Proceedings of Machine Learning Research vol 145:1–34, 2022

Stochastic and Private Nonconvex Outlier-Robust PCA

Tyler Maunu MAUNU@BRANDEIS.EDU
Brandeis University, Waltham, MA
Chenyu Yu CHENYU@PRINCETON.EDU
Princeton University, Princeton, NJ
Gilad Lerman LERMAN@UMN.EDU
University of Minnesota, Minneapolis, MN

Abstract

We develop theoretically guaranteed stochastic methods for outlier-robust PCA. Outlier-robust PCA seeks an underlying low-dimensional linear subspace from a dataset that is corrupted with outliers. We are able to show that our methods, which are variants of stochastic geodesic gradient descent over the Grassmannian manifold, converge and recover an underlying subspace in various regimes through the development of a novel convergence analysis. The main application of this method is an effective differentially private algorithm for outlier-robust PCA that uses a Gaussian noise mechanism within the stochastic gradient method. Our results emphasize the advantages of the nonconvex methods over another convex approach to solve Outlier-robust PCA in the differentially private setting. Experiments on synthetic and stylized data verify these results.

1. Introduction

Outlier-robust PCA (ORPCA) involves the problem of robustly estimating an underlying linear subspace from data in the presence of large amounts of corrupted data. While many solutions have been proposed for this problem, some particularly effective methods involve nonconvex energy minimization (Maunu et al., 2019). However, these methods require generic conditions on the full dataset, and it is not clear how they behave in the presence of *stochastic gradients*, since they typically require good initialization and control over where the iterates lie.

This work develops a deeper understanding of how nonconvex methods for ORPCA interact with stochastic gradients. Past studies have mainly looked at recovery limits of such methods (Lerman and Maunu, 2018b), both in terms of percentages of corrupted data as well as their associated statistics. In the current work, we show that it is possible to extend the results to the stochastic setting while maintaining robustness guarantees.

As an important application, we show that specific choices of stochastic gradients lead to *differential privacy*. Private algorithms provide an important way to gain insight from sensitive data. As a framework, differential privacy has harkened in a new era in the study of privacy and its interaction with data science and machine learning (Dwork et al., 2006; Dwork, 2008). To make an algorithm differentially private, one typically incorporates some sort of noise mechanism. This noise mechanism is applied to either the data itself or within the algorithm to limit the influence any single point can have on the output. In this paper, we focus on differentially private gradient descent algorithms, which use noisy gradients at each iteration to achieve differential privacy.

While differential privacy may be simple to include within an algorithm, it is less straightforward to guarantee how accurate the algorithm will be, especially in nonconvex settings. Recently, there has been work on empirical risk minimization by differentially private methods, which show that it is possible to achieve fast estimation and optimization rates with differentially private algorithms (Bassily et al., 2014; Talwar et al., 2014; Bassily et al., 2019). Such results typically focus on the convex setting, but some recent work has studied such algorithms in the nonconvex setting as well (Wang et al., 2019). While these results are quite general, they do not capture the intricacies of the analysis of robust methods. That is, especially in the setting of ORPCA, robust methods are concerned with *recovery results*, where under various conditions on a corrupted dataset, an algorithm can still recover some unknown underlying structure. Especially in nonconvex recovery problems, it is not clear how the stochastic nature of the private algorithms interacts with existing recovery guarantees. Due to our generic guarantees for stochastic gradient methods, we are able to guarantee recovery for a differentially private method.

1.1. Background

Suppose that we observe a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subset \mathbb{R}^D$. The classical problem of principal component analysis (PCA) seeks the r directions of maximum variance within this dataset, where r is a parameter chosen by the user. Equivalently, one can also try to find a linear subspace that spans these directions. It is therefore convenient to encode PCA as a problem over the Grassmannian manifold of r-dimensional linear subspaces in $\mathbb{R}^D$, $G(D, r)$. Throughout the paper, we also consider the optimization over orthogonal bases for $L \in G(D, r)$: each element of $G(D, r)$ can be spanned by the columns of a semiorthogonal matrix in $O(D, r) := \{\boldsymbol{V} \in \mathbb{R}^{D \times r} : \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r\}$.

In this language, PCA solves the geometric problem

$$\min_{\boldsymbol{V} \in O(D,r)} \frac{1}{N} \sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\boldsymbol{x}_i\|^2, \tag{PCA}$$

where $\boldsymbol{V}\boldsymbol{V}^\top$ is the orthogonal projection matrix onto $\mathrm{Sp}(\boldsymbol{V})$. PCA thus finds the subspace which minimizes the sum of squared distances between points and the subspace.

PCA is not outlier-robust due to the use of squared error. A typical way to robustify it is to remove the square, which results in the following formulation which we refer to as Grassmannian Least Absolute Deviations (GLAD). :

$$\min_{\boldsymbol{V} \in O(D,r)} \frac{1}{N} \sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\boldsymbol{x}_i\| =: F(\boldsymbol{V}; \mathcal{X}). \tag{GLAD}$$

Many methods have been proposed to solve this nonconvex and nonsmooth problem, and they are reviewed in Lerman and Maunu (2018b).

To give a high-level overview of our results, we will briefly discuss the two areas that it straddles. First, the primary result of our analysis is guarantees for a nonconvex, stochastic method for ORPCA. Typically, ORPCA algorithms assume an inlier-outlier model, $\mathcal{X} = \mathcal{X}_{\text{in}} \cup \mathcal{X}_{\text{out}}$, where $\mathcal{X}_{\text{in}}$ lie on a low-dimensional subspace $L_\star$, and the outliers $\mathcal{X}_{\text{out}}$ are corrupted to not lie on this subspace. The goal is to recover $L_\star \in G(D, r)$, or $\boldsymbol{V}_\star \in O(D, r)$

such that $\mathrm{Sp}(\boldsymbol{V}_\star) = L_\star$. For simplicity, we assume that the data is centered, so that we search for a linear subspace. Throughout the paper, we also make the simplifying assumption that $\mathcal{X} \subset S^{D-1}$, where S^{D-1} is the sphere in $\mathbb{R}^D$, so that the function F is 1-Lipschitz. This can be achieved by first normalizing all points to the sphere, which has robustifying characteristics to adversarial outliers (Maunu and Lerman, 2019).

Second, the important application of our results involves differential privacy (Dwork and Roth, 2013). A randomized algorithm $\mathcal{A}$, which takes in an input x and gives back a random output, is (ϵ, δ)-differentially private if, for all $\mathcal{S} \subseteq \mathrm{Range}(\mathcal{A})$ and for all datasets x, y that only differ in at most one data point, $P[\mathcal{A}(x) \in \mathcal{S}] \leq e^\epsilon P[\mathcal{A}(y) \in \mathcal{S}] + \delta$. Two common ways to make a first-order algorithm private include adding noise to data or adding noise to the gradients. In this work, since we study stochastic gradient methods for outlier-robust PCA, the recovery guarantees we prove naturally extend to the private setting.

1.2. Contributions

We derive the following results for ORPCA with large N:

1. We present stochastic versions of the geodesic gradient descent (GGD) algorithm, which results in the Noisy GGD (NGGD), Stochastic GGD (SGGD), and Noisy Stochastic GGD (NSGGD) methods. We give theorems guaranteeing linear convergence and subspace recovery by these three methods. Our results are the first nonconvex convergence guarantees for stochastic gradient descent in the least absolute deviations framework.

2. With specifically chosen noise parameters, we demonstrate that these methods are differentially private, and we refer to the resulting algorithms as dp-GGD and dp-SGGD, respectively. We compare these private algorithms to convex methods for differentially private outlier-robust PCA based on the REAPER problem (Lerman et al., 2015), (dp-REAP). In this setting, we extend past results on differentially private convex empirical risk minimization to give subspace recovery guarantees for the dp-REAP algorithms under generic conditions.

3. By comparing our theoretical results for the differentially private methods, we demonstrate a distinct advantage in the differentially private setting for dp-(S)GGD over dp-REAP. The nonconvex dp-(S)GGD algorithm converges at a linear rate while the convex dp-REAP methods converge at a sublinear rate, meaning that one can obtain a much more accurate approximation to the underlying subspace in less iterations. In terms of best approximations while still maintaining privacy, we achieve approximation errors (in terms of distance to ground truth squared) that are $O(N^{-1})$ for the convex methods and errors on the order of $O(2^{-N^\tau})$ for the nonconvex methods, where τ is some constant in $(0, 2)$ that depends on the statistics of the dataset.

4. Experiments on synthetic and stylized data emphasize the theoretical results of this paper. In particular, they demonstrate the advantage in terms of speed and accuracy for the nonconvex methods, and in particular demonstrate distinct advantages for the dp-SGGD method.

1.3. Review of Directly Related Work

For a comprehensive review of the many methods used for ORPCA, we direct the reader to Lerman and Maunu (2018b). Perhaps one of the most popular frameworks for ORPCA uses least absolute deviations. Originating with the study of robust orthogonal regression in Osborne and Watson (1985); Späth and Watson (1987), it was considered for ORPCA in Ding et al. (2006). More recent studies by Zhang and Lerman (2014); Lerman et al. (2015); Lerman and Maunu (2018a); Maunu et al. (2019) have demonstrated the considerable advantages of this program. This problem is distinct from what is called Robust PCA (RPCA), which considers sparse corruptions (Chandrasekaran et al., 2011; Candès et al., 2011).

The nonconvex method we propose is based on optimization on the Grassmannian manifold (Edelman et al., 1999). Manifold optimization has recently been of great interest for the machine learning community (Zhang and Sra, 2016).

Differential privacy has become the preeminent way of protecting sensitive data (Dwork and Roth, 2013). There has been a recent surge of work examining how differential privacy affects the accuracy of various methods (Bassily et al., 2014, 2019). Some recent work has been devoted to considering differentially private methods for PCA (Chaudhuri et al., 2013; Hardt and Price, 2014; Jiang et al., 2016).

1.4. Notation

We let $\sigma_j(\cdot)$ denote the jth singular value of a matrix. For measuring subspace approximation, we use a distance metric on the Grassmannian. A typical metric is $d(L_1, L_2) = \sqrt{\sum_{j=1}^{r} \theta_j^2}$, where θ_j are the principal angles between L_1 and L_2. For our later analysis of the nonconvex method, for $\boldsymbol{V}, \boldsymbol{V}' \in O(D, r)$, which are bases for two elements of $G(D, r)$, it is more convenient to work with the squared metric $d_r^2(\boldsymbol{V}, \boldsymbol{V}') = 1 - \sigma_r(\boldsymbol{V}^\top \boldsymbol{V}')$, which for subspaces that are close together is on the order of $1/2$ times the largest principal angle squared between $\mathrm{Sp}(\boldsymbol{V})$ and $\mathrm{Sp}(\boldsymbol{V}')$ (specifically, it is $1 - \cos(\theta_1)$). We denote $B_{d_r^2}(\boldsymbol{V}, \rho)$ to be the ball of radius ρ with respect to d_r^2.

2. Stochastic Algorithms to Minimize GLAD

In this paper, we propose to use stochastic gradient descent to directly minimize (GLAD). This extends the existing framework for ORPCA studied by Maunu et al. (2019), where the authors proposed to use vanilla geodesic gradient descent (GGD). Section 2.1 reviews the GGD method used to minimize (GLAD). Then, Section 2.2 discusses modifications of this method to include noisy and minibatch gradients, which result in stochastic GGD methods.

2.1. Geodesic Gradient Descent

One can directly optimize (GLAD) over the Grassmannian manifold using geometric methods. Past algorithms that accomplish this with some theoretical guarantees (despite the nonconvex setting) include IRLS (Lerman and Maunu, 2018a) and GGD (Maunu et al., 2019). On top of frequently being more accurate than their convex counterparts, these methods are also faster than convex methods, since nonconvex methods work with a $D \times r$

optimization variable rather than the typical $D \times D$ variable that replaces the orthogonal projection $\boldsymbol{V}\boldsymbol{V}^T$.

We briefly review GGD. Since $G(D,r)$ forms a Riemannian manifold, the Riemannian gradient of the energy function in (GLAD) is

$$\nabla F(\boldsymbol{V};\mathcal{X}) = \frac{1}{|\mathcal{X}|}\boldsymbol{Q}_{\boldsymbol{V}} \sum_{\boldsymbol{x}\in\mathcal{X}} \frac{\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{V}}{\|\boldsymbol{Q}_{\boldsymbol{V}}\boldsymbol{x}\|}, \tag{2.1}$$

where $\boldsymbol{V} \in O(D,r)$ is a matrix whose columns span L, $\boldsymbol{Q}_{\boldsymbol{V}} = \boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top$ projects the gradient to the tangent space of $G(D,r)$ and $|\mathcal{X}|$ denotes the number of points in the set $\mathcal{X}$. Geodesic gradient descent (GGD) then takes the form $\boldsymbol{V}_{k+1} = \mathsf{Exp}_{\boldsymbol{V}_k}(-\eta_k \nabla F(\boldsymbol{V}_k;\mathcal{X}))$ (where Exp is the exponential map). For a complete discussion of this iteration and associated concepts related to the geometry of $G(D,r)$, see Edelman et al. (1999); Maunu et al. (2019).

2.2. Stochastic Geodesic Gradient Descent Methods

In terms of optimization, the main innovation in this work is to consider stochastic gradient methods for (GLAD). One specific stochastic gradient one may consider is the addition of Gaussian noise, which enhances privacy. To go beyond this setting, we also consider stochasticity due to *minibatching*. While the addition of stochastic gradients is a small modification of the original GGD method, it is entirely nontrivial to extend convergence and recovery analysis to the stochastic setting (see Section 3).

