left\|\widehat{U}_k\right\|_F = \sqrt{k}$, that

$$\mathsf{Tr}\left(E_3^\top (\widehat{A}^\top \widehat{A})^\dagger E_3\right) = \left\|\widehat{U}_k \Phi^\top \Phi \widehat{A}(\mathbb{1} - P_1)\right\|_F^2 \leq \eta^2 \left\|\widehat{A} - [\widehat{A}]_k\right\|_F^2 \tag{27}$$

as required. Using Lemma 4, we have

$$(1 - 5\eta) \left\|\Phi \widehat{A}(\mathbb{1}_d - P)\right\|_F \leq \left\|\widehat{A}(\mathbb{1}_d - P)\right\|_F + \eta \left\|\widehat{A} - [\widehat{A}]_k\right\|_F \leq (1 + 5\eta) \left\|\Phi \widehat{A}(\mathbb{1}_d - P)\right\|_F.$$

Rearranging the term and using Equation (25) completes the proof of Lemma 11. □

D.3. Private Principal Component Analysis

The additive error of (ϵ, δ)-differentially private algorithm under sliding window model to compute the principal component using spectral histogram framework depends linearly on the dimension. In this section, we show that we can improve the accuracy guarantee significantly. Let Π be the set of all rank-k orthonormal projection matrices. We use the following algorithm for this purpose:

SLIDING-PRIV-PCA

1. Compute $\widetilde{A} \leftarrow$SLIDING-PRIV-APPROX$\left(S; \frac{k+\log(1/\beta)}{\eta}; (\epsilon, \delta); W\right)$.
2. Solve the following rank-constrained problem using the result of Friedland & Torokhti (2007)

$$\widetilde{X} := \operatorname*{argmin}_{P \in \Pi} \left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F.$$

Theorem 16. *Given privacy parameters (ϵ, δ), Π be the set of all rank-k orthonormal projection matrices and approximation parameter $\eta \in (0, 1/2)$, let A_W be the matrix formed by the last W updates as defined in equation (1). Then* SLIDING-PRIV-PCA *is an efficient (ϵ, δ)-differentially private algorithm that outputs a rank-k orthonormal projection matrix $\widetilde{X} \in \mathbb{R}^{d \times d}$ such that with probablity $1 - \beta$,*

$$\left\|A_W - A_W \widetilde{X}\right\|_F \leq \left(\frac{1 + 6\eta}{1 - 6\eta}\right) \min_{P \in \Pi} \left\|A_W(\mathbb{1}_d - P)\right\|_F + O\left(\frac{\log(1/\delta)}{\eta\epsilon} \sqrt{d \log(W)(\log(d))(k + \log(1/\beta))}\right).$$

Proof. Let $k' = k + \log(1/\beta)$. Define

$$\widehat{X} := \operatorname*{argmin}_{X \in \Pi} \|A_W(\mathbb{1}_d - P)\|_F \qquad \text{and} \qquad \widetilde{X} := \operatorname*{argmin}_{X \in \Pi} \left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F .$$

Using the left hand inequality in Lemma 11, we have

$$\left\|\widetilde{A}(\mathbb{1}_d - \widetilde{X})\right\|_F \leq \left\|\widetilde{A}(\mathbb{1}_d - \widehat{X})\right\|_F \leq \frac{1}{(1-6\eta)} \left(\left\|A_W(\mathbb{1}_d - \widehat{X})\right\|_F + \frac{k' d \log(1/\delta)}{\epsilon}\right). \tag{28}$$

Similarly, using the right hand inequality in Lemma 11, we have

$$\left\|\widetilde{A}(\mathbb{1}_d - \widetilde{X})\right\|_F \geq \frac{1}{(1+6\eta)} \left(\left\|A_W(\mathbb{1}_d - \widetilde{X})\right\|_F - \frac{k' d \log(1/\delta)}{\epsilon}\right). \tag{29}$$

Combining Equations (28) and (29), we have the result. □

D.4. Private Restricted Principal Component Analysis

In this section, we prove our result on private restricted principal component analysis. The traditional notion of principal component analysis minimizes over all possible set of rank-k projection matrices. However, recently researchers in machine learning has found applications where the rank-k projection matrices also satisfy other structural properties, like sparsity and non-negativity (Asteris et al., 2014; Yuan & Zhang, 2013; Zass & Shashua, 2006). There are also non-private algorithms for such problems as well. On the other hand, there is no prior work in privacy preserving literature for structural principal component analysis.

