Privately Learning Subspaces

Anonymous Author(s)
Affiliation
Address
email

Abstract

Private data analysis suffers a costly curse of dimensionality. However, the data often has an underlying low-dimensional structure. For example, when optimizing via gradient descent, the gradients often lie in or near a low-dimensional subspace. If that low-dimensional structure can be identified, then we can avoid paying (in terms of privacy or accuracy) for the high ambient dimension.

We present differentially private algorithms that take input data sampled from a low-dimensional linear subspace (possibly with a small amount of error) and output that subspace (or an approximation to it). These algorithms can serve as a pre-processing step for other procedures.

1 Introduction

Differentially private algorithms generally have a poor dependence on the dimensionality of their input. That is, their error or sample complexity grows polynomially with the dimension. For example, for the simple task of estimating the mean of a distribution supported on $[0,1]^d$, we have per-coordinate error $\Theta(\sqrt{d}/n)$ to attain differential privacy, where n is the number of samples. In contrast, the non-private error is $\Theta(\sqrt{\log(d)/n})$.

This cost of dimensionality is inherent [BUV14; SU17; DSSUV15]. *Any* method with lower error is susceptible to tracing attacks (a.k.a. membership inference attacks). However, these lower bounds only apply when the data distribution is "high-entropy." This leaves open the posssibility that we can circumvent the curse of dimensionality when the data has an underlying low-dimensional structure.

Data often does possess an underlying low-dimensional structure. For example, the gradients that arise in deep learning tend to be close to a low-dimensional subspace [ACGMMTZ16; LXTSG17; GARD18; LFLY18; LGZCB20; ZWB20; FT20]. Low dimensionality can arise from meaningful relationships that are at least locally linear, such as income versus tax paid. It can also arise because we are looking at a function of data with relatively few attributes.

A long line of work [BLR08; HT10; HR10; Ull15; BBNS19; BCMNUW20; ZWB20; KRRT20, etc.] has shown how to exploit structure in the data to attain better privacy and accuracy. However, these approaches assume that this structure is known *a priori* or that it can be learned from non-private sources. This raises the question:

> Can we learn low-dimensional structure from the data subject to differential privacy?

We consider the simple setting where the data lies in $\mathbb{R}^d$ but is in, or very close to a linear subspace, of dimension k. We focus on the setting where $k \ll d$ and we develop algorithms whose sample complexity does not depend on the ambient dimension d; a polynomial dependence on the true dimension k is unavoidable.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Our algorithms identify the subspace in question or, if the data is perturbed slightly, an approximation to it. Identifying the subspace structure is interesting in its own right, but it also can be used as a pre-processing step for further analysis – by projecting to the low-dimensional subspace, we ensure subsequent data analysis steps do not need to deal with high-dimensional data.

1.1 Our Contributions: Privately Learning Subspaces – Exact Case

We first consider the exact case, where the data $X_1, \cdots, X_n \in \mathbb{R}^d$ are assumed to lie in a k-dimensional subspace (rather than merely being near to it) – i.e., $\mathsf{rank}\,(A) = k$, where $A = \sum_i^n X_i X_i^T \in \mathbb{R}^{d \times d}$. In this case, we can also recover the subspace exactly.

However, we must also make some non-degeneracy assumptions. We want to avoid a pathological input dataset such as the following. Suppose $X_1, \cdots, X_k$ are linearly independent, but $X_k = X_{k+1} = X_{k+2} = \cdots = X_n$. While we can easily reveal the repeated data point, we cannot reveal anything about the other points due to the privacy constraint.

A natural non-degeneracy assumption would be to assume that the data points are in "general position" – that is, that there are no non-trivial linear dependencies among the data points. This means that *every* set of k data points spans the subspace or, equivalently, no subspace of dimension $k-1$ contains more than $k-1$ data points. This is a very natural assumption – if the data consists of n samples from a continuous distribution on the subspace, then this holds with probability 1. We relax this assumption slightly and assume that no subspace of dimension $k-1$ contains more than ℓ data points. We also assume that all points are non-zero. Note that we define subspaces to pass through the origin; our results can easily be extended to affine subspaces.

Theorem 1.1 (Main Result – Exact Case)**.** *For all $n, d, k, \ell \in \mathbb{N}$ and $\varepsilon, \delta > 0$ satisfying $n \geq O\left(\ell + \frac{\log(1/\delta)}{\varepsilon}\right)$, there exists a randomized algorithm $M : \mathbb{R}^{d \times n} \to \mathcal{S}_d^k$ satisfying the following. Here $\mathcal{S}_d^k$ denotes the set of all k-dimensional subspaces of $\mathbb{R}^d$.*

- *M is (ε, δ)-differentially private with respect to changing one column of its input.*
- *Let $X = (X_1, \cdots, X_n) \in \mathbb{R}^{d \times n}$. Suppose there exists a k-dimensional subspace $S_* \in \mathcal{S}_d^k$ that contains all but ℓ of the points – i.e., $|\{i \in [n] : X_i \in S_*\}| \geq n - \ell$. Further suppose that any $(k-1)$-dimensional subspace contains at most ℓ points – i.e., for all $S \in \mathcal{S}_d^{k-1}$, we have $|\{i \in [n] : X_i \in S\}| \leq \ell$. Then $\mathbb{P}\left[M(X) = S_*\right] = 1$.*

The parameter ℓ in Theorem 1.1 can be thought of as a robustness parameter. Ideally the data points are in general position, in which case $\ell = k - 1$. If a few points are corrupted, then we increase ℓ accordingly; our algorithm can tolerate the corruption of a small constant fraction of the data points. Theorem 1.1 is optimal in the sense that $n \geq \Omega\left(\ell + \frac{\log(1/\delta)}{\varepsilon}\right)$ samples are required.

1.2 Our Contributions: Privately Learning Subspaces – Approximate Case

Next we turn to the substantially more challenging approximate case, where the data $X_1, \cdots, X_n \in \mathbb{R}^d$ are assumed to be close to a k-dimensional subspace, but are not assumed to be contained within that subspace. Our algorithm for the exact case is robust to changing a few points, but very brittle if we change all the points by a little bit. Tiny perturbations of the data points (due to numerical errors or measurement imprecision) could push the point outside the subspace, which would cause the algorithm to fail. Thus it is important to for us to cover the approximate case and our algorithm for the approximate is entirely different from our algorithm for the exact case.

The approximate case requires us to precisely quantify how close the input data and our output are to the subspace and we also need to make quantitative non-degeneracy assumptions. It is easiest to formulate this via a distributional assumption. We will assume that the data comes from a Gaussian distribution where the covariance matrix has a certain eigenvalue gap. This is a strong assumption and we emphasize that this is only for ease of presentation; our algorithm works under weaker assumptions. Furthermore, we stress that the differential privacy guarantee is worst-case and does not depend on any distributional assumptions.

We assume that the data is drawn from a multivariate Gaussian $\mathcal{N}(0, \Sigma)$. Let $\lambda_1(\Sigma) \geq \lambda_2(\Sigma) \geq \cdots \geq \lambda_d(\Sigma)$ be the eigenvalues of $\Sigma \in \mathbb{R}^{d \times d}$. We assume that there are k large eigenvalues $\lambda_1(\Sigma), \cdots, \lambda_k(\Sigma)$ – these represent the "signal" we want – and $d-k$ small eigenvalues $\lambda_{k+1}(\Sigma), \cdots, \lambda_d(\Sigma)$ – these are the "noise". Our goal is to recover the subspace spanned by the eigenvectors corresponding to the k largest eigenvalues $\lambda_1(\Sigma), \cdots, \lambda_k(\Sigma)$. Our assumption is that there is a large *multiplicative* gap between the large and small eigenvalues. Namely, we assume $\frac{\lambda_{k+1}(\Sigma)}{\lambda_k(\Sigma)} \le \frac{1}{\mathsf{poly}(d)}$.

Theorem 1.2 (Main Result – Approximate Case)**.** *For all $n, d, k \in \mathbb{N}$ and $\alpha, \gamma, \varepsilon, \delta > 0$ satisfying*

$$n \ge \Theta\left(\frac{k \log(1/\delta)}{\varepsilon} + \frac{\ln(1/\delta)\ln(\ln(1/\delta)/\varepsilon)}{\varepsilon}\right) \text{ and } \gamma^2 \le \Theta\left(\frac{\varepsilon\alpha^2 n}{d^2 k^3 \log(1/\delta)} \cdot \min\left\{\frac{1}{k}, \frac{1}{\log(k\log(1/\delta)/\varepsilon)}\right\}\right),$$

there exists an algorithm $M : \mathbb{R}^{d\times n} \to \mathcal{S}_d^k$ satisfying the following. Here $\mathcal{S}_d^k$ is the set of all k-dimensional subspaces of $\mathbb{R}^d$ represented as projection matricies – i.e., $\mathcal{S}_d^k = \{\Pi \in \mathbb{R}^{d\times d} : \Pi^2 = \Pi = \Pi^T, \mathsf{rank}(\Pi) = k\}$.

- *M is (ε, δ)-differentially private with respect to changing one column of its input.*
- *Let $X_1, \cdots, X_n$ be independent samples from $\mathcal{N}(0, \Sigma)$. Let $\lambda_1(\Sigma) \ge \lambda_2(\Sigma) \ge \cdots \ge \lambda_d(\Sigma)$ be the eigenvalues of $\Sigma \in \mathbb{R}^{d\times d}$. Suppose $\lambda_{k+1}(\Sigma) \le \gamma^2 \cdot \lambda_k(\Sigma)$. Let $\Pi \in \mathcal{S}_d^k$ be the projection matrix onto the subspace spanned by the eigenvectors corresponding to the k largest eigenvalues of Σ. Then $\mathbb{P}[\|M(X) - \Pi\| \le \alpha] \ge 0.7$.*

The sample complexity of our algorithm $n = O(k\log(1/\delta)/\varepsilon)$ is independent of the ambient dimension d; this is ideal. We can also boost the accuracy guarantees at a small (dimension independent) cost in sample complexity, as shown in Section E. However, there is a polynomial dependence on d in γ, which controls the multiplicative eigenvalue gap. This multiplicative eigenvalue gap is a strong assumption, but it is also a necessary assumption if we want the sample complexity n to be independent of the dimension d. In fact, it is necessary *even without the differential privacy constraint* [CZ16]. That is, if we did not assume an eigenvalue gap that depends polynomially on the ambient dimension d, then it would be impossible to estimate the subspace with sample complexity n that is independent of the ambient dimension d even in the non-private setting.

Our algorithm is based on the subsample and aggregate framework [NRS07] and a differentially private histogram algorithm. These methods are generally quite robust and thus our algorithm is, too. For example, our algorithm can tolerate $o(n/k)$ input points being corrupted arbitrarily. We also believe that our algorithm's utility guarantee is robust to relaxing the Gaussianity assumption. All that we require in the analysis is that the empirical covariance matrix of a few samples from the distribution is sufficiently close to its expectation Σ with high probability.