We first describe a version of GGD which uses noisy gradients. Let $\boldsymbol{B}_k \in \mathbb{R}^{D\times r}$ whose entries are i.i.d.$\sim \mathcal{N}(0,\sigma^2)$. The noisy GGD (NGGD) iteration is given by

$$\boldsymbol{V}_{k+1} = \mathcal{P}_{O(D,r)}(\boldsymbol{V}_k - \eta_k(\nabla F(\boldsymbol{V}_k;\mathcal{X}) + \boldsymbol{B}_k)), \tag{NGGD}$$

where $\mathcal{P}_{O(D,r)}$ is the projection operator that solves $\mathcal{P}_{O(D,r)}(\boldsymbol{A}) = \operatorname{argmin}_{\boldsymbol{V}\in O(D,r)} \|\boldsymbol{V}-\boldsymbol{A}\|^2$. This is an example of the orthogonal Procrustes problem (Gower and Dijksterhuis, 2004), and it can be solved via the SVD or polar decomposition (Fan and Hoffman, 1955). This iteration is referred to as Noisy Geodesic Gradient Descent (NGGD).

We can also use stochastic estimates of $\nabla F(\boldsymbol{V}_k;\mathcal{X})$ to add further "noise" to the gradient. We call such a method noisy stochastic geodesic gradient descent, NSGGD, which is defined by the iteration

$$\tilde{\boldsymbol{V}}_{k+1} = \mathcal{P}_{O(D,r)}(\tilde{\boldsymbol{V}}_k - \eta_k(\boldsymbol{G}_k + \boldsymbol{B}_k)). \tag{NSGGD}$$

Here, $\boldsymbol{G}_k$ is an estimate of the gradient at $\tilde{\boldsymbol{V}}_k$. When using minibatch stochastic gradients, we let $\boldsymbol{G}_k = \mathsf{grad}F(\boldsymbol{V}_k;\mathcal{X}^k)$, where $\mathcal{X}^k \subset \mathcal{X}$. We refer to the method with minibatch stochastic gradients and zero noise as SGGD.

These methods have many potential applications. First, the minibatch SGGD method allows for less per-iteration complexity than that of GGD, where SGGD has complexity $O(BDd)$ per iteration (where $|\mathcal{X}^k| = B$) and GGD has complexity of $O(NDd)$ per iteration. Furthermore, the addition of noise allows for the potential development of Langevin-like algorithms on the Grassmannian. Finally, as we discuss later, when the noise has sufficiently large variance, we can show that the resulting method is differentially private.

3. Theory

In the following sections we present our theoretical results for NGGD, SGGD, and NSGGD. In particular, we prove convergence *and* subspace recovery results for these methods.

First, in Section 3.1, we recall a result from Maunu et al. (2019), which shows that PCA gives a good initial approximation to the underlying subspace with high probability. After this, Section 3.2 gives an iteration complexity and approximation result for NGGD. Then, Section 3.3 gives an iteration complexity and approximation result for SGGD as well as a convergence and recovery theorem for NSGGD. The proofs of convergence for these differentially private methods require nontrivial extensions of the past proofs of convergence for GGD seen in Maunu et al. (2019). After this, we finish in Section 3.4 by showing how one can extend these approximation guarantees to achieve linear convergence of the NGGD, SGGD, and NSGGD algorithms with a geometrically diminishing step size scheme. For brevity, all proofs are left to the Appendix.

The results in these sections represent the main theoretical innovation of this work. Similar to the analysis of the deterministic GGD method, the strategy is to prove, under a general condition called *stability*, 1) good initialization by some means, and 2) convergence of the nonconvex stochastic gradient method.

3.1. Initialization by PCA

Our nonconvex methods require initialization in a sufficiently small neighborhood of the true subspace spanned by $\boldsymbol{V}_\star$. To accomplish this, we initialize NGGD, SGGD, and NSGGD using a PCA subspace. Later, in the case of differentially private methods, we show that one can also initialize with differentially private PCA. The main result for initialization follows.

Theorem 1 (Maunu et al. (2019)) *If*

$$\mathcal{S}_\rho^{\mathsf{PCA}}(\mathcal{X}) := 2\sin(\arccos(1-\rho))\,\mu_r(\boldsymbol{X}_{\text{in}}\boldsymbol{X}_{\text{in}}^\top) - \|\boldsymbol{X}_{\text{out}}\|_2^2 > 0, \tag{3.1}$$

then $d_r^2(L_{PCA}, L_\star) < \rho$.

Here, $\mu_j(\boldsymbol{A})$ is the jth largest eigenvalue of a given matrix $\boldsymbol{A}$.

3.2. Noisy GGD

Towards a complete theory for private, nonconvex robust subspace recovery, we first prove an iteration complexity and approximation result for NGGD. This section and the following are mainly inspired by the analysis in Zhou et al. (2020). Following the analysis in Maunu et al. (2019), the goal is to show that the sequence $\sigma_r(\boldsymbol{V}_\star^\top \boldsymbol{V}_k)$ forms a sequence that rapidly increases with k.

For the convergence of GGD in Maunu et al. (2019), the key idea is the development of the stability statistic, which is defined as

$$\mathcal{S}_\rho(\mathcal{X}) = (1-\rho)\mu_r\left(\frac{1}{N}\sum_{\mathcal{X}_{\text{in}}}\frac{\boldsymbol{x}\boldsymbol{x}^\top}{\|\boldsymbol{x}\|}\right) - \max_{\boldsymbol{V}\in O(D,d)}\sigma_1\left(\nabla F(\boldsymbol{V};\mathcal{X}_{\text{out}})\right))$$

$$= (1-\rho)\mathcal{P}(\mathcal{X}_{\text{in}}) - \mathcal{A}(\mathcal{X}_{\text{out}}).$$

Note that our parametrization of this statistic is slightly different from that of Maunu et al. (2019), where we use $1-\rho$ instead of $\cos(\gamma)$. Under *stability*, or the assumption that $\mathcal{S}_\rho(\mathcal{X}) > 0$, Maunu et al. (2019) prove local convergence of GGD given initialization in $B_d(\boldsymbol{V}_\star, \arccos(1-\rho)) \equiv B_{d_r^2}(\boldsymbol{V}_\star, \rho)$ for a different metric d. As we can see, the statistic $\mathcal{S}_\rho(\mathcal{X})$ is a difference between an inlier term and an outlier term.

In the following theorem, we prove convergence of NGGD when $\mathcal{S}_\rho(\mathcal{X}) > 0$ as long as $\boldsymbol{V}_0 \in B_{d_r^2}(\boldsymbol{V}_\star, \rho/2)$ – notice that the noisy method requires some extra wiggle room in terms of the initialization to ensure that iterations do not leave the neighborhood $B_{d_r^2}(\boldsymbol{V}_\star, \arccos(1-\rho))$.

Theorem 2 *Assume that $\mathcal{S}_\rho(\mathcal{X}) > 0$, NGGD is initialized at $\boldsymbol{V}_0 \in B_{d_r^2}(\boldsymbol{V}_\star, \rho/2)$ with a constant step size $\eta_k = s = c_1 a/T^\nu$, $0.5 < \nu < 1$, and is run for T iterations, where*

$$T > \mathcal{F}_1(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0), \lambda),$$

for $\mathcal{F}_1$ defined in (B.13) that depends on all parameters. Then NGGD yields a final iterate $\boldsymbol{V}_T \in B_{d_r^2}(\boldsymbol{V}_\star, a)$ with probability at least $1-2\lambda$.

By Theorem 1, PCA initialization achieves the proper initialization with high probability when the condition $\mathcal{S}_\rho^{\mathsf{PCA}}(\mathcal{X}) > 0$ holds. Theorem 2 states that, effectively, as long as the number of iterations is larger than $\mathcal{F}_1(c)$, the NGGD final iterate lies in $B_{d_r^2}(\boldsymbol{V}_\star, c\rho)$ with high probability. We will show in Section 3.4 how one can turn this into a linear convergence result.

3.3. Noisy Stochastic GGD

We first present a novel analysis of minibatch SGGD for solving (GLAD).

To this end, we assume minibatches $\mathcal{X}^k$, $1 \le k \le T$, of size B that are drawn from $\mathcal{X}$ with replacement. We can separate each minibatch into inlier and outlier components $\mathcal{X}_{\text{in}}^k$ and $\mathcal{X}_{\text{out}}^k$, respectively. Much in the same way that one can analyze GGD and NGGD, we analyze SGGD through the use of stability statistics. For SGGD, the main difference is now each minibatch has an associated stability statistic, $\mathcal{S}_\rho(\mathcal{X}^k) = (1-\rho)\mathcal{P}(\mathcal{X}_{\text{in}}^k) - \mathcal{A}(\mathcal{X}_{\text{out}}^k)$. As there are N^B subsets $\mathcal{X}^k$, we get a range of stability statistics, some of which are positive and some of which are negative. Now, instead of assuming that $\mathcal{S}_\rho(\mathcal{X}) > 0$, we assume that for a minibatch selected uniformly at random from $\mathcal{X}$ without replacement,

$$\mathbb{E}_{\mathcal{X}^k} \mathcal{S}_\rho(\mathcal{X}^k) = \mathcal{S}_{\rho,\mathbb{E}} > 0. \tag{3.2}$$

Theorem 3 *Assume that $\mathcal{S}_{\rho,\mathbb{E}} > 0$, SGGD is initialized at $\boldsymbol{V}_0 \in B_{d_r^2}(\boldsymbol{V}_\star, \rho/2)$ with a constant step size $\eta_k = s = c_1 a/T^\nu$, $0.5 < \nu < 1$, and is run for T iterations, where*

$$T > \mathcal{F}_2(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0), \lambda),$$

for $\mathcal{F}_2$ defined in (B.15) that depends on all other parameters. Then SGGD yields a final iterate $\boldsymbol{V}_T \in B_{d_r^2}(\boldsymbol{V}_\star, a)$ with probability at least $1-2\lambda$.

To additionally prove convergence of NSGGD, we must also control the noise throughout the iterations. This result essentially combines Theorems 2 and 3. As before, this theorem states that in a number of iterations $> \mathcal{F}_3(c)$, the NSGGD final iterate lies in $B_{d_r^2}(\boldsymbol{V}_\star, c\rho)$ with high probability.

Theorem 4 *Assume that $\mathcal{S}_{\rho,\mathbb{E}} > 0$, NSGGD is initialized at $\boldsymbol{V}_0 \in B_{d_r^2}(\boldsymbol{V}_\star, \rho/2)$ with a constant step size $\eta_k = s = c_1 a/T^\nu$, $0.5 < \nu < 1$, and is run for T iterations, where*

$$T > \mathcal{F}_3(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0), \lambda),$$

for $\mathcal{F}_3$ defined in the Appendix that depends on all other parameters. Then NSGGD yields a final iterate $\boldsymbol{V}_T \in B_{d_r^2}(\boldsymbol{V}_\star, a)$ with probability at least $1 - 4\lambda$.

Finally, we demonstrate how one might hope to have $\mathcal{S}_{\rho,\mathbb{E}} > 0$ with a simple model. Using the bound from (Maunu and Lerman, 2019, Appendix C), for a sample from $\mathcal{X}$ without replacement of size B, which we denote by $\mathcal{X}^B$

$$\begin{aligned} \mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X}) &= \mathbb{E}\Big((1-\rho)\mu_r \left(\frac{1}{B}\sum_{\mathcal{X}_{\text{in}}^B} \frac{\boldsymbol{x}\boldsymbol{x}^\top}{\|\boldsymbol{x}\|}\right) - \max_{\boldsymbol{V}\in O(D,d)} \sigma_1\left(\nabla F(\boldsymbol{V}; \mathcal{X}_{\text{out}}^B)\right)\Big) \\ &\geq \mathbb{E}\left[(1-\rho)\mu_r \left(\frac{1}{B}\sum_{\mathcal{X}_{\text{in}}^B} \frac{\boldsymbol{x}\boldsymbol{x}^\top}{\|\boldsymbol{x}\|}\right)\right] - \mathbb{E}\left[\sqrt{\#(\mathcal{X}_{\text{out}}^B)}\|\boldsymbol{X}_{\text{out}}^B\|_2\right]. \end{aligned} \tag{3.3}$$

We thus believe that one can show that this holds for a range of examples by bounding expected eigenvalues of sub-Gaussian random matrices. This could be done in a similar way to what is done in (Maunu and Lerman, 2019, Theorem 14), using bounds seen for example in (Vershynin, 2012, Section 5.3).

3.4. Linear Convergence Analysis

As we commented in the previous sections, in a constant number of iterations, $\mathcal{F}_1(c)$ for NGGD, $\mathcal{F}_2(c)$ for SGGD, and $\mathcal{F}_3(c)$ for NSGGD, the stochastic GGD algorithms converge to $B_{d_r^2}(\boldsymbol{V}_\star, c\rho)$. Setting $c = 1/2$, these methods have yielded final estimates twice as close to $\boldsymbol{V}_\star$ in a constant number of iterates.

Using this fact, the following theorem guarantees linear convergence of the stochastic GGD algorithms. To accomplish this, we use a geometrically diminishing step size. That is, we run the algorithm with a constant step size s for a sufficient number of iterations. Then, the algorithm is *restarted* with a constant step size cs for some fraction $c \in (0, 1)$. This restarting procedure is then repeated R times. This is similar to the strategy used in Maunu et al. (2019) to prove linear convergence of GGD.

