PRIMO: Private Regression in Multiple Outcomes

Seth Neel *sneel@hbs.edu*
Harvard Business School

Reviewed on OpenReview: *https://openreview.net/forum?id=34vtRA3Nvu*

Abstract

We introduce a new private regression setting we call *Private Regression in Multiple Outcomes* (PRIMO), inspired by the common situation where a data analyst wants to perform a set of l regressions while preserving privacy, where the features X are shared across all l regressions, and each regression $i \in [l]$ has a different vector of outcomes y_i. Naively applying existing private linear regression techniques l times leads to a $\sqrt{l}$ multiplicative increase in error over the standard linear regression setting. We apply a variety of techniques including sufficient statistics perturbation (`SSP`) and geometric projection-based methods to develop scalable algorithms that outperform this baseline across a range of parameter regimes. In particular, we obtain *no dependence on* l in the asympotic error when l is sufficiently large. We apply our algorithms to the task of private genomic risk prediction for multiple phenotypes using data from the 1000 Genomes project and the Database of Genotypes and Phenotypes (dbGaP). Empirically, we find that even for values of l far smaller than the theory would predict, our projection-based method improves the accuracy relative to the variant that doesn't use the projection.

1 Introduction

Linear regression is one of the most fundamental statistical tools used across the applied sciences, for both inference and prediction. In genetics, polygenic risk scores Krapohl et al. (2018); Pattee & Pan (2020) are computed by regressing phenotype (e.g. disease status) onto individual genomic data (SNPs) in order to identify genetic risk factors. In the social sciences, observed societal outcomes like income or marital status might be regressed on a fixed set of demographic features Agresti & Barbara (2009). In many of these cases where the data records correspond to individuals, there are two aspects of the problem setting that co-occur:

Aspect 1. The individuals may have a legal or moral right to privacy that has the potential to be compromised by their participation in a study.

Aspect 2. Multiple regressions will be ran using the same set of individual characteristics across each regression with different outcomes, either within the same study or across many different studies.

Aspect 1 has been established as a legitimate concern through both theoretical and applied work. The seminal paper of Homer et al. (2008) showed that the presence of an individual in a genomic dataset could be identified given simple summary statistics about the dataset, leading to widespread concern over the sharing of the results of genomic analyses. In the machine learning setting, where what is being released is a model w trained on the underlying data, there is a long line of research into “Membership Inference Attacks" Hu et al. (2021); Shokri et al. (2016), which given access to w are able to identify which points are in the training set. Over the last decade, differential privacy Dwork & Roth (2014) has emerged as a rigorous solution to the privacy risk posed by Aspect 1. In the particular case of linear regression the problem of how to privately compute the optimal regressor has been studied in great detail, which we summarise in Subsection 2.1.

Aspect 2 has been studied extensively from the orthogonal perspective of multiple hypothesis testing, but until now has not been considered in the context of privacy. The problem of overfitting or “p-hacking” in

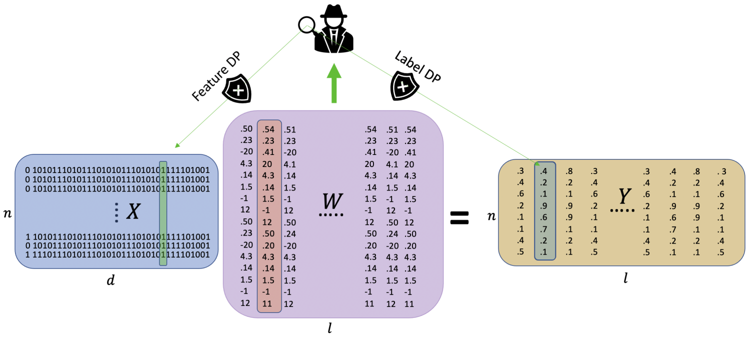


Figure 1: PRIMO imagines the scenario where the weights W computed as a function of sensitive features X and multiple outcomes Y are published or leaked. Our algorithms prevent an adversary with access to W from exposing the underlying sensitive data x_i, y^i.

the social and natural sciences has been referred to as the "statistical crisis in science" Gelman & Loken (2014), and developing methods that quantify and mitigate the effects of overfitting has been the subject of much attention in the statistics and computer science communities Dwork et al. (2015a); Bassily et al. (2021); Korthauer et al. (2019). Given the ubiquity of Aspects 1 and 2, this raises an important question that we explore in this work:

When computing $i = 1 \dots l$ distinct regressions with a common set of X's and distinct y_i's, what is the optimal accuracy-privacy tradeoff?

2 Preliminaries

We start by defining the standard linear regression problem. Let $X \in \mathcal{X}^n \subset \mathbb{R}^{n \times d}$ consist of d-dimensional samples from n individuals, $y_i \in \mathcal{Y}^n$ be a vector of n outcomes for a regression indexed by i, and parameter space $\mathcal{W} \subset \mathcal{R}^d$ denote the subset of linear regression coefficients we will optimize over. Following convention in prior work, we will let $||\mathcal{X}||_2, ||\mathcal{Y}||_2, ||\mathcal{W}||_2$ denote $\sup_{x \in \mathcal{X}} ||x||_2, \sup_{y \in \mathcal{Y}} ||y||_2, \sup_{w \in \mathcal{W}} ||w||_2$ respectively.

For $w \in \mathcal{W}$ let $f(w) := \frac{1}{n}\sum_{k=1}^{n}(w \cdot x_k - y_{ik})^2$ be the linear regression objective, and denote by $w_{i*} = \arg\min_{w \in \mathbb{R}^d} f(w) = (X^TX)^{\dagger}X^Ty_i$, where $\dagger$ is the Moore-Penrose inverse. For $\lambda > 0$ let $f_\lambda(w) = \frac{1}{n}\sum_{k=1}^{n}(w \cdot x_k - y_{ik})^2 + \lambda||w||_2^2$ the ridge regression objective, and let $w_{i*}^{\lambda} = \arg\min f_\lambda(w)_{w \in \mathbb{R}^d} = (\frac{1}{n}X^TX + \lambda I_d)^{-1}\frac{1}{n}X^Ty_i$.

The definition of data privacy we use throughout is the popular (ε, δ)-differential privacy introduced in Dwork et al. (2006). We refer the reader the to Dwork & Roth (2014) for an overview of the basic properties of $(\varepsilon, \delta)-$DP including closure under post-processing and advanced composition. In order to define a differentially private mechanism, we first define what it means for two datasets to be adjacent.

Definition 1. *Two datasets $(X, y), (X', y') \in \mathcal{X}^n \times \mathcal{Y}^n$ are adjacent if they differ in a single element, e.g. there exists $i, i' \in [n]$, such that $X \cup \{x_i'\} \setminus \{x_i\} = X', y \cup \{y_i'\} \setminus \{y_i\} = y'$. We say that $(X, y), (X', y')$ are* feature-adjacent *if they share labels $y = y'$, and $\exists i \in [n]$ such that $X \cup \{x_i'\} \setminus \{x_i\} = X'$. Similarly, they are label-adjacent if $X = X'$, and $\exists i \in [n]$ such that $y \cup \{y_i'\} \setminus \{y_i\} = y'$.*

We denote adjacency of $(X, y), (X', y')$ by $(X, y) \sim (X', y')$.

Definition 2. *Let $\mathcal{M} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{O}$ a randomized algorithm taking as input a dataset of n records. $\mathcal{M}$ is (ε, δ)-DP if $\forall$ pairs of adjacent datasets $(X, Y) \sim (X', Y'), O \subset \mathcal{O}$:*

$$\Pr[\mathcal{M}(X, Y) \in O] \leq e^{\varepsilon} \Pr[\mathcal{M}(X', Y') \in O] + \delta \tag{1}$$

When Definition 1 holds for adjacency defined over (x_i, y_i) pairs, this is the standard setting of differential privacy, which we call "Full DP" in order to differentiate it from its relaxations. In the less restrictive case where Equation 1 holds only over pairs of label-adjacent datasets, we are in the well-studied setting of *label* *differential privacy* Ghazi et al. (2021); Esfandiari et al. (2021); Papernot et al. (2016). When Equation 1 holds only over pairs of feature-adjacent datasets, we say that we have *feature differential privacy.* We develop specific PRIMO algorithms for these relaxations of DP in Section 5. We now formalize the *Private Regression In Multiple Outcomes (PRIMO)* problem.

Definition 3. *PRIMO. Let $x_i \in \mathcal{X} \subset \mathbb{R}^d, y_{ij} \in \mathcal{Y}$, for $i = 1 \dots n, j = 1 \dots l$. Let $X_{n \times d}$ the matrix with i^{th} row x_i, and let $Y_{n \times l}$ the matrix with j^{th} column $y_j = (y_{1j}, \dots y_{nj})$. The optimal solution W^* to the PRIMO problem is*

$$W^* = \underset{W \in \mathcal{W}^l \subset \mathbb{R}^{d \times l}}{\arg\min} ||XW - Y||_F^2$$

Given a randomized algorithm $\mathcal{M} : (\mathcal{X} \times \mathcal{Y}^l)^n \to \mathcal{W}^l$, we say that $\mathcal{M}$ is an $(\alpha, \beta, \varepsilon, \delta)$ solution to the PRIMO problem if (i) $\mathcal{M}$ is (ε, δ)-DP, and (ii) with probability $1 - \beta$ over $\tilde{W} \sim \mathcal{M}$:

$$\frac{1}{nl}||X\tilde{W} - Y||_F^2 - \frac{1}{nl}||XW^* - Y||_F^2 < \alpha$$

We will use y_j to denote the vector of n outcomes for the j^{th} outcome, and $y^i \in \mathcal{Y}^l$ for the vector of l outcomes corresponding to individual $i \in [n]$. We note that in PRIMO under full DP, the adjacency condition holds over pairs (x_i, y^i). The (ϵ, δ)-DP algorithm we will use most throughout is the Gaussian Mechanism. We state a version of the Gaussian mechanism with constant $c(\epsilon, \delta)$ that is valid for all $\epsilon > 0$, which follows from analyzing the mechanism using Renyi DP Mironov (2017) and converting back to (ϵ, δ)-DP.

Lemma 1 (Dwork & Roth (2014))**.** *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ an arbitrary d-dimensional function, and define it's sensitivity $\Delta_2(f) = \sup_{X \sim X'} ||f(X) - f(X')||_2$, where $X \sim X'$ are datasets that differ in exactly one element. Then the Gaussian mechanism `GaussMech`$(\varepsilon, \delta, \Delta)$ releases $f(X) + \mathcal{N}(0, \sigma^2)$, and is (ε, δ)-differentially private for $\sigma \geq c(\varepsilon, \delta)\Delta_2(f)$, where $c(\varepsilon, \delta) = \sqrt{2(\frac{1}{\epsilon} + \frac{\log \frac{1}{\delta}}{\varepsilon^2})}$.*

Lastly, we define some notation. Given a vector v and matrix A, $||v||_2, ||A||_2$ denote the l_2 and the spectral norm respectively, $||A||_F$ is the Frobenius norm, $||\mathcal{X}||_2 = \sup_{x \in \mathcal{X}} ||x||_2$, $|\mathcal{X}| = \sup_{x \in \mathcal{X}} ||x||_\infty$, $|\mathcal{Y}| = \sup_{y \in \mathcal{Y}} |y|$, $||\mathcal{W}||_2 = \sup_{w \in \mathcal{W}} ||w||_2$, and $||\hat{w}||_2 = \sqrt{\frac{1}{l}||W^*||_F^2}$, where W^* is the optimal PRIMO solution in Definition 3.