2 Related Work

To the best of our knowledge, the problem of privately learning subspaces, as we formulate it, has not been studied before. However, a closely-related line of work is on Private Principal Component Analysis (PCA) and low-rank approximations. We briefly discuss this extensive line of work below, but first we note that, in our setting, all of these techniques have a sample complexity n that grows polynomially with the ambient dimension d. Thus, they do not evade privacy's curse of dimensionality. However, we make a stronger assumption than these prior works – namely, we assume a large multiplicative eigenvalue gap. (Many of the prior works consider an *additive* eigenvalue gap, which is a weaker assumption.)

There has been a lot of interest in Private PCA, matrix completion, and low-rank approximation. One motivation for this is the infamous Netflix prize, which can be interpreted as a matrix completion problem. The competition was cancelled after researchers showed that the public training data revealed the private movie viewing histories of many of Netflix's customers [NS06]. Thus privacy is a real concern for matrix analysis tasks.

Many variants of these problems have been considered: Some provide approximations to the data matrix $X = (X_1, \cdots, X_n) \in \mathbb{R}^{d\times n}$; others approximate the covariance matrix $A = \sum_i^n X_i X_i^T \in \mathbb{R}^{d\times d}$ (as we do). There are also different forms of approximation – we can either produce a subspace or an approximation to the entire matrix, and the approximation can be measured by different norms (we consider the operator norm between projection matrices). Importantly, we define differential privacy to allow one data point X_i to be changed arbitrarily, whereas most of the prior work assumes a bound on the norm of the change or even assumes that only one coordinate of one vector can be changed. In the discussion below we focus on the techniques that have been considered for these problems, rather than the specific results and settings.

Dwork, Talwar, Thakurta, and Zhang [DTTZ14] consider the simple algorithm which adds independent Gaussian noise to each of entries of the covariance matrix A, and then perform analysis on the noisy matrix. (In fact, this algorithm predates the development of differential privacy [BDMN05] and was also analyzed under differential privacy by McSherry and Mironov [MM09] and Chaudhuri, Sarwate, and Sinha [CSS12].) This simple algorithm is versatile and several bounds are provided for the accuracy of the noisy PCA. The downside of this is that a polynomial dependence on the ambient dimension d is inherent – indeed, they prove a sample complexity lower bound of $n = \tilde{\Omega}(\sqrt{d})$ for any algorithm that identifies a useful approximation to the top eigenvector of A. This lower bound does not contradict our results because the relevant inputs do not satisfy our near low-rank assumption.

Hardt and Roth [HR12] and Arora, Braverman, and Upadhyay [ABU18] apply techniques from dimensionality reduction to privately compute a low-rank approximation to the input matrix X. Hardt and Roth [HR13] and Hardt and Price [HP13] use the power iteration method with noise injected at each step to compute low-rank approximations to the input matrix X. In all of these, the underlying privacy mechanism is still noise addition and the results still require the sample complexity to grow polynomially with the ambient dimension to obtain interesting guarantees. (However, the results can be dimension-independent if we define differential privacy so that only one entry – as opposed to one column – of the matrix X can be changed by 1. This is a significantly weaker privacy guarantee.)

Blocki, Blum, Datta, and Sheffet [BBDS12] and Sheffet [She19] also use tools from dimensionality reduction; they approximate the covariance matrix A. However, they show that the dimensionality reduction step itself provides a privacy guarantee (whereas the aforementioned results did not exploit this and relied on noise added at a later stage). Sheffet [She19] analyzes two additional techniques – the addition of Wishart noise (i.e., YY^T where the columns of Y are independent multivariate Gaussians) and sampling from an inverse Wishart distribution (which has a Bayesian interpretation).

Chaudhuri, Sarwate, and Sinha [CSS12], Kapralov and Talwar [KT13], Wei, Sarwate, Corander, Hero, and Tarokh [WSCHT16], and Amin, Dick, Kulesza, Medina, and Vassilvitskii [ADKMV18] apply variants of the exponential mechanism [MT07] to privately select a low-rank approximation to the covariance matrix A. This method is nontrivial to implement and analyse, but it ultimately requires the sample complexity to grow polynomially in the ambient dimension.

Gonem and Gilad-Bachrach [GGB18] exploit smooth sensitivity [NRS07] to release a low-rank approximation to the matrix A. This allows them to add less noise than using worst case sensitivity, under an eigenvalue gap assumption. However, the sample complexity n remains polynomial in the dimension d.

2.1 Limitations of Prior Work

Given the great variety of techniques and analyses that have been applied to differentially private matrix analysis problems, what is missing? We see that almost all of these techniques are ultimately based on some form of noise addition or the exponential mechanism. With the singular exception of the techniques of Sheffet [She19], all of these prior techniques satisfy pure[1] or concentrated differential privacy [BS16]. This is enough to conclude that these techniques cannot yield the dimension-independent guarantees that we seek. No amount of postprocessing or careful analysis can avoid this limitation. This is because pure and concentrated differential privacy have strong group privacy properties, which means "packing" lower bounds [HT10] apply.

We briefly sketch why concentrated differential privacy is incompatible with dimension-independent guarantees. Let the input be $X_1 = X_2 = \cdots = X_n = \xi/\sqrt{d}$ for a uniformly random $\xi \in \{-1, +1\}^d$. That is, the input is one random point repeated n times. If M satisfies $O(1)$-concentrated differential privacy, then it satisfies the mutual information bound $I(M(X); X) \leq O(n^2)$ [BS16]. But, if M provides a meaningful approximation to X or $A = XX^T$, then we must be able to recover an approximation to ξ from its output, whence $I(M(X); X) \geq \Omega(d)$, as the entropy of X is d bits. This gives a lower bound of $n \geq \Omega(\sqrt{d})$, even though X and A have rank $k = 1$.

[1] Pure differential privacy (a.k.a. pointwise differential privacy) is (ε, δ)-differential privacy with $\delta = 0$.

The above example shows that, even under the strongest assumptions (i.e., the data lies exactly in a rank-1 subspace), any good approximation to the subspace, to the data matrix X, or to the covariance matrix $A = XX^T$ must require the sample complexity n to grow polynomially in the ambient dimension d if we restrict to techniques that satisfy concentrated differential privacy. Almost all of the prior work in this general area is subject to this restriction.

To avoid a sample complexity n that grows polynomially with the ambient dimension d, we need fundamentally new techniques.

3 Overview of Our Techniques

For the exact case, we construct a score function for subspaces that has low sensitivity, assigns high score to the correct subspace, and assigns a low score to all other subspaces. Then we can simply apply a GAP-MAX algorithm to privately select the correct subspace [BDRS18].

The GAP-MAX algorithm satisfies (ε, δ)-differential privacy and outputs the correct subspace as long as the gap between its score and that of any other subspace is larger than $O(\log(1/\delta)/\varepsilon)$. This works even though there are infinitely many subspaces to consider, which would not be possible under concentrated differential privacy.

The simplest score function would simply be the number of input points that the subspace contains. This assigns high score to the correct subspace, but it also assigns high score to any larger subspace that contains the correct subspace. To remedy this, we subtract from the score the number of points contained in a strictly smaller subspace. That is, the score of subspace S is the number of points in S minus the maximum over all subspaces $S' \subsetneq S$ of the number of points contained in S'.

This GAP-MAX approach easily solves the exact case, but it does not readily extend to the approximate case. If we count points near to the subspace, rather than in it, then (infinitely) many subspaces will have high score, which violates the assumptions needed for GAP-MAX to work. Thus we use a completely different approach for the approximate case.

We apply the "subsample and aggregate" paradigm of [NRS07]. That is, we split the dataset $X_1, \cdots, X_n$ into $n/O(k)$ sub-datasets each of size $O(k)$. We use each sub-dataset to compute an approximation to the subspace by doing a (non-private) PCA on the sub-dataset. Let Π be the projection matrix onto the correct subspace and $\Pi_1, \cdots, \Pi_{n/O(k)}$ the projection matrices onto the approximations derived from the sub-datasets. With high probability $\|\Pi_j - \Pi\|$ is small for most j. (Exactly how small depends on the eigengap.) Now we must privately aggregate the projection matrices $\Pi_1, \cdots, \Pi_{n/O(k)}$ into a single projection matrix.

Rather than directly trying to aggregate the projection matrices, we pick a set of reference points, project them onto the subspaces, and then aggregate the projected points. We draw $p_1, \cdots, p_{O(k)}$ independently from a standard spherical Gaussian. Then $\|\Pi_j p_i - \Pi p_i\| \leq \|\Pi_j - \Pi\| \cdot O(\sqrt{k})$ is also small for all i and most j. We wish to privately approximate Πp_i and to do this we have $n/O(k)$ points $\Pi_j p_i$ most of which are close to Πp_i. This is now a location or mean estimation problem, which we can solve privately. Thus we obtain points $\hat{p}_i$ such that $\|\hat{p}_i - \Pi p_i\|$ is small for all i. From a PCA of these points we can obtain a projection $\hat{\Pi}$ with $\|\hat{\Pi} - \Pi\|$ being small, as required.

Finally, we discuss how to privately obtain $(\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_{O(k)})$ from $(\Pi_1 p_1, \cdots, \Pi_1 p_{O(k)}), \cdots,$ $(\Pi_{n/O(k)} p_1, \cdots, \Pi_{n/O(k)} p_{O(k)})$. It is better here to treat $(\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_{O(k)})$ as a single vector in $\mathbb{R}^{O(kd)}$, rather than as $O(k)$ vectors in $\mathbb{R}^d$. We split $\mathbb{R}^{O(kd)}$ into cells and then run a differentially private histogram algorithm. If we construct the cells carefully, for most j we have that $(\Pi_j p_1, \cdots, \Pi_j p_{O(k)})$ is in the same histogram cell as the desired point $(\Pi p_1, \cdots, \Pi p_{O(k)})$. The histogram algorithm will thus identify this cell, and we take an arbitrary point from this cell as our estimate $(\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_{O(k)})$. The differentially private histogram algorithm is run over exponentially many cells, which is possible under (ε, δ)-differential privacy if $n/O(k) \geq O(\log(1/\delta)/\varepsilon)$. (Note that under concentrated differential privacy the histogram algorithm's sample complexity n would need to depend on the number of cells and, hence, the ambient dimension d.)

The main technical ingredients in the analysis of our algorithm for the approximate case are matrix perturbation and concentration analysis and the location estimation procedure using differentially private histograms. Our matrix perturbation analysis uses a variant of the Davis-Kahan theorem to show that if the empirical covariance matrix is close to the true covariance matrix, then the subspaces corresponding to the top k eigenvalues of each are also close; this is applied to both the subsamples and the projection of the reference points. The matrix concentration results that we use show that the empirical covariance matrices in all the subsamples are close to the true covariance matrix. This is the only place where the multivariate Gaussian assumption arises. Any distribution that concentrates well will work.