Theorem 5 *Suppose that one of the stochastic GGD algorithms is run for R restarts and $\mathcal{S} > 0$, where $\mathcal{S} = \mathcal{S}_\rho(\mathcal{X})$ for NGGD and $\mathcal{S} = \mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})$ for SGGD and NSGGD. Suppose that in the first run of the algorithm (out of all the restarts), the step size is $s = c_1 a T_1^{-\nu}$, $0.5 < \nu < 1$ and the number of iterations is $T_1 = O(\mathcal{F}_j(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)))$, where $j = 1$ for NGGD, $j = 2$ for SGGD, and $j = 3$ for NSGGD. Suppose further that the step size for the lth restart is $s/2^{l-1}$ for $T_l = \mathcal{F}_j(1/2)$ iterations. Then, with probability at least $1 - 2R\lambda$ (or $1 - 4R\lambda$ for NSGGD), the output of the Rth restart, $\hat{\boldsymbol{V}}$, satisfies $\hat{\boldsymbol{V}} \in B_{d_r^2}(\boldsymbol{V}_\star, a/2^R)$.*

We see that this theorem guarantees an approximation that decreases at an exponential rate over the number of restarts. There is an interplay between the recovery probability and error guarantee: while the approximation decreases in R, the probability of success also decreases in R. However, the error decreases exponentially while the probability only decreases linearly. Therefore, we can specify a small recovery error E and then offset this with a sufficiently small parameter $\lambda = O(\log(1/E)$, which would then increase the number of iterations in Theorems 2-5.

4. Application: Differential Privacy

In both the NGGD and NSGGD methods, if the noise variance is sufficiently large, then the methods become differentially private. We guarantee the privacy of these methods in the following theorem.

Theorem 6 (Differential Privacy of NGGD and NSGGD) *There exists a constant c such that for any $\varepsilon < cT$, if $\sigma^2 \geq \frac{cT\log^2(1/\delta)}{\varepsilon^2 N^2}$, then NGGD run for T iterations is (ϵ, δ) differentially private. On the other hand, if the batch size is B, there exist constants c_1 and c_2 such that for any $\varepsilon < c_1 q^2 T$, if $\sigma^2 \geq c_2 \frac{(B/N)^2 T\log(1/\delta)}{\varepsilon^2 N^2}$, then NSGGD run for T iterations is (ϵ, δ) differentially private.*

The proof of differential privacy for such stochastic first-order methods is standard and follows Bassily et al. (2014); Talwar et al. (2014); Jayaraman et al. (2018). With the noise variances as specified in Theorem 6, we refer to the NGGD algorithm as dp-GGD and to NSGGD as dp-SGGD. When writing statements that apply to either dp-GGD or dp-SGGD, we will refer to dp-(S)GGD.

In the following sections, we examine the implications of our recovery results in the differentially private setting. First, Section 4.1 discusses how to initialize NGGD and NSGGD in a private way. Then, Section 4.2 explains how the results for NGGD and NSGGD translate to the differentially private setting. Lastly, in Section 4.3 we present convex differentially private methods based on REAPER (Lerman et al., 2015), which gives an important baseline for subspace recovery based on differentially private convex empirical risk minimization. In particular, we extend convergence results for convex empirical risk minimization to the case of the REAPER algorithm and show that these have implications for subspace recovery.

4.1. PCA Initialization

Throughout the paper, we refer to dp-PCA as the output of the differentially private PCA method of Jiang et al. (2016). Combining the previous result in Theorem 1 with a result of Jiang et al. (2016), we obtain the following theorem.

Theorem 7 *If N is sufficiently large and $\mathcal{S}^{\mathsf{PCA}}_{\rho/2}(\mathcal{X}) > 0$, we have that the output of dp-PCA, $\boldsymbol{V}_{dp-PCA}$, lies in $B_{d_r^2}(\boldsymbol{V}_\star, \rho/2)$ with high probability.*

4.2. Approximation for dp-(S)GGD

Notice that, in order for the conditions of Theorem 6 to be satisfied, we need the total number of iterations to be bounded as $T = O(N^2\epsilon^2)$. To get a sense of the number of

restarts we can take, we note that this implies $\sum_{l=1}^{R} T_l = O(N^2\epsilon^2)$. If we take $T_l = (\epsilon N)^\alpha$ for $0 < \alpha < 2$ for all $l = 1, \ldots, R$, the conditions of Theorem 5 are satisfied once N is sufficiently large. Therefore, we can take $R = O((\epsilon N)^{2-\alpha})$, which yields a dp-(S)GGD estimator with accuracy on the order of $1/2^{(\epsilon N)^{2-\alpha}}$, which decreases exponentially in N. Taking this all together, we have the following corollary of Theorems 5 and 6.

Corollary 8 *Running the dp-(S)GGD algorithm as in Theorem 5 for the maximum number of restarts R while still maintaining privacy in Theorem 6 yields a final iterate $\hat{\boldsymbol{V}}$ such that $d_r^2(\hat{\boldsymbol{V}}, \boldsymbol{V}_\star) = O(\rho/2^{(\epsilon N)^{2-\alpha}})$.*

4.3. Differentially Private REAPER Algorithms

One could also attempt to relax (GLAD) and solve a surrogate convex problem instead. A popular relaxation for this task is the REAPER relaxation of Lerman et al. (2015). In this section, we present a simple differentially private version of this method. We can directly apply existing empirical risk minimization results to this problem (Bassily et al., 2014, 2019) to yield subspace recovery guarantees. This will give us a baseline that demonstrates the superiority of the nonconvex method.

The REAPER program (Lerman et al., 2015) solves (GLAD) by relaxing the nonconvex constraints that $\boldsymbol{P}_L$ is an orthoprojection:

$$\min_{\boldsymbol{P} \in H} \frac{1}{N} \sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}_i\|, \ \mathcal{H} := \{\boldsymbol{P} : \boldsymbol{0} \preceq \boldsymbol{P} \preceq \boldsymbol{I}, \ \mathrm{Tr}(\boldsymbol{P}) = r\}. \tag{REAP}$$

This is a convex program, and so (REAP) can be solved by an array of standard convex optimization algorithms. Since $\mathcal{X} \subset S^{D-1}$, $G(\boldsymbol{P}; \mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}_i\|$ is 1-Lipschitz. Since the objective in (REAP) is not smooth, one must use subgradient based methods (Clarke, 1990). We use the following subgradient of REAPER:

$$\nabla G(\boldsymbol{P}; \mathcal{X}) = - \sum_{\substack{\boldsymbol{x} \in \mathcal{X} \\ \|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\| > 0}} \frac{(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}\boldsymbol{x}^T + \boldsymbol{x}\boldsymbol{x}^T(\boldsymbol{I} - \boldsymbol{P})}{2\|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\|}. \tag{4.1}$$

While Lerman et al. (2015) proposes to solve this problem using an iteratively reweighted least squares method, we instead opt to study first-order methods. The first method we consider is gradient descent, and the second is a mirror descent. To make these methods differentially private, we again use the Gaussian mechanism and add noise to the gradient. Since past work has demonstrated advantages for considering stochastic first-order methods when making convex algorithms private (Abadi et al., 2016; Bassily et al., 2019), we also give stochastic versions of each algorithm. These convex optimization methods for the REAPER problem are differentially private by the previous arguments of Bassily et al. (2014); Talwar et al. (2014).

Since our primary focus is on the nonconvex method, and some nonprivate versions of the convex methods were previously explored by Goes et al. (2014), we leave the exact formulation of these methods to the Appendix. In the Appendix, we outline 4 differentially private algorithm for solving this REAPER program: Differentially Private Gradient Descent (dp-GD-REAP), Differentially Private Stochastic Gradient Descent (dp-SGD-REAP),

Differentially Private Mirror Descent (dp-MD-REAP), and Differentially Private Stochastic Mirror Descent (dp-SMD-REAP).

Previous work on optimization with differential privacy has focused on differentially private *empirical risk minimization* (Bassily et al., 2014; Talwar et al., 2014; Bassily et al., 2019). In this general set-up, one wishes to minimize the empirical surrogate for the population loss. In the non-stochastic setting, we can use the main theorem of Talwar et al. (2014) for both dp-GD-REAP and dp-MD-REAP. Indeed, if one uses the mirror map $\Psi(\cdot) = \|\cdot\|^2/2$, then the algorithm just becomes dp-GD-REAP, whereas if one uses the negative von Neumann entropy, it yields dp-MD-REAP. The following theorem gives our main approximation result for the nonstochastic and stochastic dp-REAP algorithms. In both cases, we show that the approximation error for these private methods is on $O(1/N)$, rather than exponential like the dp-(S)GGD algorithms. In contrast to Bassily et al. (2019), this theorem does not resort to smoothing the cost function and instead uses the optimization rate for subgradient descent.

Theorem 9 *Let* D *be the diameter of the constraint set* $\mathcal{H}$*. Then, if dp-GD-REAP or dp-MD-REAP is run for* $T = O(\epsilon^2 N^2)$ *and yields the estimator* $\bar{\boldsymbol{P}}$*, we have*

$$\mathbb{E}G(\bar{\boldsymbol{P}}; \mathcal{X}) - \min_{\boldsymbol{P}} G(\boldsymbol{P}; \mathcal{X}) \lesssim \frac{\mathsf{D}\log(N/\delta)}{\epsilon N}, \tag{4.2}$$

where the expectation is taken over the randomness of the algorithm. On the other hand, for dp-SGD-REAP and dp-SMD-REAP, if the noise variance is $\sigma^2 = c_2 \frac{B^2 T \log(1/\delta)}{\varepsilon^2 N^2}$*, then*

$$\mathbb{E}G(\bar{\boldsymbol{P}}; \mathcal{X}) - \min_{\boldsymbol{P}} G(\boldsymbol{P}; \mathcal{X}) \leq \sqrt{1 + c_2 B^2 \log(1/\delta)}\mathsf{D}\frac{1}{\epsilon N},$$

where the expectation is taken over the randomness of the algorithm and D *is the diameter of the constraint set with respect to the appropriate geometry.*

The proof for the nonstochastic methods is just Theorem 3.2 of Talwar et al. (2014), and the proof for the stochastic methods is given in Appendix B.4.

4.3.1. Implications for Subspace Recovery

We show that the approximate minimization guaranteed by Theorem 9 yields in a generic setting approximate subspace recovery, or for REAPER, approximate recovery of $\boldsymbol{P}_\star = \boldsymbol{P}_{L_\star}$.

We recall the following *permeance*, *alignment*, and *stability* statistics from Lerman et al. (2015):

$$\mathcal{P}_{\mathsf{REAP}} = \inf_{\boldsymbol{u} \in L_\star \cap S^{D-1}} \frac{1}{N} \sum_{\boldsymbol{x} \in \mathcal{X}_{\text{in}}} |\boldsymbol{u}^\top \boldsymbol{P}_\star \boldsymbol{x}|, \tag{4.3}$$

$$\mathcal{A}_{\mathsf{REAP}} = \frac{1}{N} \|\boldsymbol{X}_{\text{out}}\| \|\widetilde{\boldsymbol{Q}_\star \boldsymbol{X}_{\text{out}}}\|, \tag{4.4}$$

$$\mathcal{S}_{\mathsf{REAP}} = \frac{\mathcal{P}}{4\sqrt{d}} - \mathcal{A}. \tag{4.5}$$

Here, $\boldsymbol{Q}_\star = \boldsymbol{I} - \boldsymbol{P}_\star$ and the operator $\widetilde{\cdot}$ normalizes the columns of $\boldsymbol{Q}_\star \boldsymbol{X}_{\text{out}}$ to the unit sphere. The permeance measures how well spread the inliers are on the underlying subspace, the

alignment measures how aligned the outliers are orthogonal to $L_\star$, and the stability is a tradeoff between these two terms. The result in Theorem 2.1 of Lerman et al. (2015) states that if $\mathcal{S}_{\mathsf{REAP}} > 0$, then $\|\hat{\boldsymbol{P}} - \boldsymbol{P}_\star\|_* = 0$, where $\|\cdot\|_*$ is the nuclear or Schatten 1-norm. In other words, the REAPER program exactly recovers $L_\star$ once $\mathcal{S}_{\mathsf{REAP}} > 0$.

The following Theorem states the approximation result for the REAPER algorithms of Section 4.3. In particular, it states that as N increases and the stability is bounded below, the distance between the REAPER subspace and the true subspace goes to zero at a rate of $1/N$.

Theorem 10 *Suppose that $\mathcal{S}_{\mathsf{REAP}} > 0$. If dp-MD-REAP or dp-GD-REAP are run on the REAPER problem for $T = O(\epsilon^2 N^2)$ iterations, and if $\hat{L}$ is the principal subspace of the output of one of these algorithms, then*

$$\mathbb{E}d^2(\hat{L}, L_\star) \lesssim \frac{\pi}{2\mathcal{S}_{\mathsf{REAP}}} \frac{\log(N/\delta)}{\epsilon N}. \tag{4.6}$$

On the other hand, if $\hat{L}$ is the principal subspace of the output of dp-SMD-REAP or dp-SGD-REAP,

$$\mathbb{E}d^2(\hat{L}, L_\star) \lesssim \frac{\pi}{2\mathcal{S}_{\mathsf{REAP}}} \frac{\sqrt{1 + c_2 B^2 \log^2(1/\delta)}}{\epsilon N}. \tag{4.7}$$

The GD, MD, SGD, and SMD algorithms differ in their respective constants.

4.4. Comparing Nonconvex and Convex Results

Notice that Theorem 10 and Corollary 8 use different distance metrics for $G(D, r)$. It turns out that up to a factor of r, these are equivalent: for an r-dimensional subspace, $d^2(L_1, L_2) \leq r\theta_1^2$, where θ_1 is the maximum principal angle between L_1 and L_2. On the other hand, $1 - \sigma_r(\boldsymbol{V}_1^\top \boldsymbol{V}_2) = 1 - \cos(\theta_1) = O(\theta_1^2)$ for small θ_1. Up to a constant factor of r, the metrics $d_r(\cdot,\cdot)$ and $d(\cdot,\cdot)$ are of the same order. Thus, comparing the results of Theorem 10 to Corollary 8, we see that the nonconvex methods have a distinct advantage in the private setting. That is, the convex algorithm only achieves an approximation error of $O(N^{-1})$ while the nonconvex methods achieve approximation errors that are $O(2^{-N^\tau})$

4.5. Stability and Privacy

We finish with a short discussion of the interaction between robustness and privacy. Consider the stability result of the GGD algorithm, which states that if $\mathcal{S}_\rho(\mathcal{X}) > 0$, then GGD locally recovers the underlying subspace $L_\star$. Notice that the robustness of the method itself can yield privacy. This is stated in the following theorem.