2.1 Private Linear Regression

Private linear regression is well-studied under a variety of different assumptions on the data generating distributions and parameter regimes. Typically analysis of private linear regression is done either under the fully agnostic setting where only parameter bounds $||\mathcal{X}||_2, |\mathcal{Y}|, ||\mathcal{W}||_2$ are assumed, under the assumption of a fixed design matrix and y generated by a linear Gaussian model (the so-called realizable case), or under the assumption of a random design matrix Milionis et al. (2022). In this paper we focus on the first fully agnostic setting, because in our intended applications within the social and biomedical sciences in general we neither have realizability (when y is actually a linear function of x) or Gaussian features. In the fully agnostic (not realizable) setting Wang (2018) provides a comprehensive survey of existing private regression approaches and bounds, including a discussion of the impact of different parameter norms, and proposing a new adaptive technique.

Broadly speaking, techniques for private linear regression fall into 4 classes, sufficient statistics perturbation (`SSP`) Vu & Slavkovic (2009); Foulds et al. (2016), Objective Perturbation (`ObjPert`) Kifer et al. (2012), Posterior sampling Dimitrakakis et al. (2016), and privatized (stochastic) gradient descent (`NoisySGD`) Chaudhuri et al. (2011). The methods in this paper are a sub-class of `SSP`-based methods, which correspond to Algorithm 1 where $l = 1$. Methods based on `SSP` rely on perturbing $X^T X \approx \Sigma$ and $X^T y \approx \Sigma_{Xy}$ separately, and then using these noisy estimates to compute the (regularized) least squares estimator:

$$\tilde{w}_{i*} = \underbrace{(X^T X + \lambda I + E_1)^{-1}}_{\text{noisy covariance term } \tilde{\Sigma}} \times \underbrace{(X^T Y_i + E_{2i})}_{\text{noisy association term } \tilde{\Sigma}_{Xy}} \tag{2}$$

Bassily et al. (2014) give a lower bound on the mean squared error of *any* (ϵ, δ)-DP algorithm for optimizing a Lipschitz objective function. In the case of linear regression (which corresponds to PRIMO with $l = 1$), if

$w_{private}$ denotes the output of any (ϵ, δ)-DP algorithm, then:

$$\alpha = f(w_{private}) - f(w_{i*}) \geq \min\{||\mathcal{Y}||_2^2, \frac{\sqrt{d}(||\mathcal{X}||_2^2||\mathcal{W}||_2^2 + ||\mathcal{X}||_2||\mathcal{W}||_2||\mathcal{Y}||_2}{n\epsilon}\} \tag{3}$$

Cai et al. (2020) prove a lower bound that is tailored for realizable low-dimensional ($n \geq d$) linear regression, where $y \sim x^t\beta + N(0, \sigma^2)$, with $||x||_2 \leq 1$ and $||X^TX||_\infty = O(1/d)$, showing that $\alpha = \Omega(\sigma^2(\frac{d}{n} + \frac{d^2}{n^2\epsilon^2}))$. Neither of these bounds are exactly applicable to our setting; the bound of Bassily et al. (2014) is not neccessarily tight for linear regression, and the bound of Cai et al. (2020) makes realizability assumptions that clearly fail to hold for the type of genomic applications that motivate our work. We also note that all existing lower bounds for private linear regression are under full DP, and so do not immediately apply to the algorithms in Section 5 which operate in the weaker feature-DP or label-DP settings. In general, the focus of this work is on upper bounds, and so we highlight this curious lack of relevant lower bounds in the literature as a potential future direction, given the long line of work on upper bounds for private linear regression. Acknowledging the above issues with existing bounds, throughout the paper we'll use the lower bound of Bassily et al. (2014) as a benchmark for the cost to accuracy of taking $l > 1$, and in settings in which the bounds for PRIMO match this lower bound we will say we have "PRIMO for Free."

Given any (ε, δ)-DP algorithm for computing w_j privately, we can use it as a sub-routine to solve PRIMO by simply running it l times to compute each column of W. Hence by running any of the optimal algorithms (for example `SSP` Vu & Slavkovic (2009)) l times with parameters $\varepsilon' \approx \varepsilon/\sqrt{l}, \delta' \approx \delta/l$, by advanced composition for differential privacy Dwork & Roth (2014) we can achieve MSE, subject to (ε, δ)-DP:

$$\alpha = \tilde{O}(\frac{\sqrt{ld}(||\mathcal{X}||_2^2||\mathcal{W}||_2^2 + ||\mathcal{X}||_2||\mathcal{W}||_2|\mathcal{Y}|}{n\varepsilon}), \tag{4}$$

where the $\tilde{O}$ hides factors of $\log(\frac{1}{\delta})$. So for a fixed privacy budget ε, this naive baseline is a factor of $\sqrt{l}$ worse than in the standard private regression setting where $l = 1$.

Further background on private query release for linear and low-sensitivity queries under l_∞ and l_2 error and on sub-sampled linear regression (Subsection 6.2) is included in the Appendix.

3 Results

The primary contribution of this work is to introduce the novel Private Regression in Multiple Outcomes (PRIMO) problem, and to provide a class of algorithms that trade off accuracy, privacy, and computation. In addition to introducing the PRIMO problem, to our knowledge we are the first to apply private query release methods to linear regression (Section 5). We also provide a compelling practical application of our algorithms to the problem of genomic risk prediction with multiple phenotypes using data from The 1000 Genomes Project Fairley et al. (2019) and The Database of Genotypes and Phenotypes Mailman et al. (2007).

Algorithm 1 is our meta-algorithm for the PRIMO problem, and has two variants, `ReuseCovGauss` which corresponds to $\mathcal{M} =$ `GaussMech`, and `ReuseCovProj`, which corresponds to $\mathcal{M} =$ Algorithm 2. Algorithm 2 itself has two variants, which correspond to the label private setting ($\texttt{GaussProj}_X$), and to the feature private setting ($\texttt{GaussProj}_Y$) respectively. In Section 4 we adapt the previously proposed sufficient statistics perturbation (`SSP`) algorithm of Vu & Slavkovic (2009); Foulds et al. (2016) into the `ReuseCovGauss` Algorithm. Via a novel accuracy analysis of `SSP` for the case when the privacy levels for E_1, E_{2i} differ in Equation 2 (Theorem 1), we show that since the noisy covariance matrix is reused across the regressions (as it only depends on X), by allocating the majority of our privacy budget to computing this term, we are able to obtain PRIMO for Free when l is sufficiently small, because the error term is dominated by the error in computing the noisy covariance matrix, which does not depend on l, rather than the error from computing the noisy association term (row 1 of Table 1). When $l > \frac{||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2}$, the asymptotic error of `ReuseCovGauss` is dominated by the error in computing the noisy association term, which has a $\sqrt{l}$ dependence in the error on l (row 2 of Table 1). Given this, it is natural to ask if, under parameter regimes where the error from the association term dominates, can we obtain improved dependence of α on l over the $\sqrt{l}$ given by `ReuseCovGauss`?

<table>
<tr><th colspan="5">Guarantees of <code>ReuseCov</code></th></tr>
<tr><th>$\#l$ of regressions</th><th>$\alpha = $ MSE</th><th>cost(l)</th><th>$\mathcal{M}$</th><th>Privacy</th></tr>
<tr><td>$l < \min(\frac{n}{\sqrt{d}}, \frac{||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2})$</td><td>$\tilde{O}\left(\frac{\sqrt{d}||\mathcal{X}||_2^2||\hat{w}||_2^2}{n}\right)$</td><td>1</td><td><code>Gauss</code></td><td>Full DP</td></tr>
<tr><td>$l \in (\frac{||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2}, \frac{n}{\sqrt{d}})$</td><td>$\tilde{O}\left(\frac{||\hat{w}||_2\sqrt{ld}|\mathcal{Y}|||\mathcal{X}||_2}{n}\right)$</td><td>$\sqrt{l}$</td><td><code>Gauss</code></td><td>Full DP</td></tr>
<tr><td>$l > \frac{n}{\sqrt{d}}, n < \frac{\sqrt{d}||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2}$</td><td>$\tilde{O}\left(\frac{\sqrt{d}||\mathcal{X}||_2^2||\hat{w}||_2^2}{n}\right)$</td><td>1</td><td>$\texttt{Proj}_Y$</td><td>Feature DP</td></tr>
<tr><td>$l > \frac{n}{\sqrt{d}}, n > \frac{\sqrt{d}||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2}$</td><td>$\tilde{O}\left(\frac{||\hat{w}||_2 d^{1/4}|\mathcal{Y}|||\mathcal{X}||_2}{\sqrt{n}}\right)$</td><td>$\frac{\sqrt{n}}{d^{1/4}}$</td><td>$\texttt{Proj}_Y$</td><td>Feature DP</td></tr>
<tr><td>$l > n^{\frac{2}{3}}$</td><td>$\tilde{O}\left(||\hat{w}||_2\frac{|\mathcal{X}|||Y||_2\sqrt{d}}{\sqrt{n}l^{1/4}}\right)$</td><td>$l^{1/4}\sqrt{n}$</td><td>$\texttt{Proj}_X$</td><td>Label DP</td></tr>
</table>

Table 1: In the table above cost(l) is the ratio of the PRIMO MSE α to the lower bound in Equation 3, and $\mathcal{M}$ denotes the mechanism used to compute the association term X^TY in Algorithm 1.

We answer this question in the affirmative in Section 5 with the introduction of the `ReuseCovProj` algorithms. The key idea is that instead of privately computing $\frac{1}{n}X^TY$ via the Gaussian Mechanism (Lemma 1), which necessarily has error that scales like $\sqrt{l}$, in the setting where *either* X or Y are not considered private, projecting the noisy value $\frac{1}{n}X^TY + \mathcal{N}(0, \sigma^2)$ returned by the Gaussian Mechanism onto the space of feasible values that the non-private quantity $\frac{1}{n}X^TY$ could take (Line 5 of Algorithm 2), can reduce the error for sufficiently large l. This set of "feasible" values is the image of the domain of Y under X^T when X is considered public, and it is the image of the domain of X under Y^T when Y is considered public. The assumption that either X or Y are public are critical to these algorithms, as computing the projection depends on the space we are projecting into in a way that would not be easily privatized. The setting where X is public and Y is private corresponds to feature DP, and the setting where X is public and Y is public is label differential privacy. In rows $3-5$ in Table 1 we summarise the guarantees of Theorems 3,4 that characterize the accuracy in these settings. In both the Feature DP and Label DP settings, the projection improves the dependence in the error on l, d, at the cost of a factor of $\sqrt{n}$ in the denominator. As a result, until l, d are sufficiently large relative to n we'd expect the projection to actually increase error. We discuss these bounds further in Section 5.