4 Exact case

Here, we discuss the case, where all n points lie *exactly* in a subspace s_* of dimension k of $\mathbb{R}^d$. Our goal is to privately output that subspace. We do it under the assumption that all strict subspaces of s_* contain at most ℓ points. If the points are in general position, then $\ell = k - 1$, as any strictly smaller subspace has dimension $< k$ and cannot contain more points than its dimension. Let $\mathcal{S}_d^k$ be the set of all k-dimensional subspaces of $\mathbb{R}^d$. Let $\mathcal{S}_d$ be the set of all subspaces of $\mathbb{R}^d$. We formally define that problem as follows.

Problem 4.1. Assume (i) all but at most ℓ, input points are in some $s_* \in \mathcal{S}_d^k$, and (ii) every subspace of dimension $< k$ contains at most ℓ points. (If the points are in general position – aside from being contained in s_* – then $\ell = k - 1$.) The goal is to output a representation of s_*.

We call these $\leq \ell$ points that do not lie in s_*, "adversarial points".

We prove Theorem 1.1 by proving the privacy and the accuracy guarantees of Algorithm 1. The algorithm performs a GAP-MAX (cf. Lemma A.16). It assigns a score to all the relevant subspaces, that is, the subspaces spanned by the points of the dataset X. We show that the only subspace that has a high score is the true subspace s_*, and the rest of the subspaces have low scores. Then GAP-MAX outputs the true subspace successfully because of the gap between the scores of the best subspace and the second to the best one. For GAP-MAX to work all the time, we define a default option in the output space that has a high score, which we call NULL. Thus, the output space is now $\mathcal{Y} = \mathcal{S}_d \cup \{\mathsf{NULL}\}$. Also, for GAP-MAX to run in finite time, we filter $\mathcal{S}_d$ to select finite number of subspaces that have at least 0 scores on the basis of X. Note that this is a preprocessing step, and does not violate privacy as, we will show, all other subspaces already have 0 probability of getting output. We define the score function $u : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{N}$ as follows.

$$u(x, s) := \begin{cases} |x \cap s| - \sup\{|x \cap t| : t \in \mathcal{S}_d, t \subsetneq s\} & \text{if } s \in \mathcal{S}_d \\ \ell + \frac{4\log(1/\delta)}{\varepsilon} + 1 & \text{if } s = \mathsf{NULL} \end{cases}$$

Note that this score function can be computed in finite time because for any m points and $i > 0$, if the points are contained in an i-dimensional subspace, then the subspace that contains all m points must lie within the set of subspaces spanned by $\binom{m}{i+1}$ subsets of points.

We split the proof of Theorem 1.1 into sections for privacy (Lemma 4.2) and accuracy (Lemma 4.4).

4.1 Privacy

Lemma 4.2. *Algorithm 1 is (ε, δ)-differentially private.*

The proof of privacy closely follows the privacy analysis of GAP-MAX by [BDRS18]. The only novelty is that Algorithm 1 may output NULL in the case that the input is malformed (i.e., doesn't satisfy the assumptions of Problem 4.1).

The key is that the score $u(X, s)$ is low sensitivity. Thus $\max\{0, u(X, s) - u(X, s_2) - 1\}$ also has low sensitivity. What we gain from subtracting the second-largest score and taking this maximum is that these values are also sparse – only one ($s = s_1$) is nonzero. This means we can add noise to all the values without paying for composition. We prove the privacy guarantees in Section B.

4.2 Accuracy

We start by showing that the true subspace s_* has a high score, while the rest of the subspaces have low scores.

```
Algorithm 1: DP Exact Subspace Estimator $\mathrm{DPESE}_{\varepsilon,\delta,k,\ell}(X)$

Input: Samples $X \in \mathbb{R}^{d\times n}$. Parameters $\varepsilon, \delta, k, \ell > 0$.
Output: $\hat{s} \in \mathcal{S}_d^k$.
Set $\mathcal{Y} \leftarrow \{\mathsf{NULL}\}$ and sample noise $\xi(\mathsf{NULL})$ from $\mathrm{TLap}(2, \varepsilon, \delta)$.
Set score $u(X, \mathsf{NULL}) = \ell + \frac{4\log(1/\delta)}{\varepsilon} + 1$.
// Identify candidate outputs.
For each subset $S$ of $X$ of size $k$
    Let $s$ be the subspace spanned by $S$.
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{s\}$.
    Sample noise $\xi(s)$ from $\mathrm{TLap}(2, \varepsilon, \delta)$.
    Set score $u(X, s) = |x \cap s| - \sup\{|x \cap t| : t \in \mathcal{S}_d, t \subsetneq s\}$.
// Apply GAP-MAX.
Let $s_1 = \arg\max_{s\in\mathcal{Y}} u(X, s)$ be the candidate with the largest score.
Let $s_2 = \arg\max_{s\in\mathcal{Y}\setminus\{s_1\}} u(X, s)$ be the candidate with the second-largest score.
Let $\hat{s} = \arg\max_{s\in\mathcal{Y}} \max\{0, u(X, s) - u(X, s_2) - 1\} + \xi(s)$.
// Truncated Laplace noise $\xi \sim \mathrm{TLap}(2, \varepsilon, \delta)$; see Lemma A.14
Return $\hat{s}$.
```

Lemma 4.3. *Under the assumptions of Problem 4.1,* $u(x, s_*) \geq n - 2\ell$ *and* $u(x, s') \leq 2\ell$ *for* $s' \neq s_*$.

Proof. We have $u(x, s_*) = |x \cap s_*| - |x \cap s'|$ for some $s' \in \mathcal{S}_d$ with $s' \subsetneq s_*$. The dimension of s' is at most $k - 1$ and, by the assumption (ii), $|x \cap s'| \leq \ell$.

Let $s' \in \mathcal{S}_d \setminus \{s_*\}$. There are three cases to analyse:

1. Let $s' \supsetneq s_*$. Then $u(x, s') \leq |x \cap s'| - |x \cap s_*| \leq \ell$ because the $\leq \ell$ adverserial points and the $\geq n - \ell$ non-adversarial points may not together lie in a subspace of dimension k.
2. Let $s' \subsetneq s_*$. Let k' be the dimension of s'. Clearly $k' < k$. By our assumption (ii), $|s' \cap x| \leq \ell$. Then $u(x, s') = |x \cap s'| - |x \cap t| \leq \ell$ for some t because the $\leq \ell$ adversarial points already don't lie in s_*, so they will not lie in any subspace of s_*.
3. Let s' be incomparable to s_*. Let $s'' = s' \cap s_*$. Then $u(x, s') \leq |x \cap s'| - |x \cap s''| \leq \ell$ because the adversarial points may not lie in s_*, but could be in $s' \setminus s''$.

This completes the proof. □

Now, we show that the algorithm is accurate.

Lemma 4.4. *If* $n \geq 3\ell + \frac{8\log(1/\delta)}{\varepsilon} + 2$, *then Algorithm 1 outputs* s_* *for Problem 4.1.*

Proof. From Lemma 4.3, we know that s_* has a score of at least $n - 2\ell$, and the next best subspace can have a score of at most ℓ. Also, the score of NULL is defined to be $\ell + \frac{4\log(1/\delta)}{\varepsilon} + 1$. This means that the gap satisfies $\max\{0, u(X, s_*) - u(X, s_2) - 1\} \geq n - 3\ell - \frac{4\log(1/\delta)}{\varepsilon} - 1$. Since the noise is bounded by $\frac{2\log(1/\delta)}{\varepsilon}$, our bound on n implies that $\hat{s} = s_*$ □

5 Approximate Case

In this section, we discuss the case, where the data "approximately" lies in a k-dimensional subspace of $\mathbb{R}^d$. We make a Gaussian distributional assumption, where the covariance is approximately k-dimensional, though the results could be extended to distributions with heavier tails using the right inequalities. We formally define the problem:

Problem 5.1. Let $\Sigma \in \mathbb{R}^{d \times d}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$. Fix $k \in [d]$ and let $0 < \gamma \ll 1$, be such that $\frac{\lambda_{k+1}}{\lambda_k} \leq \gamma^2$. Suppose Π is the projection matrix onto the subspace spanned by the eigenvectors of Σ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_k$. Given sample access to $\mathcal{N}(\vec{0}, \Sigma)$, and $0 < \alpha < 1$, output a projection matrix $\widehat{\Pi}$, such that $\|\Pi - \widehat{\Pi}\| \leq \alpha$.

We solve Problem 5.1 under the constraint of (ε, δ)-differential privacy. Throughout this section, we would refer to the subspace spanned by the top k eigenvectors of Σ as the "true" or "actual" subspace.

Algorithm 2 solves Problem 5.1 and proves Theorem 1.2. Here $\| \cdot \|$ is the operator norm.

Remark 5.2. We scale the eigenvalues of Σ so that $\lambda_k = 1$ and $\lambda_{k+1} \leq \gamma^2$. Also, for the purpose of the analysis, we will be splitting $\Sigma = \Sigma_k + \Sigma_{d-k}$, where Σ_k is the covariance matrix formed by the top k eigenvalues and the corresponding eigenvectors of Σ and Σ_{d-k} is remainder. We assume knowledge of k and (an upper bound on) γ.

Algorithm 2 is a type of "Subsample-and-Aggregate" algorithm [NRS07]. We consider multiple subspaces, each given by a disjoint subset of the input points which all come from the same multivariate Gaussian. Our algorithm privately finds a subspace that is close to most of those subspaces. By concentration, most of these subspaces will be close to the true subspace, and thus the privately-found subspace will also be close to the true subspace.

A little more formally, we first sample q public data points (called "reference points") from $\mathcal{N}(\vec{0}, \mathbb{I})$. Next, we divide the original dataset X into disjoint datasets of m samples each, and perform PCA on each subset to identify the rank-k subspace that best captures those samples. Then we project each of the reference points onto each of the subspaces. Now we have $t = \frac{n}{m}$ projections of each reference point, which we will privately aggregate into a single point. Finally, the aggregated points can be used to recover an approximation to the true subspace. To perform the aggregation, we use a DP histogram over a partition of $\mathbb{R}^d$. Specifically, we randomly partition $\mathbb{R}^d$ into cells such that, with high probability, most the projections will lie within one histogram cell. Thus we can privately identify that cell and output a random point from that histogram cell as the aggregated point.

5.1 Privacy

The privacy analysis of our method follows the template of the subsample-and-aggregate framework [NRS07] and our privacy guarantee directly follows from that of the DP histogram subroutine.