Theorem 11 *Let $\mathcal{X}_{-i}$ be the dataset $\mathcal{X}$ with the ith datapoint removed. Suppose that*

$$\mathcal{S}_\rho(\mathcal{X}_{-i}) > 0,\ \mathcal{S}_\rho^{\mathsf{PCA}}(\mathcal{X}_{-i}) > 0,\ i = 1, \ldots, N. \tag{4.8}$$

Then, GGD with PCA initialization is differentially private.

While the condition of this theorem is hard to verify, it says that for certain inlier-outlier datasets, one doesn't even need to add noise to the GGD algorithm, since *it is already private*. An in depth study of privacy is left to future work, as the focus of this work is on the convergence of stochastic GGD methods.

5. Differential Privacy Experiments

We performed experiments in order to demonstrate some of the predictions of the substantial theory that was developed. The settings of our experiments focus on differential privacy, but we emphasize that the results are more general and similar experiments show the benefit of NGGD, SGGD, and NSGGD in practice. Additional experiments are in the appendix.

5.1. Synthetic Experiments

We present two synthetic experiments in this section. The first tests the convergence properties of the proposed algorithms for a setting with fixed parameters. The second tests the methods over a range of sample sizes and dimensions to look at their effect on subspace recovery. More comprehensive experiments that demonstrate dependencies on other parameters are in the supplemental material. All experiments were implemented on a PC with Intel i7-9700 CPUs and 16GB RAM. Below, error refers to the distance between an iterate and the underlying subspace, $d^2(\mathrm{Sp}(\boldsymbol{V}_k), \mathrm{Sp}(\boldsymbol{V}_\star))$.

We tested the 6 proposed algorithms: dp-(S)GD-REAP, dp-(S)MD-REAP and dp-(S)GGD. We set the step size for the 4 dp-REAP algorithms to be $\eta_k = 8/\sqrt{k}$. The step size for dp-GGD and dp-SGGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. We use a fixed batch size $B = \max\left(N\sqrt{\frac{\epsilon}{4T}}, 1\right)$ for both the convex and nonconvex methods (Bassily et al., 2019).

For both experiments, we randomly generate datasets according to the haystack model, with Gaussian inliers and outliers, described in Lerman et al. (2015). Points are scaled to the sphere before running our methods.

In Figure 1, we plot the median and interquartile range of log-error versus iteration for the six algorithms on 100 randomly generated sets. The fixed model parameters are $r = 2$, $D = 20$, $N = 2000$ and an inlier ratio 0.5. We set the privacy parameters to be $\epsilon = 0.8$ and $\delta = 1/\sqrt{N}$. All algorithms are run with $T = N$ iterations. We note that dp-(S)GGD converges faster to the underlying subspace than dp-(S)GD-REAP and dp-MD-(S)REAP, since its convergence rate is linear, unlike the sublinear rate for the convex methods. Nevertheless, in the initial 600 iterations of dp-SGD-REAP and dp-SMD-REAP, they converge at a faster rate than dp-SGGD (we also observe this with the initial 100 iterations of the non-stochastic methods). If D is not large, it may be beneficial to initialize dp-(S)GGD with a corresponding dp-REAP method instead of dp-PCA.

For the second experiment, Figure 2 gives a phase transition plot of N vs. D. The data parameters are $r = 2$, percentage of inliers is 0.5, and the total number of iterations of each algorithm is $T = 2N$, and we set the privacy parameters to be $\epsilon = 0.8$ and $\delta = 1/\sqrt{N}$. The step size for the dp-(S)GD-REAP algorithms is $\eta_k = 8/\sqrt{k}$, and the step size for dp-(S)GGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. Each algorithm is run 50 times and we display the average of the log-errors of the final iterate. In the non-stochastic case, the dp-GGD method outperforms the dp-GD-REAP method. Furthermore, the stochastic versions take a smaller noise, and so the methods are able to better approximate the underlying subspace for much larger Ds. Finally, as predicted by the theory, the approximations for dp-(S)GGD are much more accurate than those for dp-(S)GD-REAP.

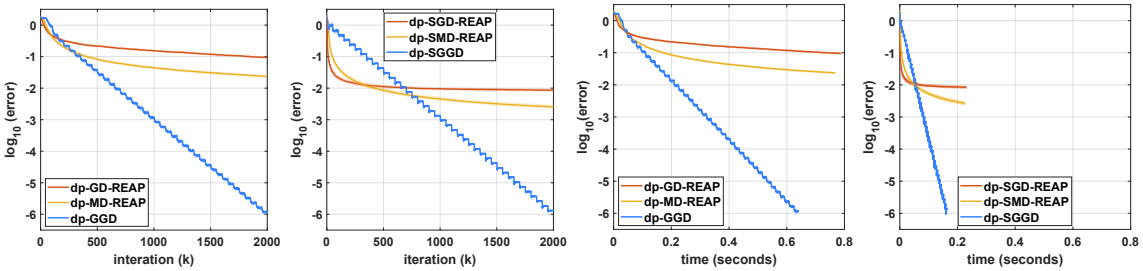

Figure 1: Convergence of the proposed algorithms with fixed parameters (see main text). Each algorithm is repeated 100 times and the median log error with a shaded region of interquartile range is plotted as a function of iterations (left two figures) and time (right two figures).

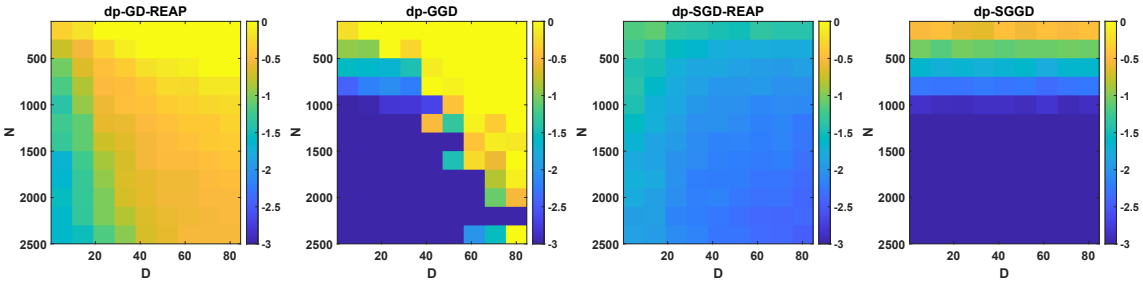

Figure 2: Phase transition plot for the number of points, N, versus the dimension, D. Each square represents the average log-distance between the final iterate and the true subspace for each algorithm. The nonconvex methods outperform the convex methods, and the stochastic method is able to perform well for much larger D due to the smaller noise required with stochastic gradients.

5.2. Stylized Application: Modified POPRES

To test on higher-dimensional data with some real characteristics, we create a stylized dataset. It aims to imitate the Population Reference Sample (POPRES) database extracted by Novembre et al. (2008). This highly private database includes 3,192 European individuals with 500,568 alleles at SNP loci. Novembre et al. (2008) filtered SNPs and screened individuals to reduce the dataset to a sample of $N = 1,387$ individuals and $D = 197,146$ SNPs. They applied PCA with $r = 2$ to the reduced data and demonstrated that the genetic information of the selected sample correlates with a geographical map of Europe.

In view of our experience with the POPRES database, we find several issues with directly using the procedure of Novembre et al. (2008) when addressing the machine learning community. First, POPRES is not publicly available. Second, the suggested preprocessing of Novembre et al. (2008) raises some questions about the meaningful selection of reduced coordinates and individuals for which a desired correlation with a given map can be demonstrated. Furthermore, the reporting on the selection of individuals (supp. material of Novembre et al. (2008)) seems to reveal some private information.

In order to avoid these sensitive issues, we generated a stylized application motivated by the work of Novembre et al. (2008). We used the publicly available dataset provided on

Github by the authors of Novembre et al. (2008). It was obtained by applying (non-private) PCA with $r = 20$ to their reduced data, so the provided data matrix $\boldsymbol{X}$ has size 1387×20. To simulate high-dimensional SNP data and further privatize $\boldsymbol{X}$, we transform it as follows: We chose $D = 10,000$ and multiplied $\boldsymbol{X}$ by a random $20 \times 10,000$ Gaussian orthogonal ensemble (GOE) matrix to get $\boldsymbol{X}'$. For outliers, we generated a $1,000 \times 30$ random matrix of uniform i.i.d. elements in $[-0.5, 0.5]$ and multiplied this matrix by a random $30 \times 10,000$ GOE matrix. We thresholded the inlier and outlier matrices to obtain the three values -1, 0 and 1 to express alleles, which we then recode as 0, 1, 2 (see details in supplemental material). We concatenated the two matrices to form an inlier-outlier $2,387 \times 10,000$ matrix $\boldsymbol{Y}$ with elements in $\{0, 1, 2\}$.

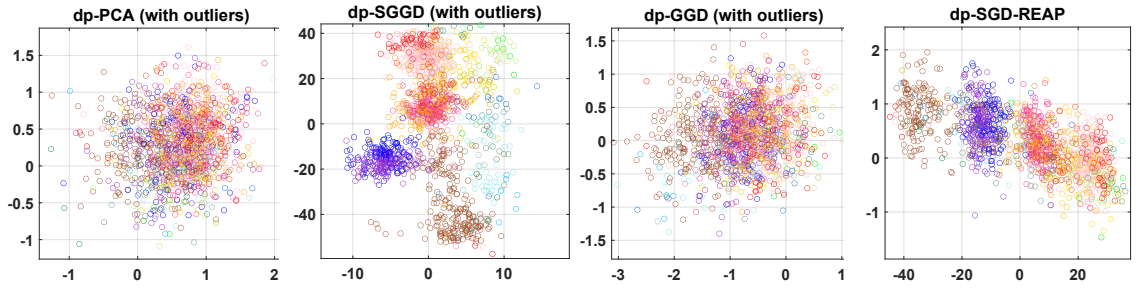

Figure 3: Recovered projections for the stylized POPRES dataset. Each algorithm is run on a synthetically generated gene matrix which mimics the original SNP data with $N = 2387$ and $D = 10000$. Out of these points, 1387 lie close to the underlying subspace, which recovers the shape of Europe, and 1000 outliers are generated to lie in the ambient space. We note that dp-SGGD is able to recover the correct shape of Europe, unlike other algorithms.

Figure 3 demonstrates the application of dp-PCA (Chaudhuri et al., 2013), dp-SGGD, dp-GGD and dp-SGD-REAP to $\boldsymbol{Y}$, and then plotting the projection of only the inliers (which are also in $\boldsymbol{X}'$). We note that dp-PCA, dp-GGD and dp-SGD-REAP are unable to recover the target 2-dimensional subspace which indicates the shape of Europe, whereas the embedding of dp-SGGD is relatively successful in doing this. Additional figures and runtimes are included in the supplemental material.

6. Conclusion

In this work, we initiate the first study of differentially private ORPCA algorithms. Our results demonstrate the distinct advantages of taking a nonconvex geometric approach to solving the ORPCA problem privately. In particular, we show that the nonconvex dp-(S)GGD algorithm converges linearly to the underlying subspace under a standard assumption of stability, in contrast to the convex method that only converges sublinearly. The techniques we use to guarantee the nonconvex dp-(S)GGD are interesting in their own right because they are the first proofs of convergence for stochastic methods in nonconvex formulation of least absolute deviations for ORPCA. Furthermore, our experiments confirm our results and demonstrate the advantages of dp-GGD and dp-SGGD. In fact, dp-SGGD seems to be superior to dp-GGD due to its ability to use smaller noise in the Gaussian mechanism.

It would be interesting to further extend the NGGD results to studying the mixing of Langevin dynamics.

There are directions to explore in future work. The following limitations are of main interest to theoreticians. We only focus here on large N, where some constants depend on D. It would be interesting to study the high-dimensional regime; however, even current works on dp-PCA do not seem to apply to this regime. There are also limitations due to the theoretical setting of ORPCA. First, we consider here the common setting of inliers lying exactly on the subspace and it would also be interesting to consider the interplay of noisy inliers and differentially private subspace recovery. Second, we assume centered data and search for linear subspaces and it will be good to justify differentially private centering approaches or extend this work searching for affine subspaces. Finally, robustness can enhance privacy (Dwork and Lei, 2009), but we did not explore this in the main text.

In addition, more practical limitations are as follows. First, we lack experiments on real data, though we explained the difficulty of working with and reporting results of the POPRES data. Second, while we theoretically verify privacy, we do not yet have a good test to verify that the algorithms are in fact private. Third, we do not know if our bounds are optimal, and it would be good to tighten these results as well as prove lower bounds. Fourth, the result for the nonconvex case is only local, and so it is unclear how the methods perform in general settings. Fifth, the choice of parameters is not sufficiently clear in the nonconvex case, and even in the convex case the estimates are only approximate. At last, we require the strong assumption of an inlier-outlier model, and it is not clear in general when data may meet this assumption.

References

Martín Abadi, H. Brendan McMahan, Andy Chu, Ilya Mironov, Li Zhang, Ian Goodfellow, and Kunal Talwar. Deep learning with differential privacy. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2016. ISBN 9781450341394. doi: 10.1145/2976749.2978318.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

R. Bassily, V. Feldman, K. Talwar, and A. Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in Neural Information Processing Systems*, 32:11282–11291, 2019.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14, 2013.

Frank H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM J. on Numerical Analysis*, 7:1–46, 1970.

C. Ding, D. Zhou, X. He, and H. Zha. R_1-PCA: rotational invariant L_1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 281–288. ACM, 2006.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2013. ISSN 15513068. doi: 10.1561/0400000042.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353 (electronic), 1999. ISSN 0895-4798.