We show the following for restricted principal component analysis:

Theorem 17. *Given privacy parameters (ϵ, δ) and approximation parameter $\eta \in (0, 1/2)$, let A_W be the matrix formed by the last W updates as defined in equation (1) and Π be a given set of rank-k orthonormal projection matrices. Algorithm* SLIDING-PRIV-PCP *outputs $\widetilde{A}$ satisfying (ϵ, δ)-differential privacy. Let $P \in \Pi$ be a projection matrix satisfying*

$$\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F \leq \gamma \cdot \min_{X \in \Pi} \left\|\widetilde{A}(\mathbb{1}_d - X)\right\|_F . \tag{30}$$

Then with probablity $1 - \beta$,

$$\|A_W(\mathbb{1}_d - P)\| \leq \left(\frac{1+\eta}{1-\eta}\right) \gamma \cdot \min_{X \in \Pi} \|A_W(\mathbb{1}_d - X)\| + O\left(\frac{\sqrt{k' d \log(W)} \log(1/\delta)}{\eta \epsilon}\right).$$

Proof. Define

$$\widehat{P} = \operatorname*{argmin}_{P \in \Pi} \|A_W(\mathbb{1}_d - P)\|_F , \quad \widetilde{P} = \operatorname*{argmin}_{P \in \Pi} \left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F .$$

Since $\widetilde{P}$ is a minimizer, it implies the following

$$\left\|\widetilde{A}(\mathbb{1}_d - \widetilde{P})\right\|_F \leq \left\|\widetilde{A}(\mathbb{1}_d - \widehat{P})\right\|_F \tag{31}$$

as $\widehat{P} \in \Pi$. Combining with the equation (30), we have

$$\left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F \leq \gamma \left\|\widetilde{A}(\mathbb{1}_d - \widetilde{P})\right\|_F . \tag{32}$$

for $P \in \Pi$ as defined in equation (30). Using the right hand side inequality of Lemma 11, we have

$$(1+\eta) \left\|\widetilde{A}(\mathbb{1}_d - P)\right\|_F + \frac{\sqrt{k' d \log(1/\delta)}}{\epsilon} \geq \|A_W(\mathbb{1}_d - P)\|_F \tag{33}$$

Combining Equation (32) and (33), we have

$$\|A_W(\mathbb{1}_d - P)\|_F \leq (1+\eta)\gamma \left\|\widetilde{A}(\mathbb{1}_d - \widetilde{P})\right\|_F + \frac{\sqrt{k'd\log(1/\delta)}}{\epsilon}. \tag{34}$$

Using the left hand side inequality of Lemma 11, we have

$$\left\|\widetilde{A}(\mathbb{1}_d - \widehat{P})\right\|_F \leq \left(\frac{1}{1-\eta}\right)\left\|A_W(\mathbb{1}_d - \widehat{P})\right\|_F + \frac{\sqrt{k'd\log(1/\delta)}}{\epsilon(1-\eta)}. \tag{35}$$

Combining Equations (35) and (33), we have the result. □

D.5. Multi-response Linear Regression

The question of multi-response linear regression problem is as follows: given two matrices $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times p}$ as input, multi-response linear regression is defined as the following minimization problem:

$$\operatorname*{argmin}_{X\in\mathbb{R}^{d\times p}} \|AX - B\|_F^2 .$$

This problem is also known as *generalized linear regression*, and is used in the analysis of low-rank approximation (see Woodruff (2014) and references therein). We show the following:

Theorem 18 (Linear regression)**.** *Given privacy parameters (ϵ,δ) and approximation parameter $\eta \in (0,1/2)$, let $A_W \in \mathbb{R}^{W\times d}$ and $B \in \mathbb{R}^{W\times p}$ be the matrix streamed during the window of size W formed as defined in equation (1). Then we can efficiently output a matrix $\widetilde{X} \in \mathbb{R}^{d\times p}$ while preserving (ϵ,δ)-differential privacy such that*

$$\left\|A_W\widetilde{X} - B_W\right\|_F^2 \leq \left(\frac{1}{1-\eta}\right)\min_{X\in\mathbb{R}^{d\times p}} \|A_WX - B\|_F^2 + \frac{c(\tau+p)^2\log(\tau+p)\log(W)}{\eta\epsilon}$$

for some large constant $c > 0$ and $\tau = d + \frac{14}{\epsilon^2}\log(4/\delta)$.