In Section 7, we implement our `ReuseCovGauss` and `ReuseCovProj` algorithms using SNP data from two of the most common genomic databases Mailman et al. (2007); Fairley et al. (2019). We compare our algorithms to each other, and to the non-private baseline. In both settings we reach the surprising conclusion that even for relatively small values of $l = 11, 100$, much smaller than the theory would predict, `ReuseCovProj` outperforms `ReuseCovGauss`. Moreover, `ReuseCovProj` is able to achieve non-trivial MSE (as measured by R^2) for very large values of l, as predicted by Theorem 3.

4 Full DP: The `ReuseCovGauss` Algorithm

We start by presenting our first algorithm for PRIMO, `ReuseCovGauss`, which corresponds to Algorithm 1 with $\mathcal{M} = $ `GaussMech`. To understand the intuition behind `ReuseCovGauss`, we start from the analysis of the ridge regression variant of `SSP` in Wang (2018). Equation 13 in Wang (2018) shows that if $\tilde{w_i}$ is the noisy ridge regressor output by `SSP`, and w_{i*} is the (non-private) OLS estimator, then w.p. $1 - \rho$:

$$f(\tilde{w_i}) - f(w_{i*}) = \tilde{O}\left(\underbrace{\frac{d}{\lambda\epsilon^2}||\mathcal{X}||_2^2|\mathcal{Y}|^2}_{\text{association error term}} + \underbrace{\frac{d}{\lambda\epsilon^2}||\mathcal{X}||_2^4||\mathcal{W}||_2^2}_{\text{covariance error term}}\right) + \underbrace{\lambda||\mathcal{W}||_2^2}_{\text{error due to ridge penalty}} \tag{5}$$

Inspecting these terms, we see that when $||\mathcal{X}||_2||\mathcal{W}||_2 \gg ||\mathcal{Y}||_2$ the error is dominated by the covariance error term, which is the error due to random noise injected when privately computing X^TX, rather than the association error term, which is error due to random noise injected when privately computing $X^T y_i$. We now turn back to our PRIMO setting, and imagine independently applying `SSP` to solve each of our $i = 1 \dots l$ regression problems. By the lower bounds in Bassily et al. (2014), given a fixed privacy budget this incurs at least a $\sqrt{l}$ multiplicative blow up in error. However, we notice that when our private linear regression subroutine is an `SSP` variant, this naive scheme of running l independent copies of `SSP` is grossly wasteful. Since the X matrix is shared across all the regressions, we can simply compute our noisy estimate of $X^T X$ once, and then reuse it across all the regressions. Combining these two observations, we present Algorithm 1. $\mathcal{M}$ can be any $(\epsilon/2, \delta/2)$-DP algorithm for estimating $X^T Y$, but when $M =$ `GaussMech` we call Algorithm 1 `ReuseCovGauss`.

Algorithm 1 Input: $n, \lambda, X \in \mathcal{X}^n \subset \mathbb{R}^{d \times n}$, $Y = [y_1, \dots y_l] \in \mathcal{Y}^{l \times n}$, privacy params: ϵ, δ

```
ReuseCov
Draw $E_1 \sim N_{d(d+1)/2}(0, \sigma_1^2)$, where $\sigma_1 = \frac{1}{n} c(\epsilon, \delta) ||\mathcal{X}||_2^2$
Compute $\hat{I} = (\frac{1}{n} X^T X + E_1 + \lambda I)$
Draw $\hat{v} = [\hat{v}_1, \dots \hat{v}_l] \sim \mathcal{M}(\epsilon/2, \delta/2, X, Y)$        ▷ $\mathcal{M} =$ GaussMech or Algorithm 2
for  $i = 1 \dots l$
    Set $\tilde{w_i} = \hat{I}^{-1} \hat{v}_i$        ▷ In practice computed via $\hat{I} =$ QR (Algorithm 3)
end for
Return $\tilde{W} = [\tilde{w_1}, \dots \tilde{w_l}]$
```

Theorem 1. *With $\mathcal{M} =$ GaussMech$(\epsilon/2, \delta/2, \Delta = \frac{1}{n}\sqrt{l}||\mathcal{X}||_2|\mathcal{Y}|)$, Algorithm 1 is an $(\alpha, \rho, \epsilon, \delta)$ solution to the PRIMO problem with*

$$\alpha = \tilde{O}\left(||\hat{w}||_2 \sqrt{\frac{d||\mathcal{X}||_2^4 ||\hat{w}||_2^2}{n^2} + \frac{ld|\mathcal{Y}|^2 ||\mathcal{X}||_2^2}{n^2}} \right), \tag{6}$$

where $||\hat{w}||_2^2 = \frac{1}{l}||W^*||_F^2$, and $\tilde{O}$ omits terms polynomial in $\frac{1}{\epsilon}, \log(1/\delta), \log(1/\rho)$.

Proof. The privacy proof follows from a straightforward application of the Gaussian mechanism. We note that releasing each $\hat{v}_i$ privately, is equivalent to computing $X^T Y + E_2$, where $E_2 \sim N_{d \times l}(0, \sigma_2^2)$. Now it is easy to compute l-sensitivity $\Delta(f)$ of $f(X) = X^T Y$. Fix an individual i, and an adjacent dataset $X' = X / \{x_i, y_i\} \cup \{x_i', y_i'\}$. Then $f(X) - f(X') = \Delta_V = [y_{i1} x_i - y_{i1}' x_{i1}', \dots y_{il} x_i - y_{il}' x_{il}']$. Then:

$$||f(X) - f(X')||_2 = \sqrt{||\Delta_V||_F^2} = \sqrt{\sum_{j=1}^{l} ||y_{ij} x_i - y_{ij}' x_{ij}'||_2^2} \leq \sqrt{l \cdot 4 ||\mathcal{X}||_2^2 ||\mathcal{Y}||_2^2} = 2\sqrt{l} ||\mathcal{X}||_2 ||\mathcal{Y}||_2$$

Hence setting $\sigma_2 = c(\epsilon, \delta) 2\sqrt{l} ||\mathcal{X}||_2 ||\mathcal{Y}||_2 / \epsilon$ by the Gaussian mechanism Dwork & Roth (2014) publishing $\hat{V}$ satisfies $(\epsilon/2, \delta/2) - DP$. Similarly if $g = X^T X$, $\Delta(g) \leq ||X||_2^2$, and so setting $\sigma_1 = c(\epsilon, \delta) ||X||_2^2 / \epsilon$, means publishing $\hat{I}$ is $(\epsilon/2, \delta/2)$-DP. By basic composition for DP, the entire mechanism is (ϵ, δ)-DP.

To prove the accuracy bound we follow the general proof technique developed in Wang (2018) analysing the accuracy guarantees of the ridge regression variant of `SSP` in the case $\lambda_{\min}(X^T X) = 0$, adding some mathematical detail to their exposition, and doing the appropriate book-keeping to handle our setting where the privacy level (as a function of the noise level) guaranteed by E_1 and E_2 differ. The reader less interested in these details can skip to Equation 11 below for the punchline.

Fix a specific index $i \in [l]$, and let $y = y_i$. We will analyze the prediction error of $\tilde{w_i}$ e.g. $F(\tilde{w_i}) - F(w_i^*)$. Then the following result is stated in Wang (2018) for which provide a short proof:

Lemma 2 (Wang (2018))**.**

$$f(\tilde{w_i}) - f(w_{i*}) = \frac{1}{n}(||y - X\tilde{w_i}||^2 - ||y - Xw_{i*}||) = \frac{1}{n}||\tilde{w_i} - w_{i*}||_{X^T X}^2$$

Proof. We note that all derivatives of orders higher than 2 of $f(w) = \frac{1}{n}||y - Xw||^2$ are zero, and that $\nabla f_{w_{i*}} = 0$ by the optimality of w_{i*}. We also note that the Hessian $\nabla^2 f_w = \frac{1}{n}X^T X$ at all points w. Then by the Taylor expansion of $f(w)$ around w_{i*}:

$$f(\tilde{w_i}) = f(w_{i*}) + (\tilde{w_i} - w_{i*}) \cdot \nabla f_{w_{i*}} + \frac{1}{n}(\tilde{w_i} - w_{i*})' X^T X(\tilde{w_i} - w_{i*})$$

Which using $\nabla f_{w_{i*}} = 0$ and rearranging terms gives the result. $\square$

Now Corollary 7 in the Appendix of Wang (2018) states (without proof) the below identity, which we provide a proof of for completeness via Lemma 12:

$$\tilde{w_i} - w_{i*} = (-X^T X + \lambda I + E_1)^{-1} E_1 w_{i*} - \lambda(X^T X + \lambda I + E_1)^{-1} w_{i*} + (X^T X + \lambda I + E_1)^{-1} E_2 \quad (7)$$

Hence, still following Wang (2018), for any psd matrix A, $||\tilde{w_i} - w_{i*}||_A^2 \leq$

$$3||(X^T X + \lambda I + E_1)^{-1} E_1 w_{i*}||_A^2 + 3\lambda^2||(X^T X + \lambda I + E_1)^{-1} w_{i*}||_A^2 + 3||(X^T X + \lambda I + E_1)^{-1} E_2||_A^2 \quad (8)$$

Lemma 3 (Wang (2018))**.** *With probability* $1 - \rho$, $||E_1|| \leq (\lambda_{\min}(X^T X) + \lambda)/2$, *and hence*

$$X^T X + \lambda I + E_1 \succ .5(X^T X + \lambda I)$$

We also remark that $||By||_A^2 = (By)^T ABy = ||y||_{B^T AB}^2$ for any vector y, and matrices A, B.

Hence, Inequality 8, with $A = X^T X$ can be further simplified to:

$$\begin{aligned} &3||(X^T X + \lambda I + E_1)^{-1} E_1 w_{i*}||_A^2 + 3\lambda^2||(X^T X + \lambda I + E_1)^{-1} w_{i*}||_A^2 + 3||(X^T X + \lambda I + E_1)^{-1} E_2||_A^2 \leq \\ &O\left(||E_1 w_{i*}||_{(X^T X + \lambda I + E_1)^{-1}}^2 + \lambda^2||w_{i*}||_{(X^T X + \lambda I + E_1)^{-1}}^2 + ||E_2||_{(X^T X + \lambda I + E_1)^{-1}}^2\right) \leq \\ &\text{(Lemma 3)} \ \ O\left(||E_1 w_{i*}||_{(X^T X + \lambda I)^{-1}}^2 + \lambda^2||w_{i*}||_{(X^T X + \lambda I)^{-1}}^2 + ||E_2||_{(X^T X + \lambda I)^{-1}}^2\right) \end{aligned} \quad (9)$$

By basic properties of the trace we have: $\text{tr}((\lambda I + X^T X)^{-1}) \leq d\lambda_{max}(\lambda I + X^T X^{-1}) = \frac{d}{\lambda_{\min}(\lambda I + X^T X)} = \frac{d}{(\lambda_{\min} + \lambda)}$, and $||w_{i*}||_{(X^T X + \lambda I)^{-1}}^2 \leq \frac{||w_{i*}||_2^2}{\lambda}$. Continuing from Wang (2018) by their Lemma 6, we can bound each $||E_1 w_{i*}||_{(X^T X + \lambda I)^{-1}}^2$ and $||E_2||_{(X^T X + \lambda I)^{-1}}^2$.