Lemma 5.3. *Algorithm 2 is (ε, δ)-differentially private.*

Proof. Changing one point in X can change only one of the X^j's. This can only change one point in Q, which in turn can only change the counts in two histogram cells by 1. Therefore, the sensitivity is 2. Because the sensitivity of the histogram step is bounded by 2 (Lemma 5.3), an application of DP-histogram, by Lemma A.15, is (ε, δ)-DP. Outputting a random point in the privately found histogram cell preserves privacy by post-processing (Lemma A.12). Hence, the claim. □

5.2 Accuracy

The accuracy analysis of Algorithm 2 is relatively complex and is deferred to the full version. The key ingredients come from the literature on matrix concentration bounds and matrix perturbation inequalities. We briefly outline the key steps: First, we apply matrix concentration to show that the empirical covariance matrix $X^j(X^j)^T$ of each subsample is, after rescaling, close to the true covariance matrix Σ with high probability. Second, we apply matrix perturbation inequalities to show that the top-k subspace Π_j corresponding to the empirical covariance matrix $X^j(X^j)^T$ is close to the true top-k subspace Π. It follows that most of the the projected reference points p_i^j are close to the desired value Πp_i. Third, we show that the aggregated projections $\hat{p}_i$ are also close to the true projections Π_i. Finally, we apply matrix perturbation inequalities again to show that the subspace derived from the aggregated projections $\hat{\Pi}$ is close to the true subspace Π.

6 Conclusion, Discussion, & Limitations of Our Work

We provide algorithms for the problem of privately learning subspaces where the sample complexity does not depend on the ambient dimension. This is the first time such results have been given and,

```
Algorithm 2: DP Approximate Subspace Estimator $\mathrm{DPASE}_{\varepsilon,\delta,\alpha,\gamma,k}(X)$

Input: Samples $X_1, \dots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \alpha, \gamma, k > 0$.
Output: Projection matrix $\widehat{\Pi} \in \mathbb{R}^{d \times d}$ of rank $k$.

Set parameters: $t \leftarrow \frac{C_0 \ln(1/\delta)}{\varepsilon}$    $m \leftarrow \lfloor n/t \rfloor$    $q \leftarrow C_1 k$    $\ell \leftarrow \frac{C_2 \gamma \sqrt{d} k (\sqrt{k} + \sqrt{\ln(kt)})}{\sqrt{m}}$

Sample reference points $p_1, \dots, p_q$ from $\mathcal{N}(\vec{0}, \mathbb{I})$ independently.

// Subsample from $X$, and form projection matrices.
For $j \in 1, \dots, t$
    Let $X^j = (X_{(j-1)m+1}, \dots, X_{jm}) \in \mathbb{R}^{d \times m}$.
    Let $\Pi_j \in \mathbb{R}^{d \times d}$ be the projection matrix onto the subspace spanned by the eigenvectors of
      $X^j (X^j)^T \in \mathbb{R}^{d \times d}$ corresponding to the largest $k$ eigenvalues.
    For $i \in 1, \dots, q$
        $p_i^j \leftarrow \Pi_j p_i$

// Create histogram cells with random offset.
Let $\lambda$ be a random number in $[0, 1)$.
Divide $\mathbb{R}^{qd}$ into $\Omega = \{\dots, [\lambda\ell + i\ell, \lambda\ell + (i+1)\ell), \dots\}^{qd}$, for all $i \in \mathbb{Z}$.
Let each disjoint cell of length $\ell$ be a histogram bucket.

// Perform private aggregation of subspaces.
For each $i \in [q]$, let $Q_i \in \mathbb{R}^{d \times t}$ be the dataset, where column $j$ is $p_i^j$.
Let $Q \in \mathbb{R}^{qd \times t}$ be the vertical concatenation of all $Q_i$'s in order.
Run $(\varepsilon, \delta)$-DP histogram over $\Omega$ using $Q$ to get $\omega \in \Omega$ that contains at least $\frac{t}{2}$ points.
If no such $\omega$ exists
    Return $\perp$

// Return the subspace.
Let $\widehat{p} = (\widehat{p}_1, \dots, \widehat{p}_d, \dots, \widehat{p}_{(q-1)d+1}, \dots, \widehat{p}_{qd})$ be a random point in $\omega$.
For each $i \in [q]$
    Let $\widehat{p}_i = (\widehat{p}_{(i-1)d+1}, \dots, \widehat{p}_{id}) \in \mathbb{R}^d$.
Let $\widehat{\Pi}$ be the projection matrix onto the subspace spanned by the eigenvectors corresponding to
  the $k$ largest eigenvalues of $\sum_{i=1}^{q} \widehat{p}_i \widehat{p}_i^T$.
Return $\widehat{\Pi}$.
```

as discussed in §2.1, prior work in the general area of private matrix analysis uses techniques that fundamentally cannot achieve sample complexity that is independent of the ambient dimension.

To achieve dimension-independent sample complexity, we must make strong assumptions about the data. Specifically, we must assume that the data points lie in or very near to a low-dimensional subspace. This is a limitation of our work. However, we emphasize that such assumptions are necessary to obtain dimension-independent sample complexity *even in the non-private setting* [CZ16].

We believe that the specific parameters in our results can be improved. We conjecture that the γ^2 parameter in Theorem 1.2 (which controls the eigenvalue gap) can be improved. Specifically, the exponent on the ambient dimension d seems like it could be improved. (Although we know that it cannot be eliminated entirely.)

Our eigenvalue gap assumption could also be relaxed – rather than requiring a gap between λ_k and λ_{k+1}, we could require a gap between λ_k and $\lambda_{k+\ell}$. However, this would require changing other aspects of the problem formulation.

We hope that our work inspires further work. Generally, we believe that exploiting structure in the data to avoid privacy's curse of dimensionality is a fruitful and valuable research direction.

References

[ABU18] R. Arora, V. Braverman, and J. Upadhyay. “Differentially private robust low-rank approximation”. In: *Advances in neural information processing systems* (2018) (cit. on p. 4).

[ACGMMTZ16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318 (cit. on p. 1).

[ADKMV18] K. Amin, T. Dick, A. Kulesza, A. M. Medina, and S. Vassilvitskii. “Private covariance estimation via iterative eigenvector sampling”. In: *2018 NIPS workshop in Privacy-Preserving Machine Learning*. Vol. 250. 2018 (cit. on p. 4).

[BBDS12] J. Blocki, A. Blum, A. Datta, and O. Sheffet. “The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy”. In: *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*. FOCS ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 410–419 (cit. on p. 4).

[BBNS19] J. Błasiok, M. Bun, A. Nikolov, and T. Steinke. “Towards instance-optimal private query release”. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2480–2497 (cit. on p. 1).

[BCMNUW20] R. Bassily, A. Cheu, S. Moran, A. Nikolov, J. Ullman, and S. Wu. “Private query release assisted by public data”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 695–703 (cit. on p. 1).

[BDMN05] A. Blum, C. Dwork, F. McSherry, and K. Nissim. “Practical Privacy: The SuLQ Framework”. In: *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS ’05. New York, NY, USA: ACM, 2005, pp. 128–138 (cit. on p. 4).

[BDRS18] M. Bun, C. Dwork, G. N. Rothblum, and T. Steinke. “Composable and Versatile Privacy via Truncated CDP”. In: *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*. STOC ’18. New York, NY, USA: ACM, 2018, pp. 74–86 (cit. on pp. 5, 6, 17).

[BLR08] A. Blum, K. Ligett, and A. Roth. “A Learning Theory Approach to Non-Interactive Database Privacy”. In: *STOC*. 2008 (cit. on p. 1).

[BNS16] M. Bun, K. Nissim, and U. Stemmer. “Simultaneous Private Learning of Multiple Concepts”. In: *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science*. ITCS ’16. New York, NY, USA: ACM, 2016, pp. 369–380 (cit. on p. 17).

[BS16] M. Bun and T. Steinke. “Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds”. In: *Proceedings of the 14th Conference on Theory of Cryptography*. TCC ’16-B. Berlin, Heidelberg: Springer, 2016, pp. 635–658 (cit. on p. 4).

[BUV14] M. Bun, J. Ullman, and S. Vadhan. “Fingerprinting Codes and the Price of Approximate Differential Privacy”. In: *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*. STOC ’14. New York, NY, USA: ACM, 2014, pp. 1–10 (cit. on p. 1).

[CSS12] K. Chaudhuri, A. Sarwate, and K. Sinha. “Near-optimal differentially private principal components”. In: *Advances in Neural Information Processing Systems* 25 (2012), pp. 989–997 (cit. on p. 4).

[CZ16] T. Cai and A. Zhang. “Rate-Optimal Perturbation Bounds for Singular Subspaces with Applications to High-Dimensional Statistics”. In: *The Annals of Statistics* 46 (May 2016) (cit. on pp. 3, 9, 14).

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Proceedings of the 3rd Conference on Theory of Cryptography*. TCC ’06. Berlin, Heidelberg: Springer, 2006, pp. 265–284 (cit. on p. 16).

[DSSUV15] C. Dwork, A. Smith, T. Steinke, J. Ullman, and S. Vadhan. “Robust Traceability from Trace Amounts”. In: *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*. FOCS ’15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 650–669 (cit. on p. 1).

[DTTZ14] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. “Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis”. In: *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*. STOC ’14. New York, NY, USA: ACM, 2014, pp. 11–20 (cit. on p. 4).

[FT20] Y. Feng and Y. Tu. “How neural networks find generalizable solutions: Self-tuned annealing in deep learning”. In: *arXiv preprint arXiv:2001.01678* (2020) (cit. on p. 1).

[GARD18] G. Gur-Ari, D. A. Roberts, and E. Dyer. “Gradient descent happens in a tiny subspace”. In: *arXiv preprint arXiv:1812.04754* (2018) (cit. on p. 1).

[GDGK20] Q. Geng, W. Ding, R. Guo, and S. Kumar. “Tight Analysis of Privacy and Utility Tradeoff in Approximate Differential Privacy”. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by S. Chiappa and R. Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 89–99 (cit. on p. 17).

[GGB18] A. Gonem and R. Gilad-Bachrach. “Smooth Sensitivity Based Approach for Differentially Private PCA”. In: *Algorithmic Learning Theory*. ALT ’18. JMLR, Inc., 2018, pp. 438–450 (cit. on p. 4).

[HP13] M. Hardt and E. Price. “The noisy power method: A meta algorithm with applications”. In: *arXiv preprint arXiv:1311.2495* (2013) (cit. on p. 4).

[HR10] M. Hardt and G. N. Rothblum. “A multiplicative weights mechanism for privacy-preserving data analysis”. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 61–70 (cit. on p. 1).

[HR12] M. Hardt and A. Roth. “Beating randomized response on incoherent matrices”. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 2012, pp. 1255–1268 (cit. on p. 4).