Proof. To reduce the notation overhead, we use $A = A_W$ and $B = B_W$ to denote the matrices streamed in the last W updates. We use $\begin{pmatrix} A & B\end{pmatrix} \in \mathbb{R}^{W\times(d+p)}$ as the matrix that is being streamed and assume $C^\top C \in \mathbb{R}^{(d+p)\times(d+p)}$ as an output. We solve the following minimization problem:

$$\text{minimize:} \quad \mathsf{Tr}\left(\begin{pmatrix} X^\top & -\mathbb{1}_p^\top\end{pmatrix} C^\top C \begin{pmatrix} X \\ -\mathbb{1}_p\end{pmatrix}\right)$$

where $X \in \mathbb{R}^{d\times p}$. Let $\widetilde{X}$ be an optimal solution to the above minimization problem. In particular, this implies that for all $X \in \mathbb{R}^{d\times p}$,

$$\mathsf{Tr}\left(\begin{pmatrix} \widetilde{X}^\top & -\mathbb{1}_p\end{pmatrix} C^\top C \begin{pmatrix} \widetilde{X} \\ -\mathbb{1}_p\end{pmatrix}\right) \leq \mathsf{Tr}\left(\begin{pmatrix} X^\top & -\mathbb{1}_p\end{pmatrix} C^\top C \begin{pmatrix} X \\ -\mathbb{1}_p\end{pmatrix}\right).$$

Let

$$\widehat{X} := \operatorname*{argmin}_{X\in\mathbb{R}^{d\times p}} \|A_WX - B_W\|_F^2$$

be an optimal solution for the original regression problem. It follows that for all $X \in \mathbb{R}^{d\times p}$,

$$\left\|A_W\widehat{X} - B_W\right\|_F^2 \leq \|A_WX - B_W\|_F^2 . \tag{36}$$

The right side of Equation (8) gives us

$$\begin{aligned}\left\|C\begin{pmatrix}\widetilde{X}\\ -\mathbb{1}_p\end{pmatrix}\right\|_F^2 &\leq \left\|C\begin{pmatrix}\widehat{X}\\ -\mathbb{1}_p\end{pmatrix}\right\|_F^2 \leq \frac{1}{(1-\eta)}\left\|\begin{pmatrix}A_W & B_W\end{pmatrix}\begin{pmatrix}\widehat{X}\\ -\mathbb{1}_p\end{pmatrix}\right\|_F^2 + \frac{c(\tau+p)^2\log(\tau+p)}{\epsilon}\\ &= \frac{1}{1-\eta}\left\|A_W\widehat{X} - B_W\right\|_F^2 + \frac{c(\tau+p)^2\log(\tau+p)}{\epsilon}\\ &= \frac{1}{1-\eta}\min_{X\in\mathbb{R}^{d\times p}}\|A_WX - B_W\|_F^2 + \frac{c(\tau+p)^2\log(\tau+p)}{\epsilon},\end{aligned} \tag{37}$$

where the second equality is by the definition of $\widehat{X}$. Similarly, the left hand side inequality of Equation (8) gives us a lower bound as follows:

$$\left\|C\begin{pmatrix}\widetilde{X}\\ -\mathbb{1}_p\end{pmatrix}\right\|_F^2 \geq \left\|\begin{pmatrix}A_W & B_W\end{pmatrix}\begin{pmatrix}\widetilde{X}\\ -\mathbb{1}_p\end{pmatrix}\right\|_F^2 - \frac{c(\tau+p)^2\log(\tau+p)}{\epsilon}$$
$$= \left\|A_W\widetilde{X} - B_W\right\|_F^2 - \frac{c(\tau+p)^2\log(\tau+p)\log(W)}{\eta\epsilon}. \tag{38}$$

Combining Equation (37) and (38) gives us the claimed result. □

E. Lower Bounds for Low-rank Approximation

This section is devoted to proving a lower bound on the space requirement for low-rank factorization with non-trivial additive error. It is well known that no private algorithm (not necessarily differentially private) incurs an additive error $o(\sqrt{kd})$ (Hardt & Roth, 2012) due to linear reconstruction attack. On the other hand, the only known space lower bound of Upadhyay (2018) holds for streaming data where the entire historic data is considered important. While the entries can be streamed in an arbitrary order, this paper considers the case when one row is streamed at a time. Hence, there might be a possibility to construct an improved space algorithm for the special case of streaming we consider. However, we show below that for any non-trivial values of τ, this is not the case.

We first note that the technique developed by Bar-Yossef (2002) can be used to give lower bounds on the number of rows to be sampled by any sampling-based algorithm for low-rank matrix approximation. However, space lower bounds, in general, is a harder problem as one can use methods other than row sampling. For example, Bar-Yossef (2002) showed that any sampling-based algorithm for computing Euclidean norm of a stream of length W requires $\Omega(W)$ samples, while Upadhyay (2019) gave a privacy-preserving sliding window algorithm using $O\left(\frac{\sqrt{W}\log^2 W}{\eta^2}\right)$ bits. Our lower bounds come from reduction from the communication complexity of AIND problem.