Lemma 4 (Wang (2018))**.** *Let* $w \in \mathbb{R}^d$ *and let* E *a symmetric Gaussian matrix where the upper triangular region is sampled from* $N(0, \sigma^2)$ *and let* A *be any psd matrix. Then with probability* $1 - \rho$*:*

$$||Ew||_A^2 \leq \sigma^2 tr(A)||w||_2^2 \log(2d^2/\rho)$$

Then recalling that:

- $\sigma_1^2 = \tilde{O}(||\mathcal{X}||_2^4/\epsilon^2)$
- $\sigma_2^2 = \tilde{O}(l||\mathcal{X}||_2^2|\mathcal{Y}|^2/\epsilon^2)$

Plugging into Lemma 4, and bringing it all together we get:

$$\begin{aligned} &|F(\tilde{w_i}) - F(w_{i*})| \leq ||w_{i*} - \tilde{w_i}||_{X^T X} = \\ &O\left(||E_1 w_{i*}||_{(X^T X + \lambda I)^{-1}}^2 + \lambda^2||w_{i*}||_{(X^T X + \lambda I)^{-1}}^2 + ||E_2||_{(X^T X + \lambda I)^{-1}}^2\right) = \\ &\tilde{O}\left(\frac{d}{\lambda_{\min} + \lambda}||w_{i*}||_2^2(||\mathcal{X}||_2^4/\epsilon^2)\log(2d^2/\rho) + \lambda||w_{i*}||_2^2 + \frac{d}{\lambda_{\min} + \lambda} l||\mathcal{X}||_2^2|\mathcal{Y}|^2/\epsilon^2 \log(2d^2/\rho)\right) \end{aligned} \quad (10)$$

Upper bounding Equation 10 by taking $\lambda_{\min} = 0$, we minimize over λ setting

$$\lambda = \tilde{O}\left(\frac{1}{\epsilon}\sqrt{d\log(2d^2/\rho)}||X||_2\sqrt{||\mathcal{X}||_2^2 + \frac{l|\mathcal{Y}|^2}{||w_{i*}||_2^2}}\right) \implies$$
$$|F(\tilde{w}_i) - F(w_{i*})| = \tilde{O}\left(\frac{1}{\epsilon}\sqrt{d\log(2d^2/\rho)}||X||_2\sqrt{||\mathcal{X}||_2^2||w_{i*}||_2^4 + l|\mathcal{Y}|^2||w_{i*}||_2^2}\right) \quad (11)$$

Now if we are in the $\eta-$small regime, we have $|\mathcal{Y}| \leq \eta||\mathcal{X}||_2||w_{i*}||_2$, and so

$$l|\mathcal{Y}|^2 w_{i*}^2 \leq l\eta^2 {||\mathcal{X}||_2}^2 {||w_{i*}||_2}^4,$$

which reduces Equation 11 to:

$$|F(\tilde{w}_i) - F(w_{i*})| = \tilde{O}\left(\frac{1}{\epsilon}\sqrt{d\log(2d^2/\rho)}||X||_2^2||w_{i*}||_2^2(\eta\sqrt{l})\right),$$

as desired.

□

Inspecting Theorem 1, we see that when $l < \min(\frac{n}{\sqrt{d}}, \frac{||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2})$, Algorithm 1 with $\mathcal{M}$ the Gaussian mechanism achieves error $\tilde{O}\left(\frac{\sqrt{d}||\mathcal{X}||_2^2||\hat{w}||_2^2}{n}\right)$. Since this matches the lower bound in Equation 3 for a single private regression in this case we say that we have achieved "PRIMO for Free!"

5 Improved Algorithms for Large l

The improvement of `ReuseCovGauss` over the naive PRIMO baseline in the previous section only applies in the parameter regime where the asymptotic error in Equation 5 is dominated by the covariance error term. In the regime where the association error term dominates, `ReuseCovGauss` still incurs a $\sqrt{l}$ multiplicative factor in the error term, which does not improve over the baseline. In this section we show that via a reduction from privately computing the association term to computing the image of a private vector under a public matrix, that under the weaker *feature differential privacy* we can obtain improved bounds for PRIMO over both the naive baseline and `ReuseCovGauss` when $l > \frac{n}{\sqrt{d}}$ (Subsection 5.1), and under *label differential privacy* we obtain improved bounds when $l > n^{2/3}$ (Subsection 5.2). In these cases, we obtain the surprising result that the MSE α has no explicit asymptotic dependence on l.

5.1 PRIMO Under Feature Differential Privacy

Algorithm 2 Input: $X \in \mathcal{X}^n \subset (\mathbb{R}^d)^n$, $Y = [y_1, \ldots y_l] \in \mathcal{Y}^{l\times n}$, privacy params: ϵ, δ.

```
GaussProj_Y
  Let $r = c(\epsilon,\delta)\frac{1}{n}\sup_i ||y^i||_2||\mathcal{X}||_2$
  Sample $w \sim N(0,1)^{dl}$
  Let $\tilde{g} = g + rw$, where $g = \frac{1}{n}X^TY$          ▷ Up to this point this is just GaussMech
  Formulate $C = C(Y) \in \frac{1}{n}\mathcal{Y}^{dl\times dn}$, vec(X).          ▷ $C$ is only used in the projection step
  Let $\hat{g} = \text{argmin}_{g\in K}||g - \tilde{g}||_2^2$, where $K = C(\sqrt{n}||\mathcal{X}||_2 B_1)$.
```

Let us reconsider the problem of privately computing the association term $\frac{1}{n}X^TY$ where $Y \in \mathcal{Y}^{n\times l}$, $X \in \mathcal{X}^n \subset \mathbb{R}^{n\times d}$, and Y is considered public, so $\mathcal{M}$ is only constrained to be feature DP. Let $\text{vec}(X) = (x_{11}, \ldots, x_{1n}, x_{21}, \ldots x_{d1}, \ldots x_{dn}) \in \mathbb{R}^{nd}$. Then since $\frac{1}{n}X^TY$ is *linear* in every entry of X, there exists a matrix $C(Y) \in \frac{1}{n}\mathcal{Y}^{dl\times dn}$ such that $\frac{1}{n}X^TY = C \cdot \text{vec}(X)$. We explicitly derive an expression for C in

Subsection 8.4 of the Appendix. Note that $||\text{vec}(X)||_2^2 \leq n||\mathcal{X}||_2^2$, hence $\frac{1}{n}X^TY \in C(\sqrt{n}||\mathcal{X}||_2 B_1)$, where $B_1 = \{x \in \mathbb{R}^{nd} : ||x||_2 \leq 1\}$.

Algorithm 2 is our projection-based subroutine for privately computing $\frac{1}{n}X^TY$, which we call $\texttt{GaussProj}_Y$. We first state the squared error of $\texttt{GaussProj}_Y$ in Theorem 2, and then translate this error into the PRIMO error of $\texttt{ReuseCovProj}_Y$ (Algorithm 1 with $\mathcal{M}$ = Algorithm 2) in Theorem 3. The crux of the proof is Lemma 8 from Nikolov et al. (2013), which uses a geometric argument to bound the error after the projection.

Theorem 2. *Let $\tilde{g}, g$ as in Line 4 of Algorithm 2. Then with probability $1-\rho$:*

$$||g - \tilde{g}||_2^2 = O\left(\frac{l\sqrt{d}c(\epsilon,\delta)\sqrt{\log(2/\rho)}|\mathcal{Y}|^2||\mathcal{X}||_2^2}{n}\right)$$

Proof. $\mathcal{M}$ is (ϵ, δ) differentially private by the Gaussian Mechanism and post-processing Dwork & Roth (2014). So we focus on the high probability accuracy bound. By Lemma 4 from Nikolov et al. (2013) we have that:

$$||\hat{g} - g||_2^2 \leq 4||rw||_{K^\circ} = r \sup_{x \in K} x \cdot w \tag{12}$$

Since $K \subset (n||\mathcal{X}||_2^2)CB_1$, and using the fact that the l_2 norm is self-dual, we have:

$$||\hat{g} - g||_2^2 \leq 4r||w||_{K^\circ} \leq 4r||\mathcal{X}||_2\sqrt{n} \sup_{z \in B_1} (Cz) \cdot w = 4r||\mathcal{X}||_2\sqrt{n}||C^Tw||_2 \tag{13}$$

So in order to bound $||\hat{g} - g||_2^2$ with high probability it suffices to bound $||C^Tw||_2$ with high probability. This is the content of the Hanson-Wright Inequality for anisotropic random variables Vershynin (2019).

Lemma 5 (Vershynin (2019))**.** *Let C^T an $m \times n$ matrix, and $X \sim N(0,1)^n \in \mathbb{R}^n$. Then for a fixed constant $c > 0$:*

$$\mathbb{P}[\ |\ ||C^TX|| - ||C||_F\ | > t] \leq 2exp(\frac{-ct^2}{||C||_F})$$

Lemma 9 shows that $||C^Tw||_2 = O(||C||_F\sqrt{\log(2/\rho)})$ with probability $1-\rho$. Given $x_{kj} \in X$, let $m = (k-1)n + j$ be the corresponding column of C, and let c^m denote this column. Then $||c^m||_2 = \frac{1}{n}||(y_{1j}, \ldots y_{lj})||_2 = \frac{1}{n}||y^j||_2 \leq \frac{\sqrt{l}}{n}$. Hence $||C||_F = \frac{\sqrt{d\sum_{i=1}^n ||y^i||_2^2}}{n}$.

Plugging in the value of r gives, with probability $1-\rho$:

$$\frac{1}{dl}\mathbb{E}[||\hat{g} - g||_2^2] = O\left(\frac{1}{dl} \cdot \sqrt{n}||\mathcal{X}||_2 \frac{\sqrt{d\sum_{i=1}^n ||y^i||_2^2}}{n}\sqrt{\log(2/\rho)} \cdot \frac{c(\epsilon,\delta)||\mathcal{X}||_2 \sup_i ||y^i||_2}{n}\right) \tag{14}$$

$$= O\left(\frac{c(\epsilon,\delta)\sqrt{\log(2/\rho)}\sqrt{\frac{1}{n}\sum_{i=1}^n ||y^i||_2^2}\sup_i ||y^i||_2||\mathcal{X}||_2^2}{nl\sqrt{d}}\right), \tag{15}$$

Since both terms involving Y in the numerator are $\leq \sqrt{l}|\mathcal{Y}|$ the bound follows. We note that since Y is public, in practice we can compute these terms rather than using $|\mathcal{Y}|$, the worst case bound. ☐

We note that the mean squared error of the Gaussian mechanism without the projection is $O(r^2) = O(\frac{l||\mathcal{X}||_2^2|\mathcal{Y}|^2}{n^2})$, which for $l\sqrt{d} \gg n$ is strictly larger than the error of the projection mechanism. We also note that the bound in Theorem 3 is strictly better than the error given by applying the Median Mechanism algorithm of Blum et al. (2011) for low-sensitivity queries, which is tailored for l_∞ error, and which also requires discrete $\mathcal{X}$. For example, when $\mathcal{Y} = \{0,1\}, \mathcal{X} = \{0,1\}^d$, then $|\mathcal{Y}| = 1, ||\mathcal{X}||_2 = \sqrt{d}$, and so Theorem 2 gives a bound of $\tilde{O}(\frac{\sqrt{d}}{n})$, whereas the Median Mechanism gives $\tilde{O}(\frac{d^{2/3}\log(dl)^2}{n^{2/3}})$ for the mean squared error. We now state the accuracy guarantees of $\texttt{ReuseCovProj}_Y$. The proof follows the proof of Theorem 1, substituting the bound from Theorem 2 into the $||E_2||_2^2$ term in Equation 7. We defer the full proof to the Appendix.