[HR13] M. Hardt and A. Roth. “Beyond worst-case analysis in private singular vector computation”. In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 2013, pp. 331–340 (cit. on p. 4).

[HT10] M. Hardt and K. Talwar. “On the Geometry of Differential Privacy”. In: *Proceedings of the 42nd Annual ACM Symposium on the Theory of Computing*. STOC ’10. New York, NY, USA: ACM, 2010, pp. 705–714 (cit. on pp. 1, 4).

[KRRT20] P. Kairouz, M. Ribero, K. Rush, and A. Thakurta. *Fast Dimension Independent Private AdaGrad on Publicly Estimated Subspaces*. 2020 (cit. on p. 1).

[KT13] M. Kapralov and K. Talwar. “On Differentially Private Low Rank Approximation”. In: *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’13. Philadelphia, PA, USA: SIAM, 2013, pp. 1395–1414 (cit. on p. 4).

[LFLY18] C. Li, H. Farkhoor, R. Liu, and J. Yosinski. “Measuring the intrinsic dimension of objective landscapes”. In: *arXiv preprint arXiv:1804.08838* (2018) (cit. on p. 1).

[LGZCB20] X. Li, Q. Gu, Y. Zhou, T. Chen, and A. Banerjee. “Hessian based analysis of sgd for deep nets: Dynamics and generalization”. In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM. 2020, pp. 190–198 (cit. on p. 1).

[LXTSG17] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. “Visualizing the loss landscape of neural nets”. In: *arXiv preprint arXiv:1712.09913* (2017) (cit. on p. 1).

[MM09] F. McSherry and I. Mironov. “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*. 2009, pp. 627–636 (cit. on p. 4).

[MT07] F. McSherry and K. Talwar. “Mechanism Design via Differential Privacy”. In: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. FOCS ’07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 94–103 (cit. on p. 4).

[NRS07] K. Nissim, S. Raskhodnikova, and A. Smith. “Smooth Sensitivity and Sampling in Private Data Analysis”. In: *Proceedings of the 39th Annual ACM Symposium on the Theory of Computing*. STOC ’07. New York, NY, USA: ACM, 2007, pp. 75–84 (cit. on pp. 3–5, 8).

[NS06] A. Narayanan and V. Shmatikov. “How to break anonymity of the netflix prize dataset”. In: *arXiv preprint cs/0610105* (2006) (cit. on p. 3).

[She19] O. Sheffet. “Old techniques in differentially private linear regression”. In: *Algorithmic Learning Theory*. PMLR. 2019, pp. 789–827 (cit. on p. 4).

[SU17] T. Steinke and J. Ullman. “Between Pure and Approximate Differential Privacy”. In: *The Journal of Privacy and Confidentiality* 7.2 (2017), pp. 3–22 (cit. on p. 1).

[Ull15] J. Ullman. “Private multiplicative weights beyond linear queries”. In: *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2015, pp. 303–312 (cit. on p. 1).

[Vad17] S. Vadhan. “The Complexity of Differential Privacy”. In: *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*. Ed. by Y. Lindell. Cham, Switzerland: Springer International Publishing AG, 2017. Chap. 7, pp. 347–450 (cit. on p. 17).

[Ver18] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018 (cit. on p. 15).

[WSCHT16] L. Wei, A. D. Sarwate, J. Corander, A. Hero, and V. Tarokh. “Analysis of a privacy-preserving PCA algorithm using random matrix theory”. In: *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2016, pp. 1335–1339 (cit. on p. 4).

[ZWB20] Y. Zhou, Z. S. Wu, and A. Banerjee. “Bypassing the ambient dimension: Private sgd with gradient subspace identification”. In: *arXiv preprint arXiv:2007.03813* (2020) (cit. on p. 1).

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Notations, Definitions, and Background Results

A.1 Linear Algebra and Probability Preliminaries

Here, we mention a few key technical results that we will be using to prove the main theorem for the approximate case. Throughout this document, we assume that the dimension d is larger than some absolute constant, and adopt the following notation: for a matrix A of rank r, we use $s_1(A) \geq \cdots \geq s_r(A)$ to denote the singular values of A in decreasing order, and use $\lambda_1(A) \geq \cdots \geq \lambda_r(A)$ to denote the eigenvalues of A in decreasing order; let $s_{\min}(A)$ denote the least, non-zero singular value of A. We omit the parentheses when the context is clear. We begin by stating two results about matrix perturbation theory. The first result says that if two matrices are close to one another in operator norm, then their corresponding singular values are also close to one another.

Define

$$\|M\| := \sup\{\|Mx\|_2 : x \in \mathbb{R}^d,\ \|x\|_2 \leq 1\}$$

to be the operator norm with respect to the Euclidean vector norm.

Lemma A.1 (Singular Value Inequality)**.** *Let $A, B \in \mathbb{R}^{d\times n}$ and let $r = \min\{d, n\}$. Then for $1 \leq i, j \leq r$,*

$$s_{i+j-1}(A+B) \leq s_i(A) + s_j(B).$$

The following result gives a lower bound on the least singular value of sum of two matrices.

Lemma A.2 (Least Singular Value of Matrix Sum)**.** *Let $A, B \in \mathbb{R}^{d\times n}$. Then*

$$s_{\min}(A+B) \geq s_{\min}(A) - \|B\|.$$

The next result bounds the angle between the subspaces spanned by two matrices that are close to one another. Let $X \in \mathbb{R}^{d\times n}$ have the following SVD.

$$X = [U \quad U_\perp] \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix}$$

In the above, $U, U_\perp$ are orthonormal matrices such that $U \in \mathbb{R}^{d\times r}$ and $U_\perp \in \mathbb{R}^{d\times(d-r)}$, Σ_1, Σ_2 are diagonal matrices, such that $\Sigma_1 \in \mathbb{R}^{r\times r}$ and $\Sigma_2 \in \mathbb{R}^{(d-r)\times(n-r)}$, and $V, V_\perp$ are orthonormal matrices, such that $V \in \mathbb{R}^{n\times r}$ and $V_\perp \in \mathbb{R}^{n\times(n-r)}$. Let $Z \in \mathbb{R}^{d\times n}$ be a perturbation matrix, and $\hat{X} = X + Z$, such that $\hat{X}$ has the following SVD.

$$\hat{X} = \begin{bmatrix} \hat{U} & \hat{U}_\perp \end{bmatrix} \cdot \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \cdot \begin{bmatrix} \hat{V}^T \\ \hat{V}_\perp^T \end{bmatrix}$$

In the above, $\hat{U}, \hat{U}_\perp, \hat{\Sigma}_1, \hat{\Sigma}_2, \hat{V}, \hat{V}_\perp$ have the same structures as $U, U_\perp, \Sigma_1, \Sigma_2, V, V_\perp$ respectively. Let $Z_{21} = U_\perp U_\perp^T Z V V^T$ and $Z_{12} = U U^T Z V_\perp V_\perp^T$. Suppose $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$ are the singular values of $U^T \hat{U}$. Let $\Theta(U, \hat{U}) \in \mathbb{R}^{r\times r}$ be a diagonal matrix, such that $\Theta_{ii}(U, \hat{U}) = \cos^{-1}(\sigma_i)$.

Lemma A.3 (Sin(Θ) Theorem [CZ16])**.** *Let $X, \hat{X}, Z, Z_{12}, Z_{21}$ be defined as above. Denote $\alpha = s_{\min}(U^T \hat{X} V)$ and $\beta = \|U_\perp^T \hat{X} V_\perp\|$. If $\alpha^2 > \beta^2 + \min\{\|Z_{12}\|^2, \|Z_{21}\|^2\}$, then we have the following.*

$$\|\mathit{Sin}(\Theta)(U, \hat{U})\| \leq \frac{\alpha\|Z_{21}\| + \beta\|Z_{12}\|}{\alpha^2 - \beta^2 - \min\{\|Z_{12}\|^2, \|Z_{21}\|^2\}}$$

The next result bounds $\|\mathrm{Sin}(\Theta)(U, \hat{U})\|$ in terms of the distance between UU^T and $\hat{U}\hat{U}^T$.

Lemma A.4 (Property of $\|\mathrm{Sin}(\Theta)\|$ [CZ16])**.** *Let $U, \hat{U} \in \mathbb{R}^{d\times r}$ be orthonormal matrices, and let $\Theta(U, \hat{U})$ be defined as above in terms of $\hat{U}, U$. Then we have the following.*

$$\|\mathit{Sin}(\Theta)(U, \hat{U})\| \leq \|\hat{U}\hat{U}^T - UU^T\| \leq 2\|\mathit{Sin}(\Theta)(U, \hat{U})\|$$

The next result bounds the singular values of a matrix, whose columns are independent vectors from a mean zero, isotropic distribution in $\mathbb{R}^d$. We first define the sub-Gaussian norm of a random variable.

Definition A.5. Let X be a sub-Gaussian random variable. The sub-Gaussian norm of X, denoted by $\|X\|_{\psi^2}$, is defined as,

$$\|X\|_{\psi^2} = \inf\{t > 0 : \mathbb{E}\left[\exp(X^2/t^2)\right] \leq 2\}.$$

Lemma A.6 (Theorem 4.6.1 [Ver18]). *Let A be an $n \times m$ matrix, whose columns A_i are independent, mean zero, sub-Gaussian isotropic random vectors in $\mathbb{R}^n$. Then for any $t \geq 0$, we have*

$$\sqrt{m} - CK^2(\sqrt{n} + t) \leq s_n(A) \leq s_1(A) \leq \sqrt{m} + CK^2(\sqrt{n} + t)$$

with probability at least $1 - 2\text{exp}(-t^2)$. Here, $K = \max_i \|A\|_{\psi^2}$ (sub-Gaussian norm of A).

In the above, $\|A\|_{\psi^2} \in O(1)$ if the distribution in question is $\mathcal{N}(\vec{0}, \mathbb{I})$. The following corollary generalises the above result for arbitrary Gaussians.

Corollary A.7. *Let A be an $n \times m$ matrix, whose columns A_i are independent, random vectors in $\mathbb{R}^n$ from $\mathcal{N}(\vec{0}, \Sigma)$. Then for any $t \geq 0$, we have*

$$(\sqrt{m} - CK^2(\sqrt{n} + t))\sqrt{s_n(\Sigma)} \leq s_n(A) \leq (\sqrt{m} + CK^2(\sqrt{n} + t))\sqrt{s_n(\Sigma)}$$

and

$$s_1(A) \leq (\sqrt{m} + CK^2(\sqrt{n} + t))\sqrt{s_1(\Sigma)}$$

with probability at least $1 - 2\text{exp}(-t^2)$. Here, $K = \max_i \|A\|_{\psi^2}$ (sub-Gaussian norm of A).