Theorem 19. *Let $n, d, k \in \mathbb{N}$ and $\eta > 0$. Then the space used by any randomized single-pass algorithm for low-rank approximation in the sliding window model is at least $\Omega(Wk\log(W)/\eta)$.*

Proof. For a matrix A and set of indices C, we use the notation $A(C)$ to denote the submatrix formed by the columns indexed by C. We use the standard extension of the proof of Upadhyay (2018) for the sliding window model. The idea is basically for Alice to generate a stream with heavier weights on the more recent rows. Then Bob simply discards the stream not in the last W updates and use the rest of the state to compute the value of x_{ind} as in the case of Upadhyay (2018). Let $\ell = \frac{\log W}{\eta}$. Suppose $n \geq d$ and let $a = \frac{k\ell}{20\eta}$. Without loss of generality, we can assume that a is at most $d/2$. We assume Alice has a string $x \in \{-1,+1\}^{(W-a)a}$ and Bob has an index $\mathsf{ind} \in [(W-a)a]$. The idea is to define the matrix A with high Frobenius norm. The matrix A is the summation of the matrix $\widetilde{A}$ constructed by Alice and $\bar{A}$ constructed by Bob. We first define how Alice and Bob construct the instant $A = \widetilde{A} + \bar{A}$. Alice constructs its matrix $\widetilde{A}$ as follows.

1. Alice partitions the set $\{1,\cdots,a\}$ in to ℓ disjoint sets $I_1,\cdots,I_\ell$ such that $I_i := \{(i-1)a/\ell + 1, \cdots ia/\ell\}$.
2. Let $M(I_i)$ be an $(W-a)\times\frac{a}{\ell}$ matrix for all $1 \leq i \leq \ell$. Alice forms a bijection between entries of x and the entries of M in the following manner. Every entry of $M(I_i)$ is defined by a unique bit of x, i.e.,
$$M(I_i)_{j,k} = (-1)^{x_p}(10)^i, \quad p = \frac{(i-1)(W-a)a}{\ell} + (k-1)(n-a) + j.$$
3. Let $M = \begin{pmatrix}M_{I_1} & \cdots & M_{I_\ell}\end{pmatrix}$. The matrix $\widetilde{A}$ is now defined as follows.
$$\widetilde{A} = \begin{pmatrix}0^{a\times a} & 0^{a\times(d-a)}\\ M & 0^{(n-a)\times(d-a)}\end{pmatrix}.$$

Suppose Bob is given an index $\mathsf{ind} \in [(W-a)a]$ such that x_{ind} corresponds to the sub-matrix $M(I_\theta)$ for some $1 \leq \theta \leq \ell$. Then we can assume that Bob also knows every entry in the sub-matrix $M(I_{\theta'})$ for $\theta' > \theta$. The idea is that Bob inserts a

scaled identity matrix in the stream, where the scaling parameter γ is large enough to make sure that most of the error of any randomized algorithm is due to other columns of A. As we shall see later, we set the value of γ as a large polynomial in the approximation error of the algorithm. Bob forms his matrix as follows:

1. Bob forms a second level partition of the columns of $M(I_\theta)$ into equal size groups $G_1, \cdots, G_{a/k\ell}$. There exists a unique r such that x_{ind} maps to an entry in the sub-matrix formed by columns indexed by one of the second level partition G_r.
2. Let $C = \{c, c+1, \cdots, c+k-1\}$ be the columns corresponding to the group of I_θ.
3. Bob expires the stream of Alice except for the current window in a matrix $\bar{A}$ which is an all-zero matrix, except for entries $\bar{A}_{c+i,c+i} = \gamma$ for $0 \le i \le k-1$ and γ to be chosen later.

Let $\mathfrak{M}$ be the algorithm that computes low-rank approximation under the turnstile model. Alice feeds its matrix $\widetilde{A}$ to $\mathfrak{M}$ in the turnstile manner and send the state of the algorithm by the end of her feed to Bob. Bob uses the state received by Alice and feed the algorithm $\mathfrak{M}$ with its own matrix $\bar{A}$ in a turnstile manner. Therefore, the algorithm $\mathfrak{M}$ gets as input a matrix $A = \widetilde{A} + \bar{A}$ and it is required to output a rank-k matrix B with additive error $\tau = O(W+d)$. We will show that any such output allows us to solve AIND. Denote by $A(C)$ the sub-matrix formed by the columns $C := \{c, c+1, \cdots, c+k-1\}$.