Ky Fan and Alan J Hoffman. Some metric inequalities in the space of matrices. *Proceedings of the American Mathematical Society*, 6(1):111–116, 1955.

Evan S Gawlik and Melvin Leok. High-order retractions on matrix manifolds using projected polynomials. *SIAM Journal on Matrix Analysis and Applications*, 39(2):801–828, 2018.

J. Goes, T. Zhang, R. Arora, and G. Lerman. Robust stochastic principal component analysis. In *Artificial Intelligence and Statistics*, pages 266–274, 2014.

J. C. Gower and G. B. Dijksterhuis. *Procrustes problems*, volume 30. Oxford University Press on Demand, 2004.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27:2861–2869, 2014.

Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Wuxuan Jiang, Cong Xie, and Zhihua Zhang. Wishart mechanism for differentially private principal components analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 1730–1736, 2016.

G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2018a.

G. Lerman and T. Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, Aug 2018b. ISSN 0018-9219. doi: 10.1109/JPROC.2018.2853141.

G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

T. Maunu, T. Zhang, and G. Lerman. A well-tempered landscape for non-convex robust subspace recovery. *JMLR*, 2019.

Tyler Maunu and Gilad Lerman. Robust subspace recovery with adversarial outliers. *arXiv preprint arXiv:1904.03275*, 2019.

J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, A. Auton, A. Indap, K. S. King, S. Bergmann, M. R. Nelson, M. Stephens, and C. D. Bustamante. Genes mirror geography within europe. *Nature*, 456(7218):98–101, November 2008. doi: 10.1038/nature07331.

M. R. Osborne and G. A. Watson. An analysis of the total approximation problem in separable norms, and an algorithm for the total l_1 problem. *SIAM Journal on Scientific and Statistical Computing*, 6(2):410–424, 1985.

H. Späth and G. A. Watson. On orthogonal linear approximation. *Numer. Math.*, 51: 531–543, October 1987.

Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed sensing*, pages 210–268. Cambridge Univ. Press, Cambridge, 2012.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: faster and more general. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2719–2728, 2017.

Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535. PMLR, 2019.

Yi Yu, Tengyao Wang, and Richard J Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR, 2016.

T. Zhang and G. Lerman. A novel M-estimator for robust PCA. *Journal of Machine Learning Research*, 15(1):749–808, 2014.

Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen P Boyd, and Peter W Glynn. On the convergence of mirror descent beyond stochastic convex programming. *SIAM Journal on Optimization*, 30(1):687–716, 2020.

Supplementary Material

Appendix A. REAPER Algorithms

In Algorithm 1, we give the dp-SGD-REAPER Algorithm. The full dp-GD-REAP algorithm follows the same steps, but takes the full dataset as a batch at each iteration, uses the noise variance $\sigma^2 \geq 32T\log^2(T/\delta)/(\varepsilon^2 N^2)$ (Talwar et al., 2014). The projection step into $\mathcal{H}$ that is in line 6 of Algorithm 1 can be implemented as the water-filling procedure of Lerman et al. (2015).

We can also use mirror descent to minimize the REAPER objective. This results in the dp-MD-REAP algorithm, which we write in Algorithm 2. As in the previous case, the full MD algorithm takes the full dataset as a batch at each iteration and uses the noise variance $\sigma^2 \geq 32T\log^2(T/\delta)/(\varepsilon^2 N^2)$ (Talwar et al., 2014). The mirror map is the von Neumann entropy, $\Psi(\cdot) = \frac{1}{4}(\mathrm{Tr}(\cdot\log(\cdot)) + \log(D))$, and this approach was used before in Goes et al. (2014). It turns out that the Bregman projection for this choice of mirror map just corresponds to trace renormalization.

Algorithm 1: dp-SGD-REAP

Input: Dataset $\mathcal{X}$; subspace dimension r; step sizes η_k; max number of step T; privacy parameters: (ϵ, δ); regularization parameter:λ; batch size B; noise variance $\sigma^2 = c_2\frac{(B/N)^2T\log(1/\delta)}{\varepsilon^2 N^2}$.

$\boldsymbol{P}_0 \leftarrow \boldsymbol{A}^T\boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ and $\boldsymbol{A}_{ij} \sim N(1, 0.01)$;

for $k = 1 : T$ **do**

Sample a batch $\mathcal{B}_k \subset \mathcal{X}$ of size B uniformly with replacement;

Sample $\boldsymbol{B}_k \in \mathbb{R}^{D\times D}$ such that $\boldsymbol{B}_k = \boldsymbol{B}_k^\top$ and $\boldsymbol{B}_{k,ij} \sim N(0, \sigma^2)$;

$\tilde{\nabla}F(\boldsymbol{P}_k; \mathcal{B}_k) = \nabla F(\boldsymbol{P}_k; \mathcal{B}_k) + \boldsymbol{B}_k$ (using (4.1));

Update: $\boldsymbol{P}_{k+1} := \mathrm{argmin}_{\boldsymbol{P}\in\mathcal{H}} \left\| \boldsymbol{P} - \left[\left(\boldsymbol{P}_k - \eta_k \left(\tilde{\nabla}F(\boldsymbol{P}_k; \mathcal{B}_k) \right) \right) \right] \right\|$;

end

Output: $\tilde{\boldsymbol{P}} \leftarrow \frac{1}{T}\sum_{k=1}^{T} \boldsymbol{P}_k$

Appendix B. Supplemental Theory

B.1. Projection and Geodesic Gradient Descent

While the methods in NGGD and (NSGGD) are projected gradient methods, due to Gawlik and Leok (2018), these iteration very well approximate geodesics:

$$\mathcal{P}_{O(D,r)}(\boldsymbol{V}_k - \eta_k(\nabla F(\boldsymbol{V}_k; \mathcal{X}) + \boldsymbol{B}_k)) = \mathsf{Exp}_{\boldsymbol{V}_k}(-\eta_k(\nabla F(\boldsymbol{V}_k; \mathcal{X}) + \boldsymbol{B}_k)) + O(\eta_k^3). \tag{B.1}$$

B.2. Doob's Maximial Inequality

For the convergence of the dp-GGD, SGGD, and dp-SGGD algorithms, we use the following version of *Doob's maximal inequality*.

Algorithm 2: dp-SMD-REAP

Input: Dataset $\mathcal{X}$; subspace dimension r; step sizes η_k; max number of step T; privacy parameters: (ϵ, δ); regularization parameter:λ; batch size B; noise variance $\sigma^2 = c_2 \frac{(B/N)^2 T \log(1/\delta)}{\epsilon^2 N^2}$.

$\boldsymbol{P}_0 \leftarrow \boldsymbol{A}^T \boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ and $\boldsymbol{A}_{ij} \sim N(1, 0.01)$;

for $k = 1 : T$ **do**

Sample a batch $\mathcal{B}_k \subset \mathcal{X}$ of size B uniformly with replacement;

Sample $\boldsymbol{B}_k \in \mathbb{R}^{D \times D}$ such that $\boldsymbol{B}_k = \boldsymbol{B}_k^\top$ and $\boldsymbol{B}_{k,ij} \sim N(0, \sigma^2)$;

$\tilde{\nabla} F(\boldsymbol{P}_k; \mathcal{B}_k) = \nabla F(\boldsymbol{P}_k; \mathcal{B}_k) + \boldsymbol{B}_k$ (using (4.1));

$\boldsymbol{P}_{k+1} = \exp\left[\log(\boldsymbol{P}_k) - \eta_k \tilde{\nabla} F(\boldsymbol{P}_k; \mathcal{B}_k)\right]$;

$\boldsymbol{P}_{k+1} = r\boldsymbol{P}_{k+1} / \operatorname{Tr}(\boldsymbol{P}_{k+1})$;

end

Output: $\tilde{\boldsymbol{P}} \leftarrow \frac{1}{T} \sum_{k=1}^{T} \boldsymbol{P}_k$

Theorem 12 (Doob's maximal inequality) *Suppose that $S_t = \sum_{i=1}^{t} X_i$ is a martingale with respect to the filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, P)$, then*

$$P\left(\max_{t=1,\ldots,T} |S_k| \geq x\right) \leq \frac{\mathbb{E}|S_T|^2}{x^2}. \tag{B.2}$$

In the following proofs, the corresponding filtered probability space should be apparent from context.

B.3. Differential Privacy of dp-GGD

Proof [Proof of Theorem 6] The result for dp-GGD is just a rehash of the proof of Wang et al. (2017, Theorem 6.1) using the strong composition theorem. The result for dp-SGGD is proven in Abadi et al. (2016) using the moment accounting method. ■

B.4. Convergence of Stochastic Mirror Descent

Here we prove Theorem 9.

Proof

By a classic result that can be found, for example, in Bubeck et al. (2015), if the stochastic oracle is such that $\mathbb{E}\|(\nabla G(\boldsymbol{P}; \mathcal{B})) + \boldsymbol{B}\|^2 \leq R^2$,

$$\mathbb{E}G(\bar{\boldsymbol{P}}) - \min_{\boldsymbol{P}} G(\boldsymbol{P}) \leq R\mathsf{D}\sqrt{\frac{2}{T}}.$$

In this statement, the randomness is taken over the randomness of the minibatches $\mathcal{B}$, as well as the randomness in the Gaussian noise $\boldsymbol{B}$. We have

$$\mathbb{E}\|(\nabla G(\boldsymbol{P}; \mathcal{B})) + \boldsymbol{B}\|^2 \leq \mathbb{E}\|(\nabla G(\boldsymbol{P}; \mathcal{B}))\|^2 + \mathbb{E}\|\boldsymbol{B}\|^2$$

For the first, we have that $\mathbb{E}\|(\nabla G(\boldsymbol{P};\mathcal{B}))\|^2 \leq 1$. For the second,

$$\begin{aligned}\mathbb{E}\|\boldsymbol{B}\|^2 &= \sigma^2\mathbb{E}\|\boldsymbol{Z}\|^2 \lesssim \sigma^2 M \\ &= c_2\frac{B^2T\log^2(1/\delta)}{\varepsilon^2N^2}M.\end{aligned}$$

Plugging in,

$$\mathbb{E}G(\bar{\boldsymbol{P}}) - \min_{\boldsymbol{P}} G(\boldsymbol{P}) \leq \sqrt{1 + c_2\frac{B^2T\log^2(1/\delta)}{\varepsilon^2N^2}M}\mathsf{D}\sqrt{\frac{2}{T}}.$$

If $T = O(N^2\epsilon^2)$, then

$$\mathbb{E}G(\bar{\boldsymbol{P}}) - \min_{\boldsymbol{P}} G(\boldsymbol{P}) \leq \sqrt{1 + c_2B^2\log^2(1/\delta)}\mathsf{D}\frac{1}{\epsilon N}.$$

■

B.5. Proof of Theorem 10

Theorem 9 only guarantees an approximation to the minimizer of F after T iterations. To turn this then into a result of approximate recovery for REAPER that we see in Theorem 10, we rely on the rate of ascent for the perturbed objective by Lemma 2.3 of Lerman et al. (2015).

Lemma 13 (Lerman et al. (2015), Lemma 2.3)

$$G(\boldsymbol{P}_\star + \boldsymbol{\Delta}) - G(\boldsymbol{P}_\star) \geq \mathcal{S}\|\boldsymbol{\Delta}\|_*$$

This states that the objective grows quickly when one is far from $\boldsymbol{P}_\star$. Therefore, if the excess risk is on the order of $1/N$, then the perturbation can also be bounded on the order of $1/N$. This is stated in the following theorem.

Theorem 14 *Suppose that an algorithm to solve the REAPER problem yields a point $\hat{\boldsymbol{P}}$ such that $G(\hat{\boldsymbol{P}}) - G(\boldsymbol{P}_\star) \leq \epsilon$. Let $\hat{L}$ be the principal r-subspace of $\hat{\boldsymbol{P}}$. Then,*

$$d(\hat{L}, L_\star) \leq \frac{\pi}{2}\frac{\epsilon}{\mathcal{S}} \tag{B.3}$$

Proof This follows from combining Lemma 13 with the Davis-Kahan $\sin\Theta$ Theorem (Davis and Kahan, 1970; Yu et al., 2015). ■

The proof of Theorem 10 follows by combining Theorem 14 with Theorem 9.

B.6. dp-GGD Proofs

B.6.1. Proof of Theorem 7

Based on Jiang et al. (2016), we have

$$\begin{aligned}1-\cos(\theta_1(L_{dp-PCA},L_\star)) &= d_r^2(\boldsymbol{V}_{dp-PCA},\boldsymbol{V}_\star) \leq \|\boldsymbol{V}_{dp-PCA}\boldsymbol{V}_{dp-PCA}^\top - \boldsymbol{V}_\star\boldsymbol{V}_\star^\top\|_2\\ &\leq \|\boldsymbol{V}_{dp-PCA}\boldsymbol{V}_{dp-PCA}^\top - \boldsymbol{V}_{PCA}\boldsymbol{V}_{PCA}^\top\|_2 + \|\boldsymbol{V}_{PCA}\boldsymbol{V}_{PCA}^\top - \boldsymbol{V}_\star\boldsymbol{V}_\star^\top\|_2\\ &\leq 2\sqrt{d}\|\boldsymbol{W}\|_2 + \rho/4\\ &\leq 2\sqrt{d}O(d\log(d)/(N\epsilon)) + \rho/4,\end{aligned}$$

with high probability. For N sufficiently large, we find $d_r^2(\boldsymbol{V}_{dp-PCA},\boldsymbol{V}_\star) < \rho/2$ with high probability.