Proof. First, we prove the lower bound on $s_n(A)$. Note that $s_n(A) = \min\limits_{\|x\|>0} \frac{\|Ax\|}{\|x\|}$, and that the columns of $\Sigma^{-\frac{1}{2}}A$ are distributed as $\mathcal{N}(\vec{0}, \mathbb{I})$. Therefore, we have the following.

$$\begin{aligned}
\min_{\|x\|>0} \frac{\|Ax\|}{\|x\|} &= \min_{\|x\|>0} \frac{\|\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}Ax\|}{\|x\|} \\
&= \min_{\|x\|>0} \frac{\|\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}Ax\|}{\|\Sigma^{-\frac{1}{2}}Ax\|} \frac{\|\Sigma^{-\frac{1}{2}}Ax\|}{\|x\|} \\
&\geq \min_{\|x\|>0} \frac{\|\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}Ax\|}{\|\Sigma^{-\frac{1}{2}}Ax\|} \min_{\|x\|>0} \frac{\|\Sigma^{-\frac{1}{2}}Ax\|}{\|x\|} \\
&\geq \min_{\|y\|>0} \frac{\|\Sigma^{\frac{1}{2}}y\|}{\|y\|} \min_{\|x\|>0} \frac{\|\Sigma^{-\frac{1}{2}}Ax\|}{\|x\|} \\
&\geq (\sqrt{m} - CK^2(\sqrt{n} + t))\sqrt{s_n(\Sigma)} && \text{(Lemma A.6)}
\end{aligned}$$

Next, we prove the upper bound on $s_n(A)$. For this, we first show that for $X \in \mathbb{R}^{m \times d}$ and $Y \in \mathbb{R}^{d \times n}$, $s_{\min}(XY) \leq s_{\min}(X) \cdot \|Y\|$.

$$\begin{aligned}
s_{\min}(XY) &= \min_{\|z\|=1} \|XYz\| \\
&\leq \min_{\|z\|=1} \|X\|\|Yz\| \\
&= \|X\| \cdot \min_{\|z\|=1} \|Yz\| \\
&= \|X\| \cdot s_{\min}(Y)
\end{aligned}$$

Now, $s_{\min}(XY) = s_{\min}(Y^TX^T) \leq \|Y\| \cdot s_{\min}(X)$ by the above reasoning. Using this results, we have the following.

$$\begin{aligned}
s_n(A) &= s_n(\Sigma^{1/2} \cdot \Sigma^{-1/2}A) \\
&\leq s_n(\Sigma^{1/2})\|\Sigma^{-1/2}A\| \\
&\leq (\sqrt{m} + CK^2(\sqrt{n} + t))\sqrt{s_n(\Sigma)} && \text{(Lemma A.6)}
\end{aligned}$$

Now, we show the upper bound on $s_1(A)$. Note that $s_1(A) = \|A\|$.

$$\begin{aligned}
\|A\| &= \|\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}A\| \\
&\leq \|\Sigma^{\frac{1}{2}}\| \cdot \|\Sigma^{-\frac{1}{2}}A\| \\
&\leq (\sqrt{m} + CK^2(\sqrt{n} + t))\sqrt{s_1(\Sigma)} && \text{(Lemma A.6)}
\end{aligned}$$

This completes the proof. □

Now, we state a concentration inequality for χ^2 random variables.

Lemma A.8. *Let X be a χ^2 random variable with k degrees of freedom. Then,*

$$\mathbb{P}\left[X > k + 2\sqrt{kt} + 2t\right] \leq e^{-t}.$$

Next, we state the well-known Bernstein's inequality for sums of independent Bernoulli random variables.

Lemma A.9 (Bernstein's Inequality)**.** *Let $X_1, \dots, X_m$ be independent Bernoulli random variables taking values in $\{0,1\}$. Let $p = \mathbb{E}[X_i]$. Then for $m \geq \frac{5p}{2\varepsilon^2}\ln(2/\beta)$ and $\varepsilon \leq p/4$,*

$$\mathbb{P}\left[\left|\frac{1}{m}\sum X_i - p\right| \geq \varepsilon\right] \leq 2e^{-\varepsilon^2 m/2(p+\varepsilon)} \leq \beta.$$

We finally state a result about the norm of a vector sampled from $\mathcal{N}(\vec{0}, \mathbb{I})$.

Lemma A.10. *Let $X_1, \dots, X_q \sim \mathcal{N}(\vec{0}, \Sigma)$ be vectors in $\mathbb{R}^d$, where Σ is the projection of $\mathbb{I}_{d\times d}$ on to a subspace of $\mathbb{R}^d$ of rank k. Then*

$$\mathbb{P}\left[\forall i, \|X_i\|^2 \leq k + 2\sqrt{kt} + 2t\right] \geq 1 - qe^{-t}.$$

Proof. Since Σ is of rank k, we can directly use Lemma A.8 for a fixed $i \in [q]$, and the union bound over all $i \in [q]$ to get the required result. This is because for any i, $\|X_i\|^2$ is a χ^2 random variable with k degrees of freedom. □

A.2 Privacy Preliminaries

Definition A.11 (Differential Privacy (DP) [DMNS06])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy ((ε, δ)-DP) if for every pair of neighboring datasets $X, X' \in \mathcal{X}^n$ (i.e., datasets that differ in exactly one entry),

$$\forall Y \subseteq \mathcal{Y} \quad \mathbb{P}[M(X) \in Y] \leq e^{\varepsilon} \cdot \mathbb{P}[M(X') \in Y] + \delta.$$

When $\delta = 0$, we say that M satisfies ε-differential privacy or pure differential privacy.

Neighbouring datasets are those that differ by the replacement of one individual's data. In our setting, each individual's data is assumed to correspond to one point in $\mathcal{X} = \mathbb{R}^d$, so neighbouring means one point is changed arbitrarily.

Throughout the document, we will assume that ε is smaller than some absolute constant less than 1 for notational convenience, but note that our results still hold for general ε. Now, this privacy definition is closed under post-processing.

Lemma A.12 (Post Processing [DMNS06])**.** *If $M : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-DP, and $P : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then the algorithm $P \circ M$ is (ε, δ)-DP.*

A.3 Basic Differentially Private Mechanisms.

We first state standard results on achieving privacy via noise addition proportional to sensitivity [DMNS06].

Definition A.13 (Sensitivity)**.** Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function, its ℓ_1-*sensitivity* and ℓ_2-*sensitivity* are

$$\Delta_{f,1} = \max_{X \sim X' \in \mathcal{X}^n} \|f(X) - f(X')\|_1 \quad \text{and} \quad \Delta_{f,2} = \max_{X \sim X' \in \mathcal{X}^n} \|f(X) - f(X')\|_2,$$

respectively. Here, $X \sim X'$ denotes that X and X' are neighboring datasets (i.e., those that differ in exactly one entry).

One way of introducing (ε, δ)-differential privacy is via adding noise sampled from the truncated Laplace distribution, proportional to the ℓ_1 sensitivity.

Lemma A.14 (Truncated Laplace Mechanism [GDGK20])**.** *Define the probability density function (p) of the truncated Laplace distribution as follows.*

$$p(x) = \begin{cases} Be^{-\frac{|x|}{\lambda}} & \textit{if } x \in [-A, A] \\ 0 & \textit{otherwise} \end{cases}$$

In the above,

$$\lambda = \frac{\Delta}{\varepsilon}, \quad A = \frac{\Delta}{\varepsilon}\log\left(1 + \frac{e^{\varepsilon} - 1}{2\delta}\right), \quad B = \frac{1}{2\lambda(1 - e^{-\frac{A}{\lambda}})}.$$

Let $\mathrm{TLap}(\Delta, \varepsilon, \delta)$ *denote a draw from the above distribution.*

Let $f : \mathcal{X}^n \to \mathbb{R}^d$ *be a function with sensitivity* Δ*. Then the truncated Laplace mechanism*

$$M(X) = f(X) + \mathrm{TLap}(\Delta, \varepsilon, \delta)$$

satisfies (ε, δ)*-DP.*

In the above $A \leq \frac{\Delta_{f,1}}{\varepsilon}\log(1/\delta)$ since ε is smaller than some absolute constant less than 1. Now, we introduce differentially private histograms.

Lemma A.15 (Private Histograms)**.** *Let* $n \in \mathbb{N}$*,* $\varepsilon, \delta, \beta > 0$*, and* $\mathcal{X}$ *a set. There exists* $M : \mathcal{X}^n \to \mathbb{R}^{\mathcal{X}}$ *which is* (ε, δ)*-differentially private and, for all* $x \in \mathcal{X}^n$*, we have*

$$\underset{M}{\mathbb{P}}\left[\sup_{y \in \mathcal{X}} \left| M(x)_y - \frac{1}{n}|\{i \in [n] : x_i = y\}| \right| \leq O\left(\frac{\log(1/\delta\beta)}{\varepsilon n}\right)\right] \geq 1 - \beta.$$

The above holds due to [BNS16; Vad17]. Finally, we introduce the GAP-MAX algorithm from [BDRS18] that outputs the element from the output space that has the highest score function, given that there is a significant gap between the scores of the highest and the second to the highest elements.

Lemma A.16 (GAP-MAX Algorithm [BDRS18])**.** *Let* $\textsc{Score} : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}$ *be a score function with sensitivity* 1 *in its first argument, and let* $\varepsilon, \delta > 0$*. Then there exists a* (ε, δ)*-differentially private algorithm* $M : \mathcal{X}^n \to \mathcal{Y}$ *and* $\alpha = \Theta(\log(1/\delta)/\varepsilon n)$ *with the following property. Fix an input* $X \in \mathcal{X}^n$*. Let*

$$y^* = \arg\max_{y \in \mathcal{Y}}\{\textsc{Score}(X, y)\}.$$

Suppose

$$\forall y \in \mathcal{Y}, y \neq y^* \implies \textsc{Score}(X, y) < \textsc{Score}(X, y^*) - \alpha n.$$

Then M *outputs* y^* *with probability 1.*

B Proof of Privacy of Algorithm 1

Proof of Lemma 4.2. First, we argue that the sensitivity of u is 1. The quantity $|X \cap s|$ has sensitivity 1 and so does $\sup\{|X \cap t| : t \in \mathcal{S}_d, t \subsetneq s\}$. This implies sensitivity 2 by the triangle inequality. However, we see that it is not possible to change one point that simultaneously increases $|X \cap s|$ and decreases $\sup\{|X \cap t| : t \in \mathcal{S}_d, t \subsetneq s\}$ or vice versa. Thus the sensitivity is actually 1.

We also argue that $u(X, s_2)$ has sensitivity 1, where s_2 is the candidate with the second-largest score. Observe that the second-largest score is a monotone function of the collection of all scores – i.e., increasing scores cannot decrease the second-largest score and vice versa. Changing one input point can at most increase all the scores by 1, which would only increase the second-largest score by 1.