Theorem 3. *Let $\mathcal{A}$ denote the variant of Algorithm 1 with $\mathcal{M} = \texttt{GaussProj}_Y(X, Y, \epsilon/2, \delta/2)$. Then $\mathcal{A}$ is an $(\alpha, \rho, \epsilon, \delta)$ solution to the feature DP PRIMO problem with*

$$\alpha = \tilde{O}\left(||\hat{w}||_2\sqrt{\frac{d||\hat{w}||_2^2(||\mathcal{X}||_2^4)}{n^2} + \frac{\sqrt{d}|\mathcal{Y}|^2||\mathcal{X}||_2^2}{n}}\right),$$

where $\tilde{O}$ omits terms polynomial in $\frac{1}{\epsilon}, \log(1/\delta), \log(1/\rho)$.

Inspecting Theorem 3 in the $l > \frac{n}{\sqrt{d}}$ regime where the projection improves the error over the Gaussian Mechanism, when $n < \frac{\sqrt{d}||\mathcal{X}||_2^2||\mathcal{W}||_2^2}{|\mathcal{Y}|^2}$ is not too large, the dominant term in the error is $\tilde{O}\left(\frac{\sqrt{d}||\mathcal{X}||_2^2||\hat{w}||_2^2}{n}\right)$, and so we achieve PRIMO for Free. When n is sufficiently large the dominant error term is $\tilde{O}\left(\frac{||\hat{w}||_2 d^{1/4}|\mathcal{Y}|||\mathcal{X}||_2}{\sqrt{n}}\right)$ which is a factor of $\frac{n}{d^{1/4}}$ worse than the lower bound. We also note that the bound in Theorem 3 is strictly better than the error given by applying the Median Mechanism algorithm of Blum et al. (2011) for low-sensitivity queries, which is tailored for l_∞ error, and which also requires discrete $\mathcal{X}$. For example, when $\mathcal{Y} = \{0, 1\}, \mathcal{X} = \{0, 1\}^d$, then $|\mathcal{Y}| = 1, ||\mathcal{X}||_2 = \sqrt{d}$, and so Theorem 2 gives a bound of $\tilde{O}(\frac{\sqrt{d}}{n})$, whereas the Median Mechanism gives $\tilde{O}(\frac{d^{2/3}\log(dl)^2}{n^{2/3}})$ for the mean squared error.

5.2 PRIMO Under Label Differential Privacy

The setting where X is public and Y is considered private is strictly easier to satisfy than setting in which they are both considered private, and in particular in order for Algorithm 1 to satisfy label DP, we only need to add noise to terms that involve Y. Thus we can compute $\hat{I}$ in Line 2 of Algorithm 1 exactly without adding any noise E_1. We still need to compute the association term $\frac{1}{n}X^TY$ privately in Y, where now the privacy guarantee holds over rows of Y rather than rows of X. Computing $\frac{1}{n}X^TY$ privately is equivalent to computing $\frac{1}{n}Y_{l\times n}^T X_{n\times d}$, where we've added subscripts indicating the dimensions of the matrices. Then by direct analogy to the previous section, we can compute $\frac{1}{n}X^TY$ via Algorithm 2, where Theorem 2 holds by switching X and Y, and d, l. We call this variant of Algorithm 2 $\texttt{GaussProj}_X$. Thus invoking Theorem 2 the error in computing the association term is with probability $1 - \rho$:

$$||g - \tilde{g}||_2^2 = O\left(\frac{c(\epsilon, \delta)\sqrt{\log(2/\rho)}|\mathcal{X}|^2||\mathcal{Y}^l||_2^2 d\sqrt{l}}{n}\right) \tag{16}$$

The error in Equation 16 improves over the error given by $\mathcal{M} = \texttt{GaussMech}$ when $O(\frac{d\sqrt{l}}{n}) < O(\frac{dl^2}{n^2}) \implies l > n^{\frac{2}{3}}$, which we summarise in row 5 of Figure 1. Interestingly, in contrast to the public label setting, this condition has no dependence on the dimension d. Substituting Equation 16 into Equation 7 with $E_1 = 0$ since we are in the label-private setting and don't have to add noise to the covariance matrix, we can optimize over λ to get:

Theorem 4. *Let $\mathcal{A}$ denote the label-private variant of Algorithm 1 with $E_1 = 0, \mathcal{M} = \texttt{GaussProj}_X(X, Y, \epsilon/2, \delta/2)$. Then $\mathcal{A}$ is an $(\alpha, \rho, \epsilon, \delta)$ solution to the label DP PRIMO problem with*

$$\alpha = \tilde{O}\left(||\hat{w}||_2\frac{|\mathcal{X}|||\mathcal{Y}^l||_2\sqrt{d}}{nl^{1/4}}\right)$$

We remark that while the form of Theorem 4 appears to suggest that the MSE decreases with increasing l, which is counterintuitive, l actually appears in the numerator through $||\mathcal{Y}^l||_2 = O(|\mathcal{Y}|\sqrt{l})$, giving MSE with (mild) dependence $O(l^{1/4})$ on l.

6 Computational Efficiency

In this section we first analyze the computational complexity of Algorithm 1, which we implement more efficiently in Algorithm 3. Theorem 5 shows this is $O(\max(\min(nl^2, n^2l), nld, nd^2, ld^2, d^3))$, which for $n > d > l$ is $O(nd^2)$. Subsection 6.2 shows how sketching the covariance matrix using s points can maintain accuracy while reducing this to $O(sd^2)$.

6.1 Computational Complexity

The computational complexity of Algorithm 1 can be broken down into 3 components:

- Step 1: Forming $X^T X$, $\big(nd^2$ or much faster$\big)$
- Step 2: the cost of computing $\hat{I}^{-1}\hat{v}_i = (X^T X + \lambda I + E_1)^{-1}\hat{v}_i \ \forall i \in [l]$, $\big(d^3 + ld^2\big)$
- Step 3: In the case where $\mathcal{M}$ is the projection algorithm, computing the projection $\text{argmin}_{v \in C(\mathcal{X}^n)} ||v - \hat{v}_i||_2^2$, $(nl \min(n, l) + nd + nld$ via diagonalization)

Forming the covariance matrix $X^T X$ is a matrix multiplication of two $d \times n$ matrices, which can be done via the naive matrix multiplication in time $O(nd^2)$, and via a long-line of "fast" matrix multiplication algorithms in time $O(d^{2+\alpha(n)})$; For example if $n < d^{.3}$ it can be done in time that is essentially $O(d^2)$ Gall (2012). Step 2 can be completed by solving the equation $\hat{I}\hat{w}_i = \hat{v}_i, i = 1 \ldots l$ via the conjugate gradient method, which takes time $O(\gamma(\hat{I})d^2 \log(1/\epsilon))$ to compute an ϵ-approximate solution Mahoney (2011) where $\gamma(\hat{I})$ is the condition number. We note that this has to be done separately for each $i = 1 \ldots l$ giving total time $O(l \cdot \gamma(\hat{I})d^2 \log(1/\epsilon))$. Alternatively, an exact solution $\hat{I}^{-1}\hat{v}_i$ can be computed directly using the QR decomposition of the matrix $\hat{I}$. The decomposition $\hat{I} = QR$ can be computed in time $O(d^3)$ Mahoney (2011) and does not depend on the $\hat{v}_i$, after which using $R\hat{w}_i = Q^T\tilde{v}_i$, $\hat{w}_i$ can be computed in time $O(d^2)$ via backward substitution. This gives a total time complexity of $O(d^3 + ld^2)$. So if d and $\frac{1}{\gamma(\hat{I})}$ are sufficiently small relative to l, e.g. if $l = \tilde{\Omega}(d/(\gamma(\hat{I})))$, it will be faster to use the QR decomposition based method. The projection in Line 5 corresponds to minimizing a quadratic over a sphere. Setting $A = C^T C \in \mathbb{R}^{dn \times dn}, b = 2C^T\tilde{g} \in \mathbb{R}^{dl}$, then $\hat{v} = (Cn||\mathcal{X}||_2^2)x$, where $x \in \mathbb{R}^{nd}$ is the minimizer of:

$$\min_{x \in B_1} x^t A x - b^t x \tag{17}$$

$$\text{s.t. } ||x||_2 \leq \sqrt{n}||\mathcal{X}||_2 \tag{18}$$

Now, given the spectral decomposition of $A = U\Lambda U^T$, and the coordinates of b in the eigenbasis $U^T b$, Lemma 2.2 in Hager (2001) gives a simple closed form for x that computes each coordinate in constant time. Since there are nd coordinates of x, this incurs an additional additive factor of $O(nd)$ in the complexity, which is dominated by the cost of diagonalizing A. So the complexity of this step is the complexity of diagonalizing $A = C^T C$, or equivalently finding the right singular vectors of C, plus the complexity of computing $U^T b$. This is seemingly bad news, as $C \in \mathbb{R}^{dn \times dl}$ is a very high-dimensional matrix, and the complexity for computing the SVD of C without any assumptions about its structure is $O(d^3 ln \min(l, n))$ Golub & Van Loan (1996). However, it is evident from the construction of C in Subsection 5 that

$$C = I_d \otimes \frac{1}{n} Y^T,$$

where $\otimes$ is the Kronecker product. Then if $L\Lambda V^T$ is the SVD of Y^T, standard properties of the Kronecker product imply that the spectral decomposition of $C^T C$ is:

$$C = \frac{1}{n} \cdot (I_d \otimes L)(I_d \otimes \Lambda)(I_d \otimes V^T) \implies \tag{19}$$

$$C^T C = (I_d \otimes V)(I_d \otimes \Lambda^2)(I_d \otimes V^T) \tag{20}$$

Hence we can compute SPEC(C) in the time it takes to compute SVD(Y), or $O(nl \min(n, l))$. Similarly, to efficiently compute the $U^T b$ term required for Lemma 2.2 Hager (2001) we can again take advantage of properties of the Kronecker product, $U^T b =$

$$(I_d \otimes V)^T 2C^T\tilde{g} = 2(I_d \otimes V^T)(I_d \otimes \frac{1}{n}Y)\tilde{g} =$$

$$2(I_d \otimes \frac{1}{n}V^T y)\tilde{g} = \text{vec}(\frac{2}{n}V^T Y \text{mat}(\tilde{g})),$$

where $\text{mat}(\tilde{g})$ is the $l \times d$ matrix with row i given by elements $(l(i-1)+1, \ldots, l(i-1)+l$ of $\tilde{g}$, and the last equality follows properties of the Kronecker product. Now $V^T Y = \Lambda U^T$ which can be computed in $O(ln)$ since Λ is diagonal. Multiplying by $\text{mat}(\tilde{g})$ can be done in another $O(nld)$, for total complexity of $O(nl \max(\min(n,l), d))$.