B.6.2. Proof of Theorem 2

Proof

We can write

$$\sigma_r(\boldsymbol{V}_\star^\top\boldsymbol{V}_{k+1}) \geq \sigma_r(\boldsymbol{V}_\star^\top\boldsymbol{V}_k) + \Big(\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top(-s(\mathsf{grad}F(\boldsymbol{V}_k;\mathcal{X}) + \boldsymbol{B}_k)\boldsymbol{\beta}_2\Big) - c(\mathcal{X})s^2.$$

Taking one minus both sides yields

$$d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_{k+1}) \leq d_r^2(\boldsymbol{V}_\star^\top\boldsymbol{V}_k) + s\Big(\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top((\mathsf{grad}F(\boldsymbol{V}_k;\mathcal{X}) + \boldsymbol{B}_k)\boldsymbol{\beta}_2\Big) + c(\mathcal{X})s^2.$$

Using the fact that $\sin(\arccos(x)) = \sqrt{1-x^2} \geq \sqrt{1-x}$, $x \geq 0$, stability implies that

$$s\Big(\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top((\mathsf{grad}F(\boldsymbol{V}_k;\mathcal{X}) + \boldsymbol{B}_k)\boldsymbol{\beta}_2\Big) \leq -s\mathcal{S}_\rho(\mathcal{X})d_r(\boldsymbol{V}_\star,\boldsymbol{V}_k) + s\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top\boldsymbol{B}_k\boldsymbol{\beta}_2$$

Thus

$$d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_{k+1}) \leq \Big(1 - \frac{s\mathcal{S}_\rho(\mathcal{X})}{d_r(\boldsymbol{V}_\star^\top\boldsymbol{V}_k)}\Big)d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_k) + s\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top\boldsymbol{B}_k\boldsymbol{\beta}_2 + c(\mathcal{X})s^2. \tag{B.4}$$

Let $m_T = \max_{j=1,\ldots,T} d_r(\boldsymbol{V}_\star,\boldsymbol{V}_j)$. We can iteratively apply this to yield

$$\begin{aligned}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_T) \leq (1-s\mathcal{S}_\rho(\mathcal{X})/m_T)^T d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_T) + &\sum_{j=1}^T s(1-s\mathcal{S}_\rho(\mathcal{X})/m_T)^{T-j}\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top\boldsymbol{B}_k\boldsymbol{\beta}_2 +\\ &\sum_{j=1}^T(1-s\mathcal{S}_\rho(\mathcal{X})/m_T)^{T-j}c(\mathcal{X})s^2.\end{aligned}$$

Bounding the maximum: We proceed by first bounding m_T in terms of $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$ and other quantities. To do this, first notice that this amounts to bounding $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_k)$ for $k = 1,\ldots,T$. We can use (B.4) along with $\sin(\arccos(x)) \geq 1-x$ to write

$$d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_{k+1}) \leq \Big(1 - s\mathcal{S}_\rho(\mathcal{X})\Big)d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_k) + s\boldsymbol{\beta}_1^\top\boldsymbol{V}_\star^\top\boldsymbol{B}_k\boldsymbol{\beta}_2 + c(\mathcal{X})s^2.$$

We proceed by applying Doob's maximal inequality

$$\Pr\Big(\max_{k=1,\dots,T}\Big|\sum_{j=1}^{k} s(1-s\mathcal{S}_\rho(\mathcal{X}))^{k-j}\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big|>\epsilon\Big)$$
$$\leq \Pr\Big(\max_{k=1,\dots,T}\Big|\sum_{j=1}^{k} s\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big|>\epsilon\Big)\leq\frac{s^2T\sigma^2(\sqrt{D}+\sqrt{d})^2}{\epsilon^2}.$$

Choosing $\epsilon=\frac{s\sqrt{T}\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}$ yields

$$\Pr\Big(\max_{k=1,\dots,T}\Big|\sum_{j=1}^{k} s(1-s\mathcal{S}_\rho(\mathcal{X}))^{k-j}\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big|>\epsilon\Big)\leq\lambda.$$

With probability at least $1-\lambda$, we thus have that for all $k=1,\dots,T$,

$$d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_k)\leq\Big(1-s\mathcal{S}_\rho(\mathcal{X})\Big)^k d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)+\frac{s\sqrt{k}\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}+\frac{c(\mathcal{X})sk}{\mathcal{S}_\rho(\mathcal{X})}.$$

Thus, if

$$\frac{s\sqrt{T}\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}<\frac{d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}{2}$$

and

$$\frac{s^2Tc(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}<\frac{d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}{2},$$

then $m_T<2d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$. In particular, for $s=c_1aT^{-\nu}$, these are satisfied if

$$T>\max\Big(\Big[\frac{2c_1a^2c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{1/(2\nu-1)},\Big[\frac{2a\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big).$$

Since $a>a^2$, a sufficient condition is

$$T>\max\Big(\Big[\frac{2c_1ac(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{1/(2\nu-1)},\Big[\frac{2a\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big).\tag{B.5}$$

Notice that this is a function of $a/d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$ and λ, although it decreases in a. Later on in the proof of convergence, we will see that other conditions on T imply an inverse scaling with respect to a.

Choosing $s=\frac{Ca}{T^\nu}\leq\frac{Cd_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}{2\sqrt{T}}$ then yields that $m_T<2d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$. In particular, as long as we initialize in $B_{d_r^2}(\boldsymbol{V}_\star,\rho/2)$, we see that stability holds throughout all iterations with probability at least $1-\lambda$.

Bounding the Tth iterate: Let $m_0=2d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$. From here the proof is straightforward: the first term geometrically decreases. The second can be bounded with Doob's maximal inequality with high probability and uses the fact that

$$\sum_j(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{2(T-j)}\leq\frac{1}{1-(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^2}=\frac{m_0^2}{2s\mathcal{S}_\rho(\mathcal{X})m_0-(s\mathcal{S}_\rho(\mathcal{X}))^2},$$

which is independent of T. More specifically, Doob's maximal inequality yields

$$\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^{k} s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big| > \epsilon\Big) \le \frac{\mathbb{E}\Big(\sum_{j=1}^{T} s^2(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{2(T-j)}\Big[\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big]^2\Big)}{\epsilon^2}.$$

We can upper bound

$$\begin{aligned}\mathbb{E}(\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j)^2 = \mathbb{E}(\boldsymbol{\alpha}_j^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j)^2 &\le \sigma^2\mathbb{E}\sigma_1(\boldsymbol{Z}_j)^2\\ &\lesssim \sigma^2(\sqrt{D}+\sqrt{d})^2.\end{aligned}$$

This implies that

$$\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^{k} s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big| \le C\frac{\sigma(\sqrt{D}+\sqrt{d})\sqrt{\sum_{j=1}^{T} s^2(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{2(T-j)}}}{\sqrt{\lambda}}\Big) \le \lambda,$$

or

$$\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^{k} s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}\boldsymbol{\beta}_1^{j\top}\boldsymbol{V}_\star^\top\boldsymbol{B}_j\boldsymbol{\beta}_2^j\Big| \le Cs\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{m_0^2}{2s\mathcal{S}_\rho(\mathcal{X})m_0-(s\mathcal{S}_\rho(\mathcal{X}))^2}}\Big) \le \lambda.$$

The last term uses the fact that

$$\sum_{j=1}^{T}(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j} \le \frac{m_0}{s\mathcal{S}_\rho(\mathcal{X})}$$

Putting these together, we find with probability at least $1-2\lambda$,

$$\begin{aligned}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_T) \le (1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^T d_r^2(\boldsymbol{V}_\star^\top\boldsymbol{V}_0)\\ + s\Big[C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\frac{m_0}{\sqrt{2s\mathcal{S}_\rho(\mathcal{X})m_0-(s\mathcal{S}_\rho(\mathcal{X}))^2}}+\frac{m_0c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big].\end{aligned}$$

Now, if T is sufficiently large so that $s=\frac{Ca}{T^\nu}$ satisfies

$$s < \frac{m_0}{\mathcal{S}_\rho(\mathcal{X})}, \tag{B.6}$$

$$s < \frac{a}{2m_0}\Big[C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\frac{1}{\sqrt{2s\mathcal{S}_\rho(\mathcal{X})m_0-(s\mathcal{S}_\rho(\mathcal{X}))^2}}+\frac{c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big]^{-1}. \tag{B.7}$$

The second condition can be satisfied for T greater than a constant C' with respect to a. Indeed,

$$\Big[\frac{1}{\sqrt{sm_0}}C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\frac{1}{\sqrt{2\mathcal{S}_\rho(\mathcal{X})-s(\mathcal{S}_\rho(\mathcal{X}))^2/m_0}}+\frac{c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big]$$
$$\leq \Big[\frac{1}{\sqrt{sm_0}}C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}+\frac{c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big]=O(1/\sqrt{sm_0}).$$

Thus, to satisfy the second condition, we need

$$\Big(\frac{Ca}{T^\nu}\Big)^{1/2}<\frac{a}{2\sqrt{m_0}}\Big[C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}+\frac{\sqrt{sm_0}c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big]^{-1} \tag{B.8}$$

or

$$T>\Big(\sqrt{\frac{m_0}{a}}2C^{1/2}\Big[C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}+\frac{\sqrt{sm_0}c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big]\Big)^{2/\nu} \tag{B.9}$$

Letting T be large enough so that

$$C\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}>\frac{\sqrt{sm_0}c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}, \tag{B.10}$$

since we can make $s=ca/T^\nu$ sufficiently small, we need

$$T>\Big(\frac{\sqrt{\lambda}\sqrt{\mathcal{S}_\rho(\mathcal{X})}}{C\sigma(\sqrt{D}+\sqrt{d})}\frac{\sqrt{cam_0}c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big)^{1/(2\nu)}, \tag{B.11}$$

and

$$T>\Big(\sqrt{\frac{m_0}{a}}4C^{3/2}\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}\Big)^{2/\nu}. \tag{B.12}$$

We combine (B.5), (B.11), and (B.12) to define

$$\begin{aligned}\mathcal{F}_1(a/d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0),\lambda):=\max\Big(&\Big(\frac{\sqrt{\lambda}\sqrt{\mathcal{S}_\rho(\mathcal{X})}}{C\sigma(\sqrt{D}+\sqrt{d})}\frac{\sqrt{cam_0}c(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})}\Big)^{1/(2\nu)},\\ &\Big(\sqrt{\frac{m_0}{a}}4C^{3/2}\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_\rho(\mathcal{X})}}\Big)^{2/\nu},\\ &\Big[\frac{2c_1ac(\mathcal{X})}{\mathcal{S}_\rho(\mathcal{X})d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{1/(2\nu-1)},\\ &\Big[\frac{2a\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big)\end{aligned} \tag{B.13}$$

Notice that the constraints on T given by (B.5), (B.11), and (B.13) depend only on a and $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$ the ratio $a/d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)$.

■

B.6.3. Proof of Theorem 3

Since $\mathcal{S}(\mathcal{X}^k)$ is bounded between $[-\max_i \|\boldsymbol{x}_i\|^2, \max_i \|\boldsymbol{x}_i\|^2]$, we also have the uniform bound

$$\mathbb{E}[\mathcal{S}(\mathcal{X}^k) - \mathcal{S}_{\rho,\mathbb{E}}]^2 \leq \max_i \|\boldsymbol{x}_i\|^2, \tag{B.14}$$

although in general we expect this to be much smaller. In particular, if the data is spherized, then this is bounded by 1.

Proof The proof of the theorem follows from the same reasoning as dp-GGD after splitting the sequence of errors as

$$\begin{aligned} d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_{k+1}) &\leq d_r^2(\boldsymbol{V}_\star^\top \boldsymbol{V}_k) + s\Big(\boldsymbol{\beta}_1^\top \boldsymbol{V}_\star^\top ((\mathsf{grad} F(\boldsymbol{V}_k; \mathcal{X}))\boldsymbol{\beta}_2\Big) + c(\mathcal{X})s^2 \\ &\leq \Big(1 - \frac{s\mathcal{S}_{\rho,\mathbb{E}}}{d_r(\boldsymbol{V}_\star^\top \boldsymbol{V}_k)}\Big) d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_k) + (\mathcal{S}_\rho(\mathcal{X}^k) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_k) + c(\mathcal{X})s^2 \\ &\leq \Big(1 - \frac{s\mathcal{S}_{\rho,\mathbb{E}}}{m_T}\Big) d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_k) + (\mathcal{S}_\rho(\mathcal{X}^k) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_k) + c(\mathcal{X})s^2, \end{aligned}$$

and then controlling $(\mathcal{S}_\rho(\mathcal{X}^k) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_k)$. Here, again, $m_T = \max_{j=1,\ldots,T} d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)$.