Let us first understand the properties of the constructed matrix A. To compute the Frobenius norm of this matrix, we need to consider two cases: the case for sub-matrices in which ind belongs, i.e, $M(I_r)$, and the rest of the matrix. For the sub-matrix corresponding to the columns indexed by C, the columns of $A(I_\theta)$ have Euclidean length $(\gamma^2 + (n-a)100^\theta)^{1/2}$. For $\theta' < \theta$, every columns have Euclidean norm $(a(n-a))^{1/2}10^{\theta'}$. Therefore, we have the following:

$$\begin{aligned}\|A - [A]_k\|_F^2 &\le \frac{((a-k)(W-a)100^\theta}{\ell} + \sum_{\theta'<\theta} \frac{a(W-a)100^{\theta'}}{\ell} \\ &\le \frac{((a-k)(W-a)100^\theta}{\ell} + \frac{a(W-a)100^\theta}{99\ell} \le 2\cdot(100)^\theta Wd/\ell = \tau\end{aligned}$$

In order to solve low-rank approximation, the algorithm needs to output a matrix B of rank at most k such that, with probability $5/6$ over its random coins,

$$\begin{aligned}\|A-B\|_F^2 &\le \left[(1+\eta)\sqrt{\tau}+\tau\right]^2 \le 2(1+\eta)\tau + 2\tau^2 \\ &\le 2\tau + 100^\theta k(W-a)\left(\frac{1}{10}+\frac{1}{99}\right) + 2\tau^2 \le 4\cdot(100)^\theta Wd/\ell + \frac{100^\theta k(n-a)}{5} + 2\tau^2\end{aligned}$$

Let $\Psi := 4\cdot(100)^\theta Wd/\ell + 100^\theta k(W-a)\left(\frac{1}{10}+\frac{1}{99}\right) + 2\tau^2$. The proof idea is now to show the following: (i) columns of B corresponding to index set in C are linearly independent, and (ii) bound the error incurred by $\|A-B\|_F$ in terms of the columns indexed by G_r.

The idea is to show that most of the error is due to the other columns in B; and therefore, sign in the submatrix $A(C)$ agrees with that of the signs of those in the submatrix $B(C)$. This allows Bob to solve the AIND problem as Bob can just output the sign of the corresponding position. Let $R := \{ra/k+1, \cdots, (r+1)a/k\}$ and $C := \{c, \cdots, c+k-1\}$. Let Y be the submatrix of B formed by the rows indexed by R and columns indexed by C. The following lemma proves that when γ is large enough, then the columns of B corresponding to index set C are linearly independent.

Lemma 12. *Let $B(C) := [B_{:c} \quad \cdots B_{:c+k-1}]$ be the columns corresponding to the sub-matrix formed by columns $c, \cdots, c+k-1$ of B. If $\gamma \ge 2\Psi^2$, then the column space of $B(C)$ is the same as that of $[A]_k$.*

Proof. We will prove the lemma by considering the $k \times k$ sub-matrix, say Y. Recall that Y is a submatrix of B formed by the rows indexed by R and the columns indexed by C. For the sake of brevity and abuse of notation, let us denote the restriction of B to this sub-matrix $Y := [Y_{:1}, \cdots, Y_{:k}]$. In what follows, we prove a stronger claim that the submatrix Y is a rank-k matrix.

Suppose, that the vectors $\{Y_{:1}, \cdots, Y_{:k}\}$ are linearly dependent. In other words, there exists a vector $Y_{:i}$ and real numbers $a_1, \cdots, a_k$, not all of which are identically zero, such that $Y_{:i} = \sum_{j=1, j\neq i}^{k} a_j Y_{:j}$.

From the construction, since Bob inserts a sub-matrix $\gamma \mathbb{1}_k$, we know that

$$\sum_{j=1}^{k}(Y_{j,j}-\gamma)^2 \leq \|A-B\|_F^2 \leq \Psi \quad \text{and} \quad \sum_{j=1}^{k}\sum_{p\neq j} Y_{p,j}^2 \leq \|A-B\|_F^2 \leq \Psi. \tag{39}$$

From equation (39) and choice of γ, for all j, we have $Y_{j,j} \geq \Psi^2$ and $Y_{p,j} \leq \sqrt{\Psi}$. We thus have

$$Y_{i,i} = \sum_{j=1, j\neq i}^{k} a_j Y_{i,j} \geq \Psi^2$$

This imply that there is an $p \in \{1,\cdots,k\}\setminus\{i\}$ such that $|a_p| \geq \frac{\Psi^2}{k\sqrt{\Psi}}$.