Putting the complexity of these steps together we get:

Theorem 5. *The complexity of Algorithm 3 is* $O(\max(\min(nl^2, n^2 l), nld, nd^2, ld^2, d^3))$.

Algorithm 3 Input: $\lambda, \mathcal{X} \in \mathcal{X}^n \subset \mathbb{R}^{d\times n}$, $Y = [y_1, \ldots y_l] \in \mathcal{Y}^{l\times n}$, privacy params: ϵ, δ. We denote by $\mathcal{B}$ the Algorithm in Lemma 2.2 Hager (2001)

```
λ − ReuseCov
Draw $E_1 \sim N_{d(d+1)/2}(0, \sigma_1^2)$, where $\sigma_1 = \frac{1}{n} 2\sqrt{2\log(2.5/\delta)}||\mathcal{X}||_2^2/\epsilon$
Compute $\hat{I} = (\frac{1}{n} X^T X + E_1 + \lambda I)$
Compute the QR decomposition $\hat{I} = QR$
Draw $\hat{v} = [\hat{v}_1, \ldots \hat{v}_l] \sim$ GaussMech($\epsilon/2, \delta/2, \Delta = \frac{1}{n}\sqrt{l}||\mathcal{X}||_2|\mathcal{Y}|$)
Compute SVD($Y^T$) $= U\Lambda V^T$
Compute $\Pi_{C(Y)}\hat{v} = \mathcal{B}(V, \Lambda, \hat{v})$
for $i = 1 \ldots l$
    Solve $R\hat{w}_i = Q^T \hat{v}_i$ by back substitution.
end for
Return $\hat{W} = [\hat{w}_1, \ldots \hat{w}_l]$
```

6.2 Sketching `ReuseCov`

The discussion in the previous section shows that when $n > d > l$, the complexity of Algorithm 3 is $O(nd^2)$ or the cost of forming the covariance matrix. In this section we show how sub-sampling $s < n$ points can improve this to $O(sd^2)$ by giving an analysis of sub-sampled SSP. The key ingredient is marrying the convergence of the sub-sampled covariance matrix to $X^T X$ with the accuracy analysis of SSP we saw in Section 4.

Algorithm 4 Input: $s, \lambda, \mathcal{X} \in \mathcal{X}^n \subset \mathbb{R}^{d\times n}$, $Y = [y_1, \ldots y_l] \in \mathcal{Y}^{l\times n}$, privacy params: ϵ, δ

```
λ − SubSampReuseCov
Sub-sample $s$ points without replacement from $\mathcal{X}$, we denote the sub-sampled design matrix by $X_S$.
Let $(\epsilon_1, \delta_1) = (\frac{n}{s}\epsilon/2, \delta/2)$
Draw $E_1 \sim N_{d(d+1)/2}(0, \sigma_1^2)$, where $\sigma_1 = \frac{1}{n} 2\sqrt{2\log(2.5/\delta)}||\mathcal{X}||_2^2/\epsilon_1$
Compute $\hat{I}_s = (\frac{1}{s} X_S^T X_S + E_1 + \lambda I)$
Draw $\hat{v} = [\hat{v}_1, \ldots \hat{v}_l] \sim$ GaussMech($\epsilon/2, \delta/2, \Delta = \frac{1}{n}\sqrt{l}||\mathcal{X}||_2|\mathcal{Y}|$)
for $i = 1 \ldots l$
    Set $\hat{w}_i = \hat{I_s}^{-1} \hat{v}_i$
end for
Return $\hat{W} = [\hat{w}_1, \ldots \hat{w}_l]$
```

Our algorithm is based on the observation that if we sub-sample $S \subset [n], |S| = s$ points without replacement then:

- The cost of computing the covariance matrix $\Sigma_S = \sum_{k\in S} x_k x_k^T$ is $O(sd^2)$
- By the "secrecy of the sub-sample" principle Dwork & Roth (2014), our privacy cost for estimating Σ_S is scaled down by a factor of s/n
- With high probability for sufficiently large s, $\Sigma_S \to \Sigma$ by a matrix-Chernoff bound for sampling without replacement Tropp (2010)

Theorem 6. *With $\mathcal{M} = \texttt{GaussMech}(\epsilon/2, \delta/2, \Delta = \frac{1}{n}\sqrt{l}||\mathcal{X}||_2|\mathcal{Y}|)$, Algorithm 4 is an $(\alpha, \rho, O(\epsilon), \delta)$ solution to the PRIMO problem with $\alpha^2 =$*

$$O(||\hat{w}||_2^2||\mathcal{X}||_2^2|\mathcal{Y}|^2(\frac{\log(2d/\rho)}{s}) + \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s})\cdot \\ (\frac{||X||_2^4 n^2}{s^4\epsilon^2}||\mathcal{W}||_2^2 + \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2})) \tag{21}$$

Proof Sketch. Our analysis will hinge on the case where $l = 1$ e.g. that of standard private linear regression, which we will extend to the PRIMO case by our choice of ϵ as in the proof of Theorem 1. Now let:

- $w_{i*} = (X^TX)^{-1}X^TY$ the least squares estimator as before
- $w_{i*}^\lambda = (\frac{1}{n}X^TX + \lambda I)^{-1}\frac{1}{n}X^TY$ the ridge regression estimator
- $w_s = (\frac{1}{s}X_S^TX_S + \lambda I)^{-1}(\frac{1}{n}X^TY)$ the sub-sampled least squares estimator
- $\tilde{w}_s = (\frac{1}{s}X_S^TX_S + E_1 + \lambda I)^{-1}(\frac{1}{n}X^TY + E_2)$ our differentially private estimate of w_s

We note that $0 \leq f(w_{i*}^\lambda) - f(w_{i*}) \leq \lambda\left(||w_{i*}||_2^2 - ||w_{i*}^\lambda||_2^2\right) \leq \lambda||\mathcal{W}||_2^2$. Then by the Lemma 2 and Cauchy-Schwartz with respect to the norm $||\cdot||_{\frac{X^TX}{n}}$:

$$|f(\tilde{w}_s) - f(w_{i*})| = ||\tilde{w}_s - w_{i*}||_{\frac{X^TX}{n}}^2 \leq \\ 3||w_{i*} - w_{i*}^\lambda||_{\frac{X^TX}{n}}^2 + 3||w_{i*}^\lambda - w_s||_{\frac{X^TX}{n}}^2 + 3||w_s - \tilde{w}_s||_{\frac{X^TX}{n}}^2 \leq \\ 3\lambda||\mathcal{W}||^2 + \underbrace{3||w_s - w_{i*}^\lambda||_{\frac{X^TX}{n}}^2}_{\text{Matrix Chernoff}} + \underbrace{3||\tilde{w}_s - w_s||_{\frac{X^TX}{n}}^2}_{\text{SSP analysis+Matrix Chernoff}} \tag{22}$$

So it suffices to bound each term with high probability. The second term, $||\tilde{w}_s - w_s||_{X^TX+\lambda I}$ can be bounded using the same arguments as in Theorem 1, with small differences due to scaling. Crucially though, as we need to bound this in the norm induced by X^TX rather than $X_S^TX_S$, we will need to utilize the convergence of $X_S^TX_S \to X^TX$ via Matrix-Chernoff bounds.

Lemma 6. *Under the assumption $||\mathcal{X}||_2 = O(n\lambda)$, then w.p. $1 - \rho/2$:*

$$||w_s - \tilde{w}_s||_{\frac{X^TX}{n}} = O(\frac{||X||_2^4}{n^2\epsilon^2}||\mathcal{W}||_2^2 \cdot \frac{d}{\lambda}\frac{\log(2d/\rho)}{s} + \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2} \cdot \frac{d}{\lambda}\frac{\log(2d/\rho)}{s}) \tag{23}$$

Bounding the first term can be reduced to bounding $||I - (\frac{1}{n}X^TX)^{-1/2}(\frac{1}{s}X_S^TX_S)(\frac{1}{n}X^TX)^{-1/2}||_2$ which follows more directly via the Matrix-Chernoff bound for sub-sampling without replacement:

Lemma 7. *Under the assumption $||\mathcal{X}||_2 = O(n\lambda)$, then w.p. $1 - \rho/2$:*

$$||w_{i*}^\lambda - w_s||_{\frac{X^TX}{n}} = O\left(\frac{||\mathcal{X}||_2|\mathcal{Y}|\log(2d/\rho}{\lambda s}\right)$$

Substituting into Equation 23 and minimizing over λ gives the desired result.

7 Experiments

Datasets. We evaluate the accuracy of our algorithms `ReuseCovGauss` and `ReuseCovProj`$_X$ on the task of genomic risk prediction using real X data from two of the largest publicly available databases, and simulated outcomes Y so that we can easily vary the number of outcomes l. In addition, in Appendix 8.6 we include experiments on two additional datasets over smaller values of l, one constructed by sub-sampling MNIST Deng (2012) and generating synthetic outcomes from a noisy linear model, and the other from entirely synthetic Gaussian data with outcomes generated either from a 2-layer MLP or a noisy linear model.

The genomic datasets are from two sources: the 1000 Genomes project (1KG) Fairley et al. (2019), and the Database of Genomes and Phenotypes Mailman et al. (2007) (accession phs000688.v1.p1). The datasets contain $n = 5008, 6042$ haplotypes at $d = 78961, 16721$ SNPs, respectively. Whenever experiments are run with a fixed value of $d < 78961, 16721$, we have randomly sub-sampled d SNPs without replacement. In order to vary the number of phenotypes l we generate synthetic phenotype data using our haplotype dataset. After centering our haplotype matrix X by subtracting off the row means we generate synthetic phenotypes y_i for $i = 1 \dots l$ by generating a random $\theta_i \sim \mathcal{N}(0, \frac{I_d}{\sqrt{d}}), y_{ij} \sim \theta_i \cdot x_j + \mathcal{N}(0, 1)$. In order to evaluate our algorithms at larger values of n than exist in our genomic datasets, we train a GAN on the dbGaP dataset following Olagoke et al. (2023), and use it to generate synthetic samples to simulate having up to a million data points.

Experimental Details. For a given dataset and setting of (n, d, l) we run $\texttt{ReuseCovProj}_X$ and `ReuseCovGauss` 10 different times, and calculate the resulting average MSE. We study two different regimes, one where n is of moderate size corresponding to real genomic datasets, d is small, and l is varied from 1 to $1e5$ (Figure 2). The other regime models the scenario where in the near future genomic databases will have millions of individuals (e.g. UK Biobank already has > 400000 samples Privé (2022)), generating $n = 1e6$ samples from the GAN trained on dbGaP and again varying l from 1 to $1e5$ (Figure 4). Note that we do not have experiments with larger values of d for computational reasons; as discussed in Section 6 the runtime of both algorithms is quadratic in d.