Bounding m_T: As before, be begin by bounding m_T by first looking at the looser bound

$$d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_{k+1}) \leq \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big) d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_k) + (\mathcal{S}_\rho(\mathcal{X}^k) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_k) + c(\mathcal{X})s^2.$$

Telescoping yields

$$\begin{aligned} d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_k) \leq \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^k d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0) \\ + \sum_{j=1}^k \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^{k-j} (\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j) + \sum_{j=1}^k \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^{k-j} c(\mathcal{X})s^2. \end{aligned}$$

The last term is bounded by

$$\sum_{j=1}^k \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^{k-j} c(\mathcal{X})s^2 \leq \frac{sc(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}}.$$

The other term can be bounded by Doob's maximal inequality

$$\begin{aligned} &\Pr\Big(\max_{k=1,\ldots,T} \Big| \sum_{j=1}^k \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^{k-j} (\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)\Big| > \epsilon\Big) \\ &\qquad \leq \Pr\Big(\max_{k=1,\ldots,T} \Big| \sum_{j=1}^k (\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) s\Big| > \epsilon\Big) \leq \frac{s^2 \sum_{j=1}^T \mathbb{E}(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}})}{\epsilon^2} \leq \frac{s^2 T}{\epsilon^2}. \end{aligned}$$

Setting $\epsilon = \frac{s\sqrt{T}}{\sqrt{\lambda}}$ yields

$$\Pr\Big(\max_{k=1,\ldots,T} \Big| \sum_{j=1}^k \Big(1 - s\mathcal{S}_{\rho,\mathbb{E}}\Big)^{k-j} (\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) s d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)\Big| > \frac{s\sqrt{T}}{\sqrt{\lambda}}\Big) \leq \lambda.$$

Thus, if

$$\frac{s\sqrt{T}}{\sqrt{\lambda}} < \frac{d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}{2}$$

and

$$\frac{s^2 T c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}} < \frac{d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}{2},$$

then $m_T < 2d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)$. In particular, for $s = c_1 a T^{-\nu}$, these are satisfied if

$$T > \max\Big(\Big[\frac{2c_1 a^2 c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}} d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}\Big]^{1/(2\nu-1)}, \Big[\frac{2a}{\sqrt{\lambda} d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big). \tag{B.15}$$

Bounding the Tth iterate: Let $m_0 = 2d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)$. From here the proof is straightforward: the first term geometrically decreases. The second can be bounded with Doob's maximal inequality with high probability and uses the fact that

$$\sum_j (1 - s\mathcal{S}_{\rho,\mathbb{E}}/m_0)^{2(T-j)} \le \frac{1}{1-(1-s\mathcal{S}_{\rho,\mathbb{E}}/m_0)^2} = \frac{m_0^2}{2s\mathcal{S}_{\rho,\mathbb{E}} m_0 - (s\mathcal{S}_{\rho,\mathbb{E}})^2},$$

which is independent of T. More specifically, Doob's maximal inequality yields

$$\begin{aligned}
&\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^k s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)\Big| > \epsilon\Big) \\
&\quad\le \Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^k s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}})\Big| > \epsilon\Big) \\
&\quad\le \frac{\mathbb{E}\Big(\sum_{j=1}^T s^2(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{2(T-j)}\Big[(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}})\Big]^2\Big)}{\epsilon^2}.
\end{aligned}$$

We can upper bound

$$\mathbb{E}\Big[(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}})\Big]^2 \le 1.$$

In any case, this implies that

$$\begin{aligned}
&\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^k s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)\Big| > \epsilon\Big) \\
&\qquad\qquad\qquad\qquad\le \frac{m_0^2}{2s\mathcal{S}_{\rho,\mathbb{E}} m_0 - (s\mathcal{S}_{\rho,\mathbb{E}})^2}\frac{s^2}{\epsilon^2},
\end{aligned}$$

or

$$\Pr\Big(\inf_{1\le k\le T}\Big|\sum_{j=1}^k s(1-s\mathcal{S}_\rho(\mathcal{X})/m_0)^{T-j}(\mathcal{S}_\rho(\mathcal{X}^j) - \mathcal{S}_{\rho,\mathbb{E}}) d_r(\boldsymbol{V}_\star, \boldsymbol{V}_j)\Big|$$

$$> s\frac{m_0}{\sqrt{2s\mathcal{S}_{\rho,\mathbb{E}}m_0 - (s\mathcal{S}_{\rho,\mathbb{E}})^2\sqrt{\lambda}}}\Big) \leq \lambda.$$

Putting these together, we find with probability at least $1-2\lambda$,

$$d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_T) \leq (1 - s\mathcal{S}_\rho(\mathcal{X})/m_0)^T d_r^2(\boldsymbol{V}_\star^\top \boldsymbol{V}_0) + s\Big[\frac{m_0}{\sqrt{2s\mathcal{S}_{\rho,\mathbb{E}}m_0 - (s\mathcal{S}_{\rho,\mathbb{E}})^2\sqrt{\lambda}}} + \frac{m_0 c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}}\Big].$$

Now, if T is sufficiently large so that $s = \frac{Ca}{T^\nu}$ satisfies

$$s < \frac{m_0}{\mathcal{S}_\rho(\mathcal{X})}$$
$$s < \frac{a}{2m_0}\Big[\frac{1}{\sqrt{2s\mathcal{S}_{\rho,\mathbb{E}}m_0 - (s\mathcal{S}_{\rho,\mathbb{E}})^2\sqrt{\lambda}}} + \frac{c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}}\Big]^{-1}.$$

Letting T satisfy

$$T > \Big(\frac{\sqrt{cam_0}c(\mathcal{X})}{\sqrt{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{1/(2\nu)} \tag{B.16}$$

so that

$$\sqrt{\frac{1}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}} > \frac{\sqrt{sm_0}c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})} \tag{B.17}$$

In a similar way to before, we therefore define

$$\begin{aligned}\mathcal{F}_2(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0), \lambda) := \max\Big(&\Big(\sqrt{\frac{m_0}{a}}4C^{1/2}\sqrt{\frac{1}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{2/\nu}, \\ &\Big(\frac{\sqrt{cam_0}c(\mathcal{X})}{\sqrt{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{1/(2\nu)}, \\ &\Big[\frac{2c_1a^2c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}\Big]^{1/(2\nu-1)}, \\ &\Big[\frac{2a}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big)\end{aligned} \tag{B.18}$$

Then, with probability at least $1-2\lambda$, after $T > \mathcal{F}_2(a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0), \lambda)$ iterations,

$$d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_T) < a. \tag{B.19}$$

Notice again that the constraints on T given by (B.5), (B.11), and (B.13) depend only on a and $d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)$ the ratio $a/d_r^2(\boldsymbol{V}_\star, \boldsymbol{V}_0)$. ■

B.6.4. Proof of Theorem 4

Proof

Combining the results of the previous two theorems yields the result by simultaneously controlling both martingales. Notice that $\mathcal{F}_3$ will be defined similarly to $\mathcal{F}_1$ and $\mathcal{F}_2$. In

particular, it will be the maximum of $\mathcal{F}_2$ and a term with the same form as (B.11), where we replace $\mathcal{S}_\rho$ with $\mathcal{S}_{\rho,\mathbb{E}}$. In particular, it can be defined in the following way:

$$\begin{aligned}\mathcal{F}_3(a/d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0),\lambda) := \max\Big(&\Big(\frac{\sqrt{\lambda}\sqrt{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}{C\sigma(\sqrt{D}+\sqrt{d})}\frac{\sqrt{cam_0}c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}\Big)^{1/(2\nu)},\\ &\Big(\sqrt{\frac{m_0}{a}}4C^{3/2}\frac{\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}}\sqrt{\frac{1}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{2/\nu},\\ &\Big[\frac{2c_1ac(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{1/(2\nu-1)},\\ &\Big[\frac{2a\sigma(\sqrt{D}+\sqrt{d})}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{2/(2\nu-1)},\\ &\Big(\sqrt{\frac{m_0}{a}}4C^{1/2}\sqrt{\frac{1}{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{2/\nu},\\ &\Big(\frac{\sqrt{cam_0}c(\mathcal{X})}{\sqrt{\mathcal{S}_{\rho,\mathbb{E}}(\mathcal{X})}}\Big)^{1/(2\nu)},\\ &\Big[\frac{2c_1a^2c(\mathcal{X})}{\mathcal{S}_{\rho,\mathbb{E}}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{1/(2\nu-1)},\\ &\Big[\frac{2a}{\sqrt{\lambda}d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0)}\Big]^{2/(2\nu-1)}\Big).\end{aligned} \tag{B.20}$$

The two leading terms for small a depend on $(m_0/a)^{1/\nu}$.

■

B.6.5. Proof of Theorem 5

Proof Set $s = \frac{c_1 a}{T^\nu}$. For T_1 sufficiently large so that the conditions within the theorem hold. Then, with probability $1-2\lambda$ (or $1-4\lambda$ for dp-SGGD), in T_1 iterations, $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_{T_1}) < a$.

Now suppose that we restart with $s' = s/2$ and $\boldsymbol{V}_0 = \boldsymbol{V}_{T_1}$. Notice that this is equivalent to taking $a' = a/2$ and starting distance $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_0') = d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_T) < a$. In particular, this takes $T' > \mathcal{F}_\bullet(1/2)$ iterations to reach $\boldsymbol{V}_{T+T'}$ such that $d_r^2(\boldsymbol{V}_\star,\boldsymbol{V}_{T+T'}) < a/2$. Repeating this procedure for r restarts every $\mathcal{F}_\bullet(1/2)$ iterations yields the desired result.

■

Appendix C. Supplemental Experiments

C.1. Supplemental Synthetic Experiments

This section gives additional plots demonstrating the performance of the various differentially private methods we discuss.

First, we give a phase transition plot with respect to ϵ and δ. Figure 4 shows that dp-GGD's transition from small ϵ and δ is more abrupt than that of dp-GD-REAP and dp-MD-REAP. The data parameters are as follows, the inlier dimension $r = 2$, total dimension

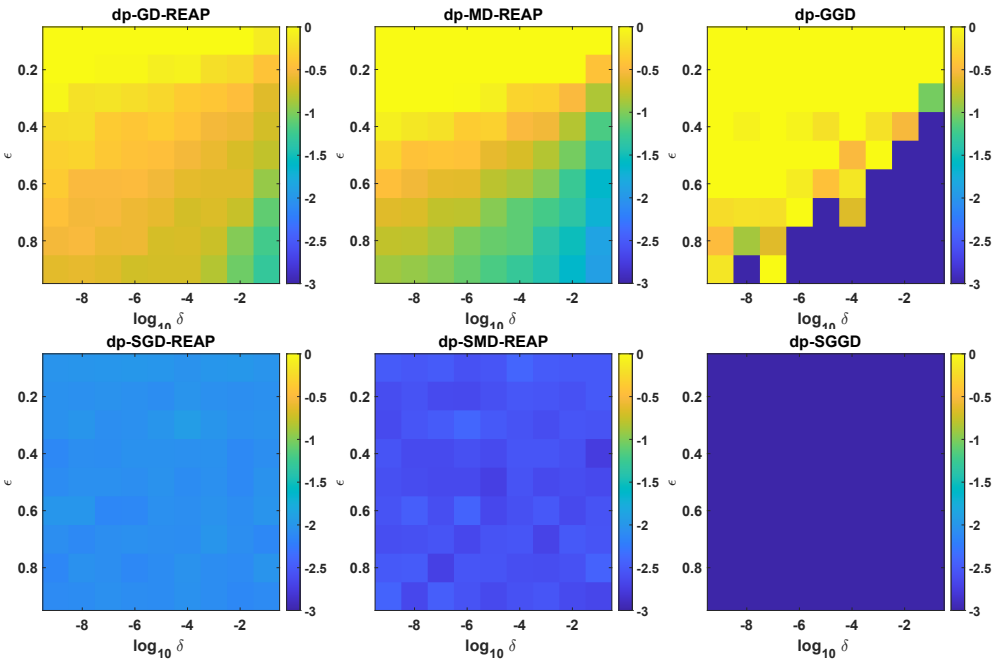

Figure 4: δ and ϵ phase transition plot. Each square represents $\log_{10}$ final error.

$D = 20$, number of points $N = 2,000$ and inlier ratio $= 0.5$. The algorithms' parameters are as follows, total number of iterations of each algorithm are the same as the number of data points N, the step size for the four dp-REAP algorithms to be $\eta_k = 8/\sqrt{k}$, the step size for dp-GGD and dp-SGGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. The experiment is repeated 50 times and the medium error is plotted.

Next, we plot the ratio of successful attempts to converge to tolerance 10^{-2} in 50 repetitions of each algorithm as a function of inlier ratio and batch size in Figure 5. (plot inlier ratio vs N) The data parameters are as follows, number of points $N = 2,000$, total dimension $D = 20$, and inlier dimension $r = 2$. The algorithms' parameters are as follows, total number of iterations of each algorithm are the same as the number of points N, the step size for the four dp-REAP algorithms to be $\eta_k = 8/\sqrt{k}$, the step size for dp-GGD and dp-SGGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. The plot shows that dp-SGD-REAP converges in the regime where inlier ratio is ≥ 0.6, dp-SMD-REAP converges in the regime where inlier ratio is ≥ 0.2, and dp-SGGD converges almost for all inlier ratios when the batch size is greater than 2.

We plot the percentage of repetitions that algorithms converge to tolerance 10^{-2} in Figure 5.

We also give a phase transition plot of D versus r, where the value is the final $\log_{10}$ error. Figure 6 shows that dp-GD-REAP, dp-MD-REAP and dp-GGD work well in the regime where both D and r are small. The data parameters are as follows, number of points $N = 2,000$ and inlier ratio $= 0.5$. The algorithms' parameters are as follows, total number of iterations of each algorithm are the same as the number of points N, the step size for the four dp-REAP algorithms to be $\eta_k = 8/\sqrt{k}$, the step size for dp-GGD and

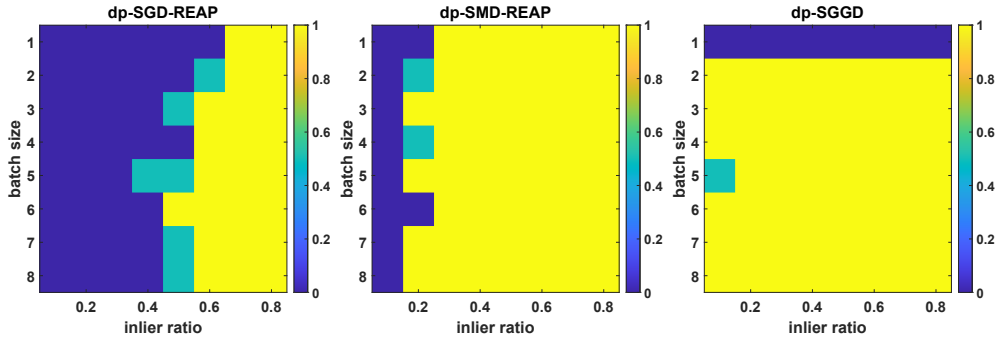

Figure 5: Inlier ratio and batch size transition plot. Each square represents the percentage of time that the algorithm achieves the given tolerance for a given combination of inlier ratio and batch size.