Let $\widehat{i}$ be the index in $\{1,\cdots,k\}\setminus\{i\}$ for which $|a_{\widehat{i}}|$ attains the maximum value. We have $|a_{\widehat{i}}Y_{\widehat{i},\widehat{i}}| \geq |a_{\widehat{i}}|\Psi^2$ and $|a_j Y_{\widehat{i},j}| \leq |a_{\widehat{i}}|\sqrt{\Psi}$. Now consider the $\widehat{i}$-entry of $Y_{:i}$. Note that $\widehat{i} \neq i$. Since Ψ depends quadratically on m and τ, we have

$$\left|\sum_{j=1, j\neq i}^{k} a_j Y_{\widehat{i},j}\right| \geq |a|(\Psi^2 - k\sqrt{\Psi}) \geq (\Psi^2 - k\sqrt{\Psi})\frac{\Psi^2}{k\sqrt{\Psi}} > \sqrt{\Psi}.$$

This is a contradiction because for $p \neq j$, $Y_{p,j} \leq \sqrt{\Psi}$ (equation (39)). This finishes the proof. □

For the sake of brevity, let $V_{:1},\cdots,V_{:k}$ be the columns of $B(C)$ and $\widetilde{V}_{:1},\cdots,\widetilde{V}_{:k}$ be the restriction of these column vectors to the rows $a+1,\cdots,m$. In other words, vectors $\widetilde{V}_{:1},\cdots,\widetilde{V}_{:k}$ are the column vectors corresponding to the columns in M. We showed in Lemma 12 that the columns $B(C)$ spans the column space of B. We can assume that the last $n-a$ columns of B are all zero vectors because B is a rank-k matrix. We can also assume without any loss of generality that, except for the entries in the row indexed by R, all the other entries of $B(C)$ are zero. This is because we have shown that the submatrix of $B(C)$ formed by rows indexed by R and columns indexed by C have rank k.

Now any row i of B can be therefore represented as $\sum \eta_{i,j}V_{:j}$, for real numbers $\eta_{i,j}$, not all of which are identically zero. The following lemma proves part (ii) of our proof idea.

Lemma 13. *Let $V_{:1},\cdots,V_{:k}$ be as above. Then column i of B can be written as linear combination of real numbers $\eta_{i,1},\cdots\eta_{i,k}$ of the vectors $V_{:1},\cdots,V_{:k}$ such that, for all j and $i \in R$, $\eta_{i,j}^2 \leq 4/\Psi^3$.*

Proof. Let $M_{:1},\cdots M_{:a}$ be the columns of M, where M is the $(W-a)\times a$ submatrix of the matrix $\widetilde{A}$ corresponding to the input of Alice. We have

$$\Psi \geq \|A-B\|_F^2 \sum_{i=1}^{k}(\gamma - V_{r(a/k)+i,i})^2 + \sum_{i=1}^{k}\sum_{j\neq i} V_{r(a/k)+i,j}^2 + \sum_{i=1}^{k}\|M_{:r(a/k)+i} - \widetilde{V}_{:i}\|^2$$
$$+ \sum_{i\notin R}\sum_{j=1}^{k}\left(\eta_{i,j}V_{ra/k+j,j} + \sum_{j'\neq j}\eta_{i,j'}V_{ra/k+j,j'}\right)^2 + \sum_{i\notin R}\left\|M_{:i} - \sum_{j=1}^{k}\eta_{i,j}\widetilde{V}_{:j}\right\|^2.$$

As before, we have $|V_{r(a/k)+i,j}^2| \leq \sqrt{\Psi}$ and $|V_{r(a/k)+i,i}| \geq \Psi^2$. Let j_i be the index such that $|\eta_{i,j_i}|$ is the maximum. Then the above expression is at least $|\eta_{i,j_i}|^2(\Psi^2 - k\sqrt{\Psi})^2 \geq |\eta_{i,j_i}|^2\Psi^4/4$. Since this is less than Ψ, the result follows from the definition of j_i. □

We can now complete the proof. First note that since M is a signed matrix, each $\widetilde{V}_i$ in the third term of the above expression is at least $\sqrt{\Psi}$. Therefore, for all $i \notin S$ and all j

$$\left|\sum_{j=1}^{k}\eta_{i,j}\widetilde{V}_{:j}\right| \leq \frac{4k\Psi^{1/2}}{\Psi^{3/2}} = \frac{4k}{\Psi}.$$