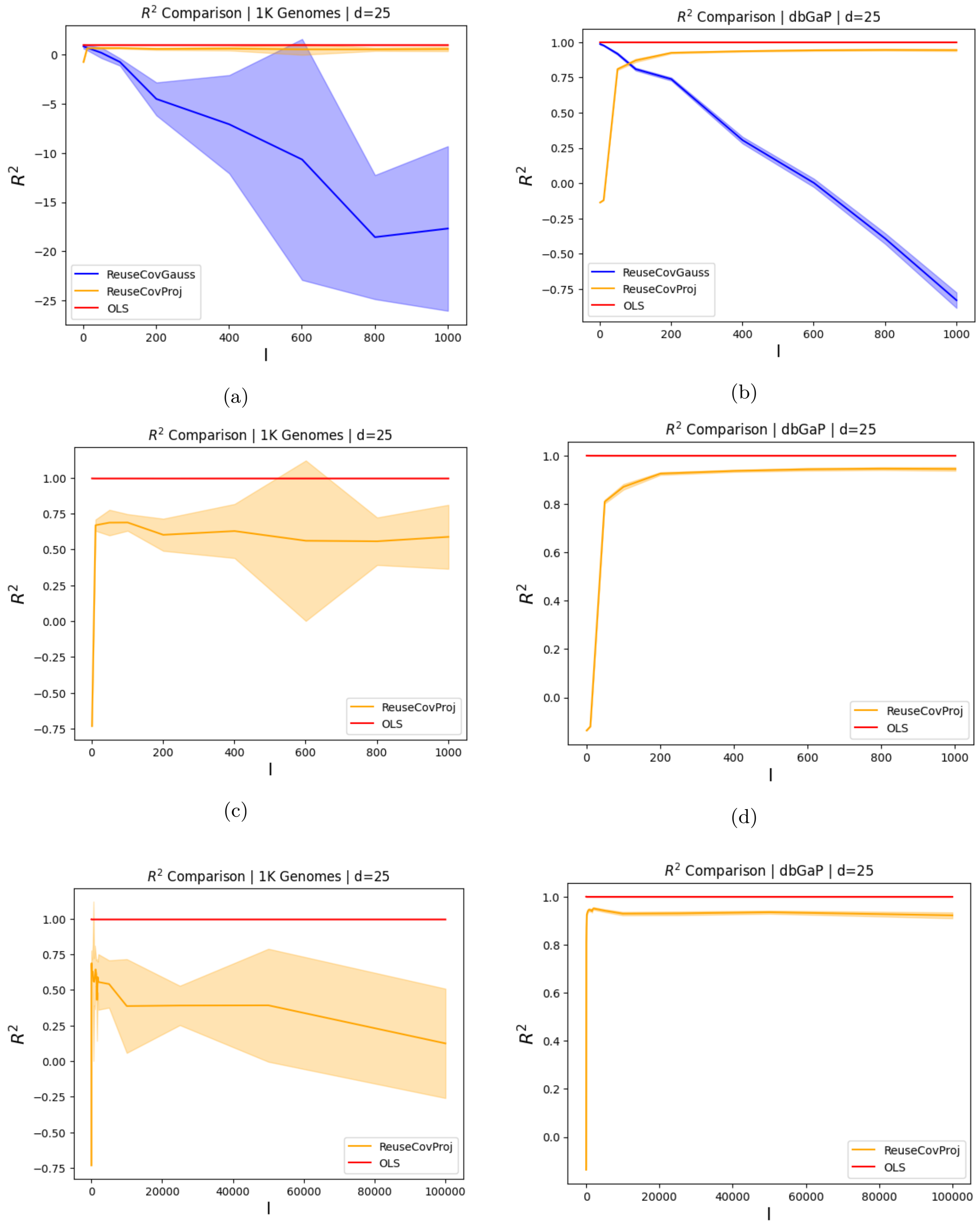
Rather than plotting the MSE α directly, in Figure 2 we plot the coefficient of determination R^2, which is equivalent in that $R^2 = 1 - \frac{\alpha}{\frac{1}{nl} SS_{tot}}$. We use R^2 because it is easier to interpret given that achieving error better than the constant predictor implies $R^2 \in (0, 1]$. In Figures 3d,4 we also visualize the results using a different metric, the log of the ratio of the MSE of the private predictor to the MSE of the optimal non-private predictor, where a larger ratio indicates a greater multiplicative error.

Realistic n, varying l. We find that for values of l starting as small as 11 for 1KG and 101 for dbGaP, $\texttt{ReuseCovProj}_X$ achieves higher R^2 than `ReuseCovGauss`, and is able to achieve non-trival R^2 for large l, which is consistent with our Theorem 3. For $l < 11$ on 1KG and $l < 101$ on dbGaP, `ReuseCovGauss` achieves higher R^2. In addition to being consistent with the theory, this is intuitive; when l is small the bias induced by the projection outweighs the reduction in the overall noise. Figures 2a, 2b show that for $l > 101, 801$ on 1KG and dbGaP respectively `ReuseCovGauss` has negative R^2 – by contrast, zooming in on the R^2 curve for `ReuseCovProj` in Figures 2c, 2d we are able to achieve non-trivial R^2 at all values of l, and see minimal increase in error as l increases.

Large n, varying l. Our upper bounds suggest that for large values of n, `ReuseCovGauss` will outperform $\texttt{ReuseCovProj}_X$, and that the MSE should increase polynomially with increasing l, or equivalently linearly in $\log l$ when we take the log ratio. Both of these trends can be seen in Figure 4(a). Given that the relative performance of $\texttt{ReuseCovProj}_X$ is improving as l increases, we expect that for larger values of l that were computationally prohibitive to run at such large n, $\texttt{ReuseCovProj}_X$ would again be the superior algorithm.

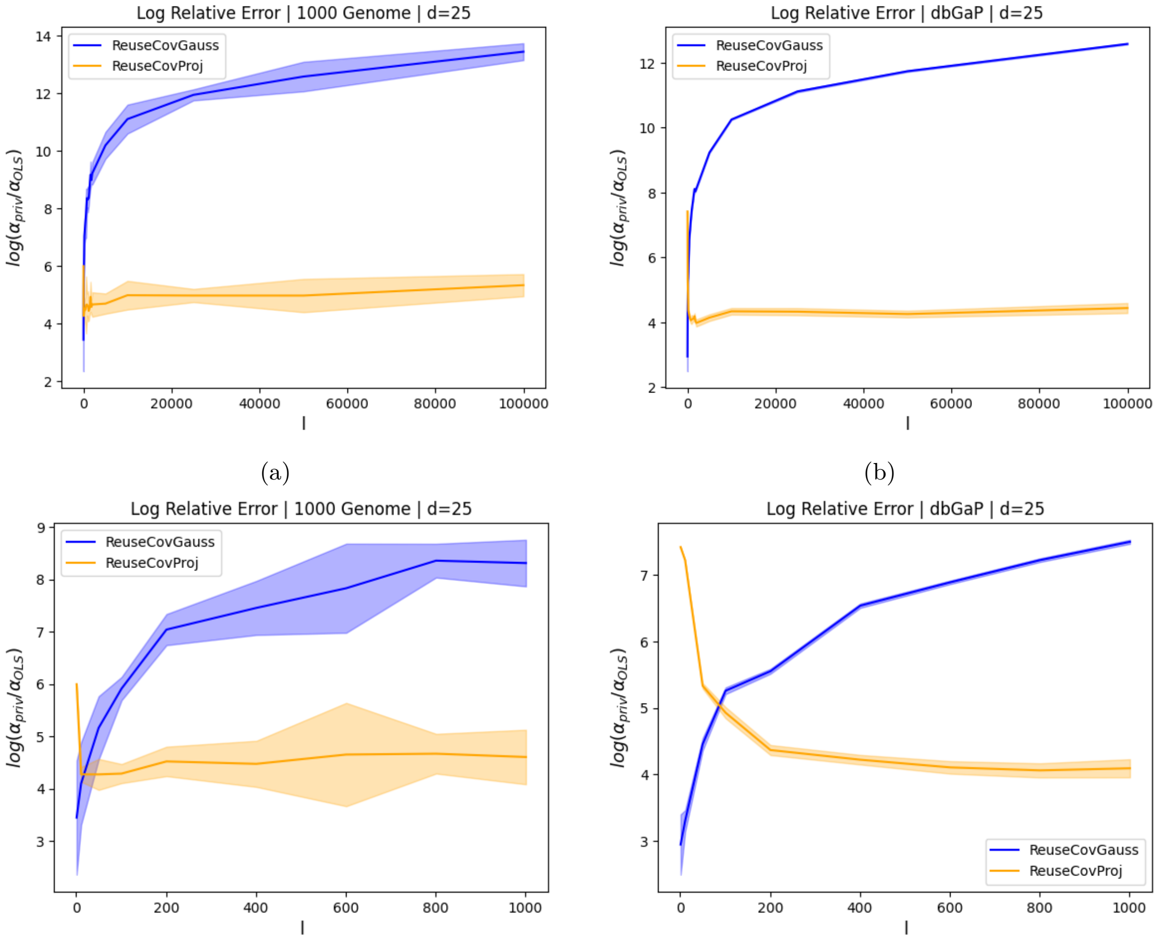
Additional Ablations. In the setting where $l < \frac{n}{\sqrt{d}}$ (Row 2 of Table 1), theory suggests our algorithms should not outperform naive composition of private regressions. Since private regression methods like `DP-SGD` Bassily et al. (2019) are known to equal or outperform `SSP`, our `ReuseCov` algorithms which are variants of `SSP` might actually perform worse than composing `DP-SGD` across l regressions. We construct such a dataset and show this is in fact the case in Figure 6 in the Appendix.

The experiments above on genomic data generate outcomes Y from a noisy linear model; in Figure 5 in the Appendix we show that generating outcomes from a 2-layer MLP with noise doesn't change the relative ordering of the algorithms by performance.



(a) (b) (c) (d) (e) (f)

Figure 2: Each R^2 value is averaged over 10 iterations. The shaded area around the lines indicates the error bars for the R^2 value at a given value of (l, d). In Figures (a)-(d) we plot the average R^2 for $d = 25, l = (1, 11, 101, 201, 401, 601, 801, 1001)$, fixing $(\epsilon, \delta) = (5, \frac{1}{n^2})$ with $n = 5008, 6042$. In Figures (e)-(f) we show l up to $1e5$.