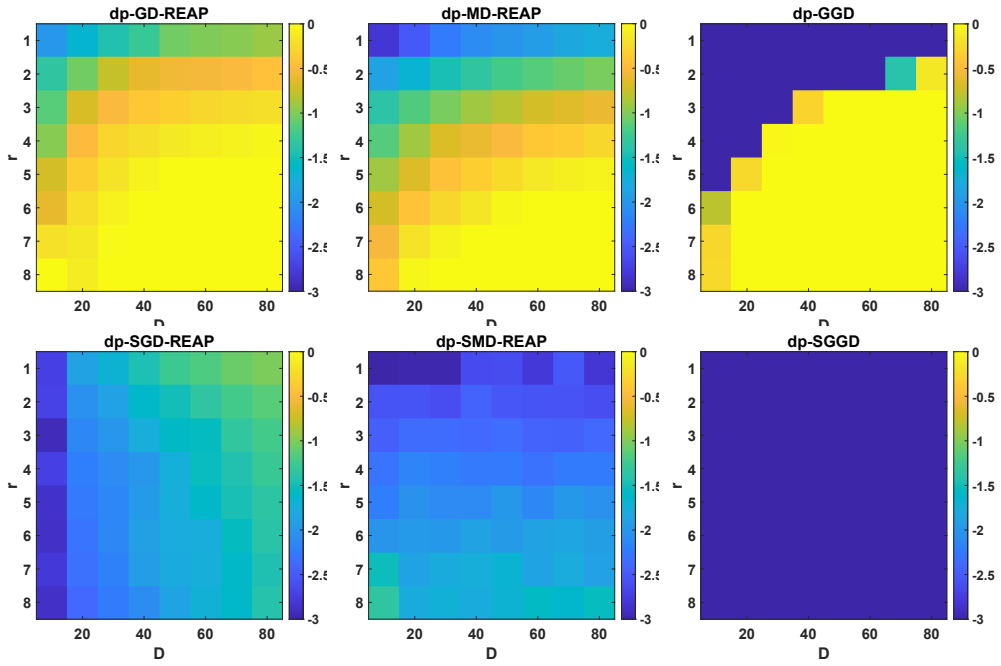

Figure 6: D and r phase transition plot. Each square represents $\log_{10}$ final error. The x-axis correspond to different values of D and the y-axis correspond to different values of r.

dp-SGGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. The experiment is repeated 50 times and the medium error of all repetitions is plotted.

Next, we give a phase transition plot of N versus D with $\log_{10}$ error in Figure 7, and time to error $= 10^{-2}$ in Figure 8. We plot $\log_{10}$ final error, and time to converge to tolerance (Figure 8), and ratio of failed attempt to reach tolerance in 50 repetitions as a function of N and D in Figure 9. In the event that none of the repetitions successfully reaches tolerance, the square shows yellow in Figure 8, this corresponds with number of failed attempts in Figure 9. Figure 7 shows that the dp-GD-REAP, dp-MD-REAP and dp-GD work well in

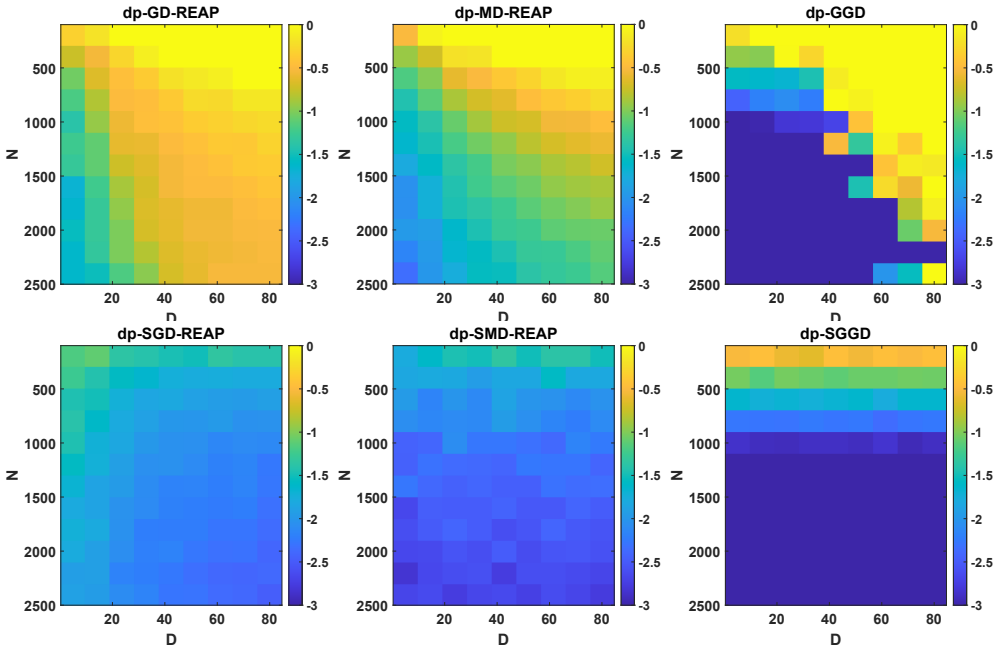

Figure 7: D and N phase transition plot. Each square represents $\log_{10}$ final error.

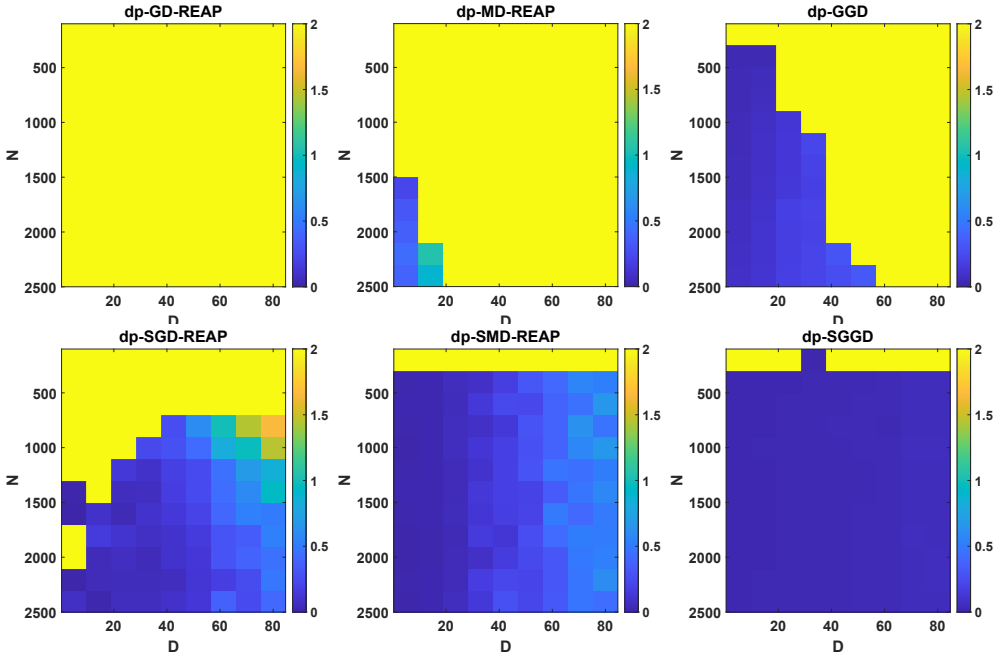

Figure 8: D and N phase transition plot. Each square represents the time to reach tolerance 10^{-2}. If in all repetitions the algorithms failed to converge, the square is shown in yellow (I imputed a large number, 2 in this case).

the regime where D is small and N is large. The data parameters are as follows, the inlier

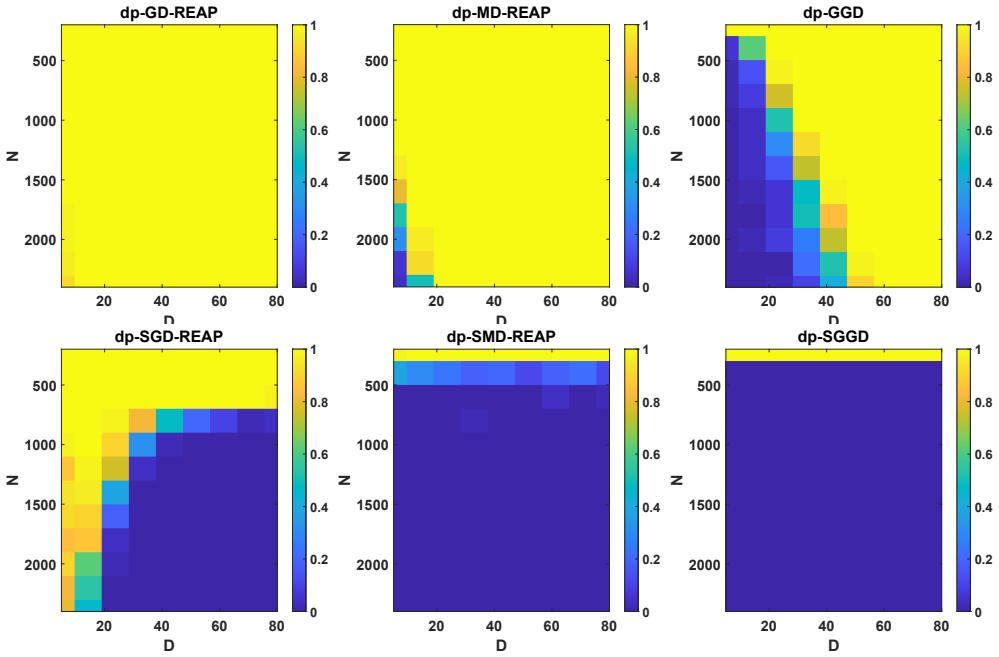

Figure 9: D and N phase transition plot. Each square represents the percentage of repetitions each algorithm fails to reach tolerance 10^{-2}.

dimension $r = 2$, and the inlier ratio $= 0.5$. The algorithms' parameters are as follows, total number of iterations of each algorithm are the same twice the number of points $2N$, the step size for the four dp-REAP algorithms to be $\eta_k = 8/\sqrt{k}$, the step size for dp-GGD and dp-SGGD is $\eta_k = 1/2^{\lfloor k/50 \rfloor}$. Each algorithm is run repetitions of 50 times.

C.2. Supplemental Plots for POPRES Data

At last we present supplemental plots of the performance of various differentially-private methods we discuss on the stylized POPRES dataset Figure 10. The top left plot shows recovered projections by applying PCA to stylized POPRES dataset without outliers ($\boldsymbol{X}$ in the main text). And the rest of plots show recovered projections by applying various differentially-private methods to stylized POPRES dataset with outliers.

We report the details of stylized POPRES experiment. The experiment for dp-(S)GGD algorithm is performed on a machine with Intel i7 processor and 16GB RAM, the running time of dp-SGGD is 124 seconds and the running time of dp-GGD is 679 seconds. The experiment for dp-REAP algorithms is performed on a cluster of 24 Intel Haswell E5-2680v3 processors and 60GB RAM. Each dp-REAP algorithm is run for 100 iterations. The running time of dp-SMD-REAP is 5,471 seconds, that of dp-MD-REAP is 5,684 seconds, that of dp-GD-REAP is 36,027 seconds and that of dp-SGD-REAP is 35,125 seconds. Note that dp-REAP methods are more computationally expensive than dp-(S)GGD. Beside what is noted in the main text, dp-SGD-REAP and dp-SMD-REAP are not able to recover the subspace exactly but can roughly discern clusters of individuals with the same ancestry.

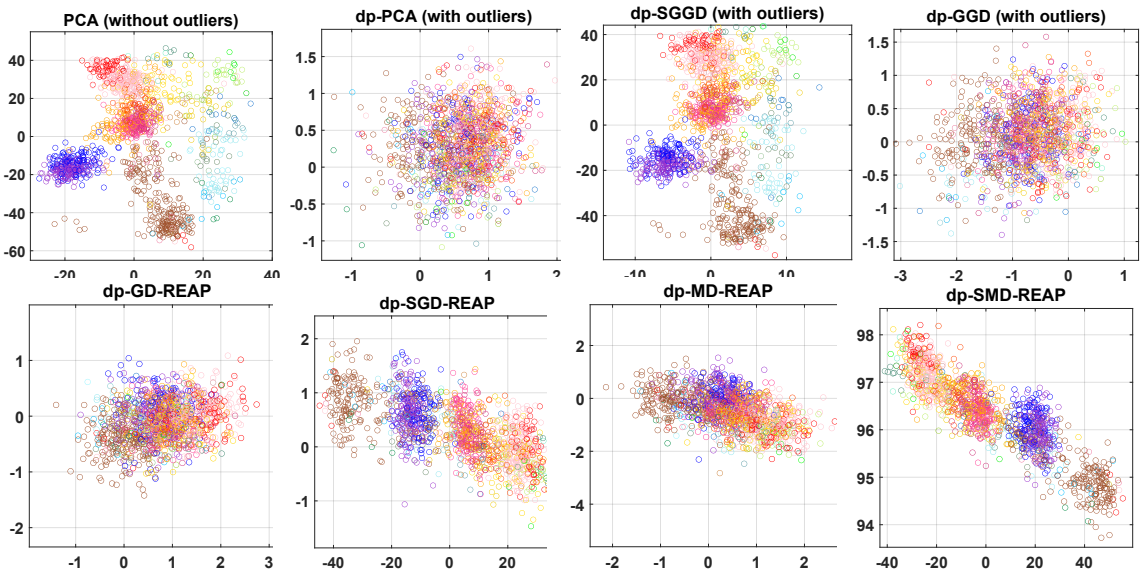

Figure 10: Recovered projections for the stylized POPRES dataset. Each algorithm is run on a synthetically generated gene matrix which mimics the original SNP data with $N = 2387$ and $D = 10000$. Out of these points, 1387 lie close to the underlying subspace, which recovers the shape of Europe, and 1000 outliers are generated to lie in the ambient space