As $M_{:i}$ is a sign vector and if $\tau = O(m+n) = O(m)$, this implies that

$$\sum_{i \notin R} \left\| M_{:i} - \sum_{j=1}^{k} \eta_{i,j} \widetilde{V}_{:j} \right\|^2 \geq \sum_{i \notin R} \|M_{:i}\|^2 \left(1 - \frac{4k}{\Psi}\right) \geq O((100)^\theta W d/\ell) - O(100^\theta a)$$

$$\sum_{i=1}^{k} \left\| M_{:r(a/k)+i} - \widetilde{V}_{:i} \right\|^2 = \sum_{i=1}^{k} \sum_{j=1}^{W-a} (M_{j,r(a/k)+i} - (\widetilde{V}_i)_j)^2 \leq \frac{100^\theta k(W-a)}{5} + O(100^\theta a)$$

Since there are $k(n-a)$ entries in the submatrix formed by the columns indexed by C, at least $1 - \left(\frac{1}{10} + \frac{1}{99} + o(1)\right)$ fraction of the entries have the property that the sign of $M_{j,ra/k+i}$ matches the sign of $\widetilde{V}_{j,i}$. Since ind is in one of the columns of $M_{:ra/k+1}, \cdots M_{:ra/k+k}$, with probability at least $1 - \left(\frac{1}{10} + \frac{1}{99} + o(1)\right)$, if Bob outputs the sign of the corresponding entry in B, then Bob succeeds in solving AIND. This gives a lower bound of $\Omega((W-a)a) = \Omega(Wk\ell/\eta)$ space. □

F. Extension to continual release

Until now, we consider only one-shot algorithm, that is, an algorithm to compute spectral approximation with additive error of $O\left(\frac{cr\log^2(1/\delta)}{\epsilon^2}\right)\mathbb{1}_d$, but the output is produced just once. If we naively use this algorithm to publish a matrix continually over the entire window, it would lead to a total accuracy loss of $O\left(\frac{crW\log^2(1/\delta)}{\epsilon^2}\right)\mathbb{1}_d$. In this section, we show an algorithm that computes spectral approximation with small additive error over the entire window, i.e., $o(W\tau)$.

The continual release model was proposed by Dwork et al. (Dwork et al., 2010). In contrast to our setting, continual release model consider the entire data useful and does not put any space constraints. We provide two different protocols, in both of which we consider accuracy for only the update that came during the current window.

The first approach uses the same binary tree method introduced by (Bentley & Saxe, 1980) and used in (Dwork et al., 2010) and (Chan et al., 2011), and in the sliding window model by (Bolot et al., 2013) and (Upadhyay, 2019). However, we depart from their technique in the sense that we only build the binary tree. Let $a_{T-W+1}, \cdots, a_T$ be the updates at any time T. In particular, we construct a binary tree as follows:

1. Every leaves consists of a single update privatized using Step 3.
2. For every other node, n, other than the leaf nodes, let C be the set of updates on the leaves of the subtree of n. Then we first compute
$$S_{\mathsf{n}} = \sum_{a_i \in C} a_i^\top a_i$$
Then we store $\widetilde{S}_{\mathsf{n}}$ on the node n, where $\widetilde{S}_{\mathsf{n}}$ is formed using the privitization step (Step 3) on S_{n}.

This construction mimics the construction of (Dwork et al., 2010) and hence using their analysis, we get the following result:

Theorem 20 (Private spectral approximation under sliding window)**.** *Given the privacy parameter ϵ, window size W, approximation parameter β, let $\Omega = (a_t)_{t>0}$ be the stream such that $a_t \in \mathbb{R}^d$. For every $t > 0$, define $A_W(t)$ to be the matrix formed at time t by the last W updates. Then* SLIDING-PRIV$(\Omega; (\epsilon, \delta); W)$ *is (ϵ, δ)-differential private and requires $O\left(\frac{d^2 W}{\eta}\log W\right)$ space. Further the mapping $\widetilde{C} \leftarrow$*SLIDING-PRIV$(\Omega; (\epsilon, \delta); W)$ *satisfies the following:*

$$\mathsf{Pr}\left[A_W^\top A_W - (c\tau\log\tau)\mathbb{1}_d \preceq \widetilde{C} \preceq A_W^\top A_W + (C\tau\log\tau)\mathbb{1}_d\right] \geq 1 - \frac{1}{\mathrm{poly}(d)}$$

for constants $c, C > 1$ and $\tau := \left(d + \frac{14\log(1/\delta)}{\epsilon^2}\right)\log^{3/2}(W)$.