This implies that $\max\{0, u(X, s) - u(X, s_2) - 1\}$ has sensitivity 2 by the triangle inequality and the fact that the maximum does not increase the sensitivity.

Now we observe that for any input X there is at most one s such that $\max\{0, u(X, s) - u(X, s_2) - 1\} \neq 0$, namely $s = s_1$. We can say something even stronger: Let X and X' be neighbouring datasets with s_1 and s_2 the largest and second-largest scores on X and s_1' and s_2' the largest and second-largest scores on X'. Then there is at most one s such that $\max\{0, u(X, s) - u(X, s_2) - 1\} \neq 0$ or $\max\{0, u(X', s) - u(X', s_2') - 1\} \neq 0$. In other words, we cannot have both $u(X, s_1) - u(X, s_2) > 1$ and $u(X', s_1') - u(X', s_2') > 1$ unless $s_1 = s_1'$. This holds because $u(X, s) - u(X, s_2)$ has sensitivity 2.

With these observations in hand, we can delve into the privacy analysis. Let X and X' be neighbouring datasets with s_1 and s_2 the largest and second-largest scores on X and s_1' and s_2' the largest and second-largest scores on X'. Let $\mathcal{Y}$ be the set of candidates from X and let $\mathcal{Y}'$ be the set of candidates from X'. Let $\check{\mathcal{Y}} = \mathcal{Y} \cup \mathcal{Y}'$ and $\hat{\mathcal{Y}} = \mathcal{Y} \cap \mathcal{Y}'$.

We note that, for $s \in \check{\mathcal{Y}}$, if $u(X, s) \leq \ell$, then there is no way that $\hat{s} = s$. This is because $|\xi(s)| \leq \frac{2\log(1/\delta)}{\varepsilon}$ for all s and hence, there is no way we could have $\arg\max_{s \in \mathcal{Y}} \max\{0, u(X, s) - u(X, s_2) - 1\} + \xi(s) \geq \arg\max_{s \in \mathcal{Y}} \max\{0, u(X, \mathsf{NULL}) - u(X, s_2) - 1\} + \xi(\mathsf{NULL})$.

If $s \in \check{\mathcal{Y}} \setminus \hat{\mathcal{Y}}$, then $u(X, s) \leq |X \cap s| \leq k + 1 \leq \ell$ and $u(X', s) \leq \ell$. This is because $s \notin \hat{\mathcal{Y}}$ implies $|X \cap s| < k$ or $|X' \cap s| < k$, but $|X \cap s| \leq |X' \cap s| + 1$. Thus, there is no way these points are output and, hence, we can ignore these points in the privacy analysis. (This is the reason for adding the NULL candidate.)

Now we argue that the entire collection of noisy values $\max\{0, u(X, s) - u(X, s_2) - 1\} + \xi(s)$ for $s \in \hat{\mathcal{Y}}$ is differentially private. This is because we are adding noise to a vector where (i) on the neighbouring datasets only 1 coordinate is potentially different and (ii) this coordinate has sensitivity 2. □

C Lower Bound for Exact Case

Here, we show that our upper bound is optimal up to constants for the exact case.

Theorem C.1. *Any (ε, δ)-DP algorithm that takes a dataset of n points satisfying the conditions in Problem 4.1 and outputs s_* with probability > 0.5 requires $n \geq \Omega\left(\ell + \frac{\log(1/\delta)}{\varepsilon}\right)$.*

Proof. First, $n \geq \ell + k$. This is because we need at least k points to span the subspace, and ℓ points could be corrupted. Second, $n \geq \Omega(\log(1/\delta)/\varepsilon)$ by group privacy. Otherwise, the algorithm is $(10, 0.1)$-differentially private with respect to changing the *entire* dataset and it is clearly impossible to output the subspace under this condition. □

D Proof of Accuracy of Algorithm 2

Now we delve into the utility analysis of the algorithm. For $1 \leq j \leq t$, let X^j be the subsets of X as defined in Algorithm 2, and Π_j be the projection matrices of their respective subspaces. We now show that Π_j and the projection matrix of the subspace spanned by Σ_k are close in operator norm.

Lemma D.1. *Let Π be the projection matrix of the subspace spanned by the vectors of Σ_k, and for each $1 \leq j \leq t$, let Π_j be the projection matrix as defined in Algorithm 2. If $m \geq O(k + \ln(qt))$, then*

$$\mathbb{P}\left[\forall j, \|\Pi - \Pi_j\| \leq O\left(\frac{\gamma\sqrt{d}}{\sqrt{m}}\right)\right] \geq 0.95$$

Proof. We show that the subspaces spanned by X^j and the true subspace spanned by Σ are close. Formally, we invoke Lemmata A.3 and A.4. This closeness follows from standard matrix concentration inequalities.

Fix a $j \in [t]$. Note that X^j can be written as $Y^j + H$, where Y^j is the matrix of vectors distributed as $\mathcal{N}(\vec{0}, \Sigma_k)$, and H is a matrix of vectors distributed as $\mathcal{N}(\vec{0}, \Sigma_{d-k})$, where Σ_k and Σ_{d-k} are defined as in Remark 5.2. By Corollary A.7, with probability at least $1 - \frac{0.02}{t}$, $s_k(Y^j) \in \Theta((\sqrt{m} + \sqrt{k})(\sqrt{s_k(\Sigma_k)})) = \Theta(\sqrt{m} + \sqrt{k}) > 0$. Therefore, the subspace spanned by Y^j is the same as the subspace spanned by Σ_k. So, it suffices to look at the subspace spanned by Y^j.

Now, by Corollary A.7, we know that with probability at least $1 - \frac{0.02}{t}$, $\|X^j - Y^j\| = \|H\| \leq O((\sqrt{m} + \sqrt{d})\sqrt{s_1(\Sigma_{d-k})}) \leq O(\gamma(\sqrt{m} + \sqrt{d})\sqrt{s_k(\Sigma_k)}) \leq O(\gamma(\sqrt{m} + \sqrt{d}))$.

We wish to invoke Lemma A.3. Let UDV^T be the SVD of Y^j, and let $\hat{U}\hat{D}\hat{V}^T$ be the SVD of X^j. Now, for a matrix M, let Π_M denote the projection matrix of the subspace spanned by the columns of M. Define quantities a, b, z_{12}, z_{21} as follows.

$$\begin{aligned}
a &= s_{\min}(U^T X^j V) \\
&= s_{\min}(U^T Y^j V + U^T H V) \\
&= s_{\min}(U^T Y^j V) && \text{(Columns of } U \text{ are orthogonal to columns of } H\text{)} \\
&= s_k(Y^j) \\
&\in \Theta(\sqrt{m} + \sqrt{k}) \\
&\in \Theta(\sqrt{m}) \\
b &= \|U_\perp^T X^j V_\perp\| \\
&= \|U_\perp^T Y^j V_\perp + U_\perp^T H V_\perp\| \\
&= \|U_\perp^T H V_\perp\| && \text{(Columns of } U_\perp \text{ are orthogonal to columns of } Y^j\text{)} \\
&\leq \|H\| \\
&\leq O(\gamma(\sqrt{m} + \sqrt{d})) \\
z_{12} &= \|\Pi_U H \Pi_{V_\perp}\| \\
&= 0 \\
z_{21} &= \|\Pi_{U_\perp} H \Pi_V\| \\
&= \|\Pi_{U_\perp} \Sigma_{d-k}^{1/2} (\Sigma_{d-k}^{-1/2} H) \Pi_V\|
\end{aligned}$$

Now, in the above, $\Sigma_{d-k}^{-1/2} H \in \mathbb{R}^{d \times m}$, such that each of its entry is an independent sample from $\mathcal{N}(0,1)$. Right-multiplying it by Π_V makes it a matrix in a k-dimensional subspace of $\mathbb{R}^m$, such that each row is an independent vector from a spherical Gaussian. Using Corollary A.7, $\|\Sigma_{d-k}^{-1/2} H\| \leq O(\sqrt{d} + \sqrt{k}) \leq O(\sqrt{d})$ with probability at least $1 - \frac{0.01}{t}$. Also, $\|\Pi_{U_\perp} \Sigma_{d-k}^{1/2}\| \leq O(\gamma\sqrt{s_k(\Sigma_k)}) \leq O(\gamma)$. This gives us:

$$z_{21} \leq O(\gamma\sqrt{d}).$$

Since $a^2 > 2b^2$, we get the following by Lemma A.3.

$$\begin{aligned}
\|\text{Sin}(\Theta)(U, \hat{U})\| &\leq \frac{a z_{21} + b z_{12}}{a^2 - b^2 - \min\{z_{12}^2, z_{21}^2\}} \\
&\leq O\left(\frac{\gamma\sqrt{d}}{\sqrt{m}}\right)
\end{aligned}$$

Therefore, using Lemma A.4, and applying the union bound over all j, we get the required result. □

Let $\xi = O\left(\frac{\gamma\sqrt{d}}{\sqrt{m}}\right)$. We show that the projections of any reference point are close.

Corollary D.2. *Let $p_1, \ldots, p_q$ be the reference points as defined in Algorithm 2, and let Π and Π_j (for $1 \leq j \leq t$) be projections matrices as defined in Lemma D.1. Then*

$$\mathbb{P}\left[\forall i, j, \|(\Pi - \Pi_j)p_i\| \leq O(\xi(\sqrt{k} + \sqrt{\ln(qt)}))\right] \geq 0.9.$$

Proof. We know from Lemma D.1 that $\|\Pi - \Pi_j\| \leq \xi$ for all j with probability at least 0.95. For $j \in [t]$, let $\widehat{\Pi}_j$ be the projection matrix for the union of the j^{th} subspace and the subspace spanned by Σ_k. Lemma A.10 implies that with probability at least 0.95, for all i, j, $\|\widehat{\Pi}_j p_i\| \leq O(\sqrt{k} + \sqrt{\ln(qt)})$. Therefore,

$$\|(\Pi - \Pi_j)p_i\| = \|(\Pi - \Pi_j)\widehat{\Pi}_j p_i\| \leq \|\Pi - \Pi_j\| \cdot \|\widehat{\Pi}_j p_i\| \leq O(\xi(\sqrt{k} + \sqrt{\ln(qt)})).$$

Hence, the claim. □

The above corollary shows that the projections of each reference point lie in a ball of radius $O(\xi\sqrt{k})$. Next, we show that for each reference point, all the projections of the point lie inside a histogram cell with high probability. For notational convenience, since each point in Q is a concatenation of the projection of all reference points on a given subspace, for all i, j, we refer to $(0, \ldots, 0, Q^j_{(i-1)d+1}, \ldots, Q^j_{id}, 0, \ldots, 0) \in R^{qd}$ (where there are $(i-1)d$ zeroes behind $Q^j_{(i-1)d+1}$, and $(q-i)d$ zeroes after Q^j_{id}) as p^j_i.

Lemma D.3. *Let ℓ and λ be the length of a histogram cell and the random offset respectively, as defined in Algorithm 2. For each $1 \leq i \leq q$, define the following event.*

$$E_i \equiv \exists \omega \in \Omega : \left|\omega \cap \{p^1_i, \ldots, p^t_i\}\right| = t$$

Then $\mathbb{P}\left[E_i \cap \cdots \cap E_q\right] \geq 0.8$. Thus there exists $\omega \in \Omega$ that, such that all points in Q lie within ω.

Proof. Let $r = O(\xi(\sqrt{k} + \sqrt{\ln(qt)}))$. This implies that $\ell = 20rq$. The random offset could also be viewed as moving along a diagonal of a cell by $\lambda\ell\sqrt{dq}$. We know that with probability at least 0.8, for each i, all projections of reference point p_i lie in a ball of radius r. Fix an $i \in [q]$. Then

$$\mathbb{P}\left[\overline{E}_i\right] \leq \mathbb{P}\left[\frac{1}{20q} \geq \lambda \vee \lambda \geq \frac{19}{20q}\right] = \frac{1}{10q}.$$

Taking the union bound over all q and the failure of the event in Corollary D.2, we get the first part of the claim. Since p^j_i's are non-zero in disjoints sets of coordinates, the second part follows. □

Now, we analyse the sample complexity due to the private algorithm, that is, DP-histograms.

Lemma D.4. *For each $1 \leq i \leq q$, let ω_i be the histogram cell as defined in Algorithm 2. If $t \geq O\left(\frac{\log(1/\delta)}{\varepsilon}\right)$, then $\mathbb{P}\left[\forall i, \left|\omega_i \cap \{p^1_i, \ldots, p^t_i\}\right| = t\right] \geq 0.75$.*

Proof. Lemma D.3 implies that with probability at least 0.8, for each i, all projections of p_i lie in a histogram cell, that is, all points of Q lie in a histogram cell in Ω. Because of the error bound in Lemma A.15 and our bound on t, we see at least $\frac{q}{2}$ points in that cell with probability at least $1 - 0.05$. Therefore, by taking the union bound, the proof is complete. □

We finally show that the error of the projection matrix that is output by Algorithm 2 is small.

Lemma D.5. *Let $\widehat{\Pi}$ be the projection matrix as defined in Algorithm 2, and n be the total number of samples. If*

$$\gamma^2 \in O\left(\frac{\varepsilon\alpha^2 n}{d^2k^3\ln(1/\delta)} \cdot \min\left\{\frac{1}{k}, \frac{1}{\ln(k\ln(1/\delta)/\varepsilon)}\right\}\right),$$

$n \geq O(\frac{k\log(1/\delta)}{\varepsilon} + \frac{\ln(1/\delta)\ln(\ln(1/\delta)/\varepsilon)}{\varepsilon})$, and $q \geq O(k)$ the with probability at least 0.7, $\|\widehat{\Pi} - \Pi\| \leq \alpha$.

Proof. For each $i \in [q]$, let p^*_i be the projection of p_i on to the subspace spanned by Σ_k, $\widehat{p}_i$ be as defined in the algorithm, and p^j_i be the projection of p_i on to the subspace spanned by the j^{th} subset of X. From Lemma D.4, we know that all p^j_i's are contained in a histogram cell of length ℓ. This implies that $\|p^j_i - \widehat{p}_i\| \leq \ell\sqrt{dq}$. Since p^j_i's and p^*_i are contained in a ball of radius $\xi\sqrt{3d}$, it must be the case that $\|\widehat{p}_i - p^*_i\| \leq 2\ell\sqrt{dq}$.

Now, let $P = (p^*_1, \ldots, p^*_q)$ and $\widehat{P} = (\widehat{p}_1, \ldots, \widehat{p}_q)$. Then by above, $\widehat{P} = P + E$, where $\|E\|_F \leq 2\ell\sqrt{dq}$. Therefore, $\|E\| \leq 2\ell\sqrt{dq}$. Let $E = E_P + E_{\overline{P}}$, where E_P is the component of E in the subspace spanned by P, and $E_{\overline{P}}$ be the orthogonal component. Let $P' = P + E_P$. We will be analysing $\widehat{P}$ with respect to P'.

Now, with probability at least 0.95, $s_k(P) \in \Theta(\sqrt{k})$ due to our choice of q and using Corollary A.7, and $s_{k+1}(P) = 0$. So, $s_{k+1}(P') = 0$ because E_P is in the same subspace as P. Now, using Lemma A.2, we know that $s_k(P') \geq s_k(P) - \|E_P\| \geq \Omega(\sqrt{k}) > 0$. This means that P' has rank k, so the subspaces spanned by Σ_k and P' are the same.

As before, we will try to bound the distance between the subspaces spanned by P' and $\widehat{P}$. Note that using Lemma A.1, we know that $s_k(P') \leq s_k(P) + \|E_P\| \leq O(\sqrt{k})$.

We wish to invoke Lemma A.3 again. Let UDV^T be the SVD of P', and let $\hat{U}\hat{D}\hat{V}^T$ be the SVD of $\widehat{P}$. Now, for a matrix M, let Π_M denote the projection matrix of the subspace spanned by the columns of M. Define quantities a, b, z_{12}, z_{21} as follows.

$$\begin{aligned}
a &= s_{\min}(U^T \widehat{P} V) \\
&= s_{\min}(U^T P' V + U^T E_{\overline{P}} V) \\
&= s_{\min}(U^T P' V) && \text{(Columns of U are orthogonal to columns of $E_{\overline{P}}$)} \\
&= s_k(P') \\
&\in \Theta(\sqrt{k}) \\
b &= \|U_\perp^T \widehat{P} V_\perp\| \\
&= \|U_\perp^T P' V_\perp + U_\perp^T E_{\overline{P}} V_\perp\| \\
&= \|U_\perp^T E_{\overline{P}} V_\perp\| && \text{(Columns of $U_\perp$ are orthogonal to columns of P')} \\
&\leq \|E_{\overline{P}}\| \\
&\leq O(\ell\sqrt{d}q) \\
z_{12} &= \|\Pi_U E_{\overline{P}} \Pi_{V_\perp}\| \\
&= 0 \\
z_{21} &= \|\Pi_{U_\perp} E_{\overline{P}} \Pi_V\| \\
&\leq \|E_{\overline{P}}\| \\
&\leq O(\ell\sqrt{d}q)
\end{aligned}$$

Using Lemma A.3, we get the following.

$$\begin{aligned}
\|\mathrm{Sin}(\Theta)(U, \hat{U})\| &\leq \frac{az_{21} + bz_{12}}{a^2 - b^2 - \min\{z_{12}^2, z_{21}^2\}} \\
&\leq O\left(\ell\sqrt{dk}\right) \\
&\leq \alpha
\end{aligned}$$

This completes our proof. □

E Boosting

In this section, we discuss boosting of error guarantees of Algorithm 2. The approach we use is very similar to the well-known Median-of-Means method: we run the algorithm multiple times, and choose an output that is close to all other "good" outputs. We formalise this in Algorithm 3.

Now, we present the main result of this section.

Theorem E.1. *Let $\Sigma \in \mathbb{R}^{d \times d}$ be an arbitrary, symmetric, PSD matrix of rank $\geq k \in \{1, \ldots, d\}$, and let $0 < \gamma < 1$. Suppose Π is the projection matrix corresponding to the subspace spanned by the vectors of Σ_k. Then given*

$$\gamma^2 \in O\left(\frac{\varepsilon\alpha^2 n}{d^2 k^3 \ln(1/\delta)} \cdot \min\left\{\frac{1}{k}, \frac{1}{\ln(k\ln(1/\delta)/\varepsilon)}\right\}\right),$$

such that $\lambda_{k+1}(\Sigma) \leq \gamma^2 \lambda_k(\Sigma)$, for every $\varepsilon, \delta > 0$, and $0 < \alpha, \beta < 1$, there exists and (ε, δ)-DP algorithm that takes

$$n \geq O\left(\frac{k\log(1/\delta)\log(1/\beta)}{\varepsilon} + \frac{\log(1/\delta)\log(\log(1/\delta)/\varepsilon)\log(1/\beta)}{\varepsilon}\right)$$

samples from $\mathcal{N}(\vec{0}, \Sigma)$, and outputs a projection matrix $\widehat{\Pi}$, such that $\|\Pi - \widehat{\Pi}\| \leq \alpha$ with probability at least $1 - \beta$.

```
**Algorithm 3:** DP Approximate Subspace Estimator Boosted $\mathrm{DPASEB}_{\varepsilon,\delta,\alpha,\beta,\gamma,k}(X)$
**Input:** Samples $X_1, \dots, X_n \in \mathbb{R}^d$. Parameters $\varepsilon, \delta, \alpha, \beta, \gamma, k > 0$.
**Output:** Projection matrix $\widehat{\Pi} \in \mathbb{R}^{d \times d}$ of rank $k$.
Set parameters: $t \leftarrow C_3 \log(1/\beta)$        $m \leftarrow \lfloor n/t \rfloor$
Split $X$ into $t$ datasets of size $m$: $X^1, \dots, X^t$.
// Run DPASE $t$ times to get multiple projection matrices.
For $i \leftarrow 1, \dots, t$
    $\widehat{\Pi}_i \leftarrow \mathrm{DPASE}_{\varepsilon,\delta,\alpha,\gamma,k(X^i)}$
// Select a good subspace.
For $i \leftarrow 1, \dots, t$
    $c_i \leftarrow 0$
    For $j \in [t] \setminus \{i\}$
        If $\|\widehat{\Pi}_i - \widehat{\Pi}_j\| \leq 2\alpha$
            $c_i \leftarrow c_i + 1$
    If $c_i \geq 0.6t - 1$
        Return $\widehat{\Pi}_i$.
// If there were not enough good subspaces, return $\perp$.
Return $\perp$.
```

Proof. Privacy holds trivially by Theorem 1.2.

We know by Theorem 1.2 that for each i, with probability at least 0.7, $\|\widehat{\Pi}_i - \Pi\| \leq \alpha$. This means that by Lemma A.9, with probability at least $1 - \beta$, at least $0.6t$ of all the computed projection matrices are accurate.

This means that there has to be at least one projection matrix that is close to $0.6t - 1 > 0.5t$ of these accurate projection matrices. So, the algorithm cannot return $\perp$.

Now, we want to argue that the returned projection matrix is accurate, too. Any projection matrix that is close to at least $0.6t - 1$ projection matrices must be close to at least one accurate projection matrix (by pigeonhole principle). Therefore, by triangle inequality, it will be close to the true subspace. Therefore, the returned projection matrix is also accurate. □