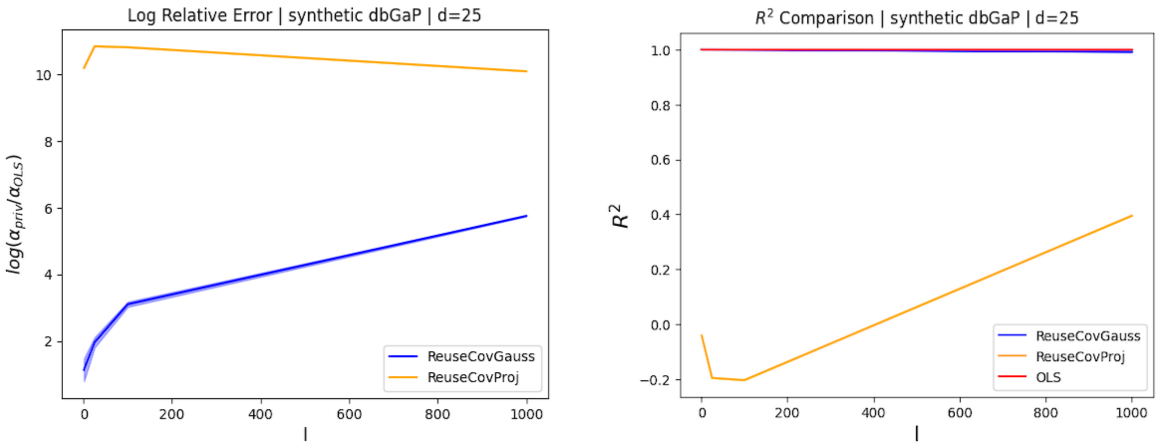
(a) (b)

(c) (d)

Figure 3: Comparing the log of the ratio of the squared loss of the private estimator to the square loss of the OLS estimator. Each value is averaged over 10 iterations. The shaded area around the lines indicates the error bars at the given value of l. (a) and (b) range l up to $100,000$, (c) and (d) show $l = (1, 11, 101, 201, 401, 601, 801, 1001)$ while we fixed $d = 25$ and $(\epsilon, \delta) = (5, \frac{1}{n^2})$ with $n = 5008, 6042$ for 1000 Genomes and dbGaP respectively

7.1 Limitations

There are 3 main criticisms that can be levied at the results in this paper. The first is that the reductions in error in the feature DP and label DP settings due to projection in Lines $3-5$ of Figure 1 rely on sufficiently large $l : l > \frac{n}{\sqrt{d}}, l > n^{2/3}$ respectively. As a result, there are practical settings where n is particularly large or l is small where the theory does not suggest `ReuseCovProj` will outperform `ReuseCovGauss`. The second critique is that this improvement requires us relaxing our notion of DP to label or feature DP. In settings where both X and Y are highly sensitive, if we want to guarantee full DP `ReuseCovGauss` would be is our only option. Finally, all of our algorithms are variants of `SSP` Vu & Slavkovic (2009), which has complexity that scales super-linearly in d due to forming the covariance matrix X^TX. For high-dimensional regression settings algorithms like private SGD are preferred due to their scalability Chaudhuri et al. (2011). With respect to the first critique, we note that in common settings, like in genomics, d can be very large relative to n, and so $\frac{n}{\sqrt{d}}$ can be quite small in practice. For example, in the popular Fairley et al. (2019) dataset used in the experiments, $n = 5008$, and $d = 78961$, and so $\frac{n}{\sqrt{d}} \approx 17$. Similiarly, if n is of modest size, it is plausible



(a) (b)

Figure 4: (a) shows the log of the ratio of the squared loss of the private estimator to the square loss of the OLS estimator and (b) shows the R^2 values. For both of these, the value was averaged over 10 iterations. The shaded area around the lines indicates the error bars at a given value of l. We plot this for $l = (1, 10, 25, 100, 500, 1000, 5000)$ with $d = 25$, $(\epsilon, \delta) = (5, \frac{1}{n^2})$ and $n = 5 \cdot 1e5$ on a synthetic dbGaP dataset.

that $l > n^{2/3}$. We also note that our experimental results show that the projection improves error at values of l *much* smaller than the theory would suggest, making this algorithm the practical choice for values of l as small as 11! Finally, the `ReuseCovGuass` Algorithm is itself also a contribution of this work. On the issue of scalability in Subsections 6.1, 6.2 in the Appendix we consider the computational complexity of our methods. When $n > d > l$, the computational complexity of the `ReuseCovGauss` algorithm is the cost of computing the covariance matrix $X^T X$, which is $O(nd^2)$. In Subsection 6.2 in the Appendix we show this can be reduced to $O(sd^2)$, where $s < n$ is the number of rows sub-sampled without replacement from X, and we quantify the resulting error from sampling (Theorem 6). Despite these improvements, the computational complexity of our methods is still super-linear in d.

Obtaining versions of PRIMO that scale for large values of d, and that do not require relaxing full DP in order to improve the dependence on l as we do in Section 5, are exciting directions for future work.

References

Alan Agresti and Finlay Barbara. Statistical methods for the social sciences. *Pearson Prentice Hall*, 2009.

Sergül Aydöre, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Amaresh Ankit Siva. Differentially private query release through adaptive projection. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 457–467. PMLR, 2021. URL `http://proceedings.mlr.press/v139/aydore21a.html`.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization, revisited. *CoRR*, abs/1405.7085, 2014. URL `http://arxiv.org/abs/1405.7085`.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Thakurta. Private stochastic convex optimization with optimal rates. *CoRR*, abs/1908.09970, 2019. URL `http://arxiv.org/abs/1908.09970`.

Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan R. Ullman. Algorithmic stability for adaptive data analysis. *SIAM J. Comput.*, 50(3), 2021. doi: 10.1137/16M1103646. URL `https://doi.org/10.1137/16M1103646`.

Brett Beaulieu-Jones, Zhiwei Wu, Chris Williams, Ran Lee, Sanjeev Bhavnani, James Byrd, and Casey Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12, 07 2019. doi: 10.1161/CIRCOUTCOMES.118.005122.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. *CoRR*, abs/1109.2229, 2011. URL `http://arxiv.org/abs/1109.2229`.

Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *CoRR*, abs/1311.3158, 2013. URL `http://arxiv.org/abs/1311.3158`.

T. Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy, 2020.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(29):1069–1109, 2011. URL `http://jmlr.org/papers/v12/chaudhuri11a.html`.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Michal Derezinski and Manfred K. Warmuth. Unbiased estimates for linear regression via volume sampling. *CoRR*, abs/1705.06908, 2017. URL `http://arxiv.org/abs/1705.06908`.

Michal Derezinski, Manfred K. Warmuth, and Daniel Hsu. Tail bounds for volume sampled linear regression. *CoRR*, abs/1802.06749, 2018. URL `http://arxiv.org/abs/1802.06749`.

Christos Dimitrakakis, Blaine Nelson, , Zuhe Zhang, Aikaterini Mitrokotsa, and Benjamin Rubinstein. Bayesian differential privacy through posterior sampling, 2016. URL `https://arxiv.org/abs/1306.1066`.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for l2 regression and applications. In *SODA '06*, 2006.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. URL `http://dblp.uni-trier.de/db/journals/fttcs/fttcs9.html#DworkR14`.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay (ed.), *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006. doi: 10.1007/11761679_29. URL `https://doi.org/10.1007/11761679_29`.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506.02629, 2015a. URL `http://arxiv.org/abs/1506.02629`.

Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discret. Comput. Geom.*, 53(3):650–673, 2015b. doi: 10.1007/s00454-015-9678-x. URL `https://doi.org/10.1007/s00454-015-9678-x`.

Alexander Edmonds, Aleksandar Nikolov, and Jonathan R. Ullman. The power of factorization mechanisms in local and central differential privacy. *CoRR*, abs/1911.08339, 2019. URL `http://arxiv.org/abs/1911.08339`.

Hossein Esfandiari, Vahab S. Mirrokni, Umar Syed, and Sergei Vassilvitskii. Label differential privacy via clustering. *CoRR*, abs/2110.02159, 2021. URL `https://arxiv.org/abs/2110.02159`.

Susan Fairley, Ernesto Lowy-Gallego, Emily Perry, and Paul Flicek. The International Genome Sample Resource (IGSR) collection of open human genomic variation resources. *Nucleic Acids Research*, 48(D1): D941–D947, 10 2019. ISSN 0305-1048. doi: 10.1093/nar/gkz836. URL `https://doi.org/10.1093/nar/gkz836`.

James R. Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving bayesian data analysis. *CoRR*, abs/1603.07294, 2016. URL `http://arxiv.org/abs/1603.07294`.

François Le Gall. Faster algorithms for rectangular matrix multiplication. *CoRR*, abs/1204.1111, 2012. URL `http://arxiv.org/abs/1204.1111`.

Andrew Gelman and Eric R. Loken. The statistical crisis in science. *American Scientist*, 102:460, 2014. URL `https://api.semanticscholar.org/CorpusID:146855235`.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. On deep learning with label differential privacy. *CoRR*, abs/2102.06062, 2021. URL `https://arxiv.org/abs/2102.06062`.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release, 2011.

William W. Hager. Minimizing a quadratic over a sphere. *SIAM J. Optim.*, 12:188–208, 2001.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 61–70, 2010. doi: 10.1109/FOCS.2010.85.

Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLOS Genetics*, 4(8):1–9, 08 2008. doi: 10.1371/journal.pgen.1000167. URL `https://doi.org/10.1371/journal.pgen.1000167`.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 2021.

Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam D. Smith, and Jonathan R. Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In Leonard J. Schulman (ed.), *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pp. 775–784. ACM, 2010. doi: 10.1145/1806689.1806795. URL `https://doi.org/10.1145/1806689.1806795`.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal (eds.), *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pp. 392–401. ACM, 1993. doi: 10.1145/167088.167200. URL `https://doi.org/10.1145/167088.167200`.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In Shie Mannor, Nathan Srebro, and Robert C. Williamson (eds.), *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pp. 25.1–25.40, Edinburgh, Scotland, 25–27 Jun 2012. PMLR. URL `https://proceedings.mlr.press/v23/kifer12.html`.

Keegan Korthauer, Patrick K. Kimes, Claire Duvallet, Alejandro Reyes, Ayshwarya Subramanian, Mingxiang Teng, Chinmay Shukla, Eric J. Alm, and Stephanie C. Hicks. A practical guide to methods controlling false discoveries in computational biology. *Genome Biology*, 20(1):118, 2019. doi: 10.1186/s13059-019-1716-1. URL `https://doi.org/10.1186/s13059-019-1716-1`.

E. Krapohl, H. Patel, S. Newhouse, C J Curtis, S. von Stumm, P S Dale, D. Zabaneh, G. Breen, P F O'Reilly, and R. Plomin. Multi-polygenic score approach to trait prediction. *Molecular Psychiatry*, 23(5):1368–1374, 2018. doi: 10.1038/mp.2017.163. URL `https://doi.org/10.1038/mp.2017.163`.

Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *CoRR*, abs/1508.03261, 2015. URL `http://arxiv.org/abs/1508.03261`.

Michael W. Mahoney. Randomized algorithms for matrices and data. *CoRR*, abs/1104.5557, 2011. URL `http://arxiv.org/abs/1104.5557`.

Matthew D Mailman, Michael Feolo, Yan Jin, Michi Kimura, Kimberly Tryka, Rinat Bagoutdinov, Li Hao, Alex Kiang, Justin Paschall, Lon Phan, Natalia Popova, Sherri Pretel, Luda Ziyabari, Ming Lee, Yongzhao Shao, Zhining Y Wang, Karl Sirotkin, Michael Ward, Maxim Kholodov, Karen Zbicz, Judith Beck, Michael Kimelman, Sergey Shevelev, Don Preuss, Elena Yaschenko, Alexander Graeff, James Ostell, and Stephen T Sherry. The ncbi dbgap database of genotypes and phenotypes. *Nature genetics*, 39(10):1181–1186, 2007.

Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *CoRR*, abs/1808.03537, 2018. URL `http://arxiv.org/abs/1808.03537`.

Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. *CoRR*, abs/1901.09136, 2019. URL `http://arxiv.org/abs/1901.09136`.

Jason Milionis, Alkis Kalavasis, Dimitris Fotakis, and Stratis Ioannidis. Differentially private regression with unbounded covariates. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3242–3273. PMLR, 28–30 Mar 2022. URL `https://proceedings.mlr.press/v151/milionis22a