Note that this result uses η-approximate histogram property.

Making space requirement sublinear in window-size at the cost of accuracy loss. We now improve this bound by incurring an accuracy loss that scales only logarithmic in the window size instead of linear. For this, we borrow the idea of

(Bentley & Saxe, 1980) to move from one-shot algorithms to continually release algorithm. This technique was also used in (Dwork et al., 2010) and subsequently improved in (Chan et al., 2011) and (Bolot et al., 2013). The idea is to build binary tree with leaves being the graphs at the checkpoint. For this, we fix some notation:

1. $\widetilde{\mathfrak{B}}$ be the binary tree formed by the leaves $\widetilde{K}(1), \cdots, \widetilde{K}(\ell)$.
2. $\widetilde{\mathfrak{B}}_{\mathsf{n}}$ be the subtree of the internal node, n, of the tree $\widetilde{\mathfrak{B}}$.
3. $\widetilde{\mathfrak{L}}_{\mathsf{n}}$ be the leaves of $\widetilde{\mathfrak{B}}$ in the subtree $\widetilde{\mathfrak{B}}_{\mathsf{n}}$; i.e., a subset of the graphs $\widetilde{K}(1), \cdots, \widetilde{K}(\ell)$.

We divide our window in to $\sqrt{W}$ sub-windows, each of size $\sqrt{W}$. We run an instantiation of our algorithm for each of these subwindows. Let these subwindows terminates at timestamps $T_1, T_2, \cdots, T_{\sqrt{W}} = T$. For j-th subwindow that terminates at time T_j, we also augment our data structure $\mathsf{DS}_{\mathsf{priv}}$ for each of these windows to contains the following:

1. A set of covariance matrix for every timestamps stored in the data structure in Section B. That is, for timestamps, $t_1, \cdots, t_\ell$, apart from the privatized covariance matrix, $\widetilde{K}(1), \cdots, \widetilde{K}(\ell)$ we also store $K(i)$ such that

$$K(i) = \sum_{t=t_i}^{T_j} a_t^\top a_t \quad \text{for all } 1 \le i \le \ell.$$

2. A binary tree formed using an algorithm BINARY-TREE that uses covariance matrices $K(1), \cdots, K(\ell)$ and $\widetilde{K}(1), \cdots, \widetilde{K}(\ell)$. BINARY-TREE operates as follows:
   (a) The leaves of the tree are $\widetilde{K}(1), \cdots, \widetilde{K}(\ell)$.
   (b) For every internal node, n, let $\mathfrak{L}_{\mathsf{n}}$ be the covariance matrix from the set $\{K(1), \cdots, K(\ell)\}$ corresponding to the covariance matrices in the set $\widetilde{\mathfrak{L}}_{\mathsf{n}}$. Then the covariance matrix stored in the node n is the privatization of the following covariance matrix:

$$K_{\mathsf{n}} = \sum_{\widetilde{K}(i) \in \widetilde{\mathfrak{L}}_n} K(i),$$

   where the privatization is done as in Step 3.

3. Delete all the internal nodes whose leaves contains covariance matrix is formed before time t_1.

Since the number of checkpoints is $\ell = O\left(\frac{n}{\rho} \log W\right)$, combining Theorem 10 with that of (Dwork et al., 2010), we have the following theorem:

Theorem 21 (Private spectral approximation under continual release for a sliding window)**.** *Given the privacy parameter (ϵ, δ), window size W, approximation parameter β, let $\Omega = (a_t)_{t>0}$ be the stream such that $a_t \in \mathbb{R}^d$. For every $t > 0$, define $A_W(t)$ to be the matrix formed at time t by the last W updates. Then* SLIDING-PRIV$(\Omega; (\epsilon, \delta); W)$ *is (ϵ, δ)-differential private and requires $O\left(\frac{d^3\sqrt{W}}{\eta} \log W\right)$. Further, the mapping $\widetilde{C} \leftarrow$*SLIDING-PRIV$(\Omega; (\epsilon, \delta); W)$ *satisfies the following:*

$$\mathsf{Pr}\left[\left(A_W^\top A_W - (c\tau \log \tau)\mathbb{1}_d\right) \preceq \widetilde{C} \preceq \left(\frac{A_W^\top A_W}{(1-\eta)} + (C\tau \log \tau)\mathbb{1}_d\right)\right] \ge 1 - \frac{1}{\mathrm{poly}(d)}$$

for constants $c, C > 1$ and $\tau := \left(d + \frac{14 \log(1/\delta)}{\epsilon^2}\right) W^{3/4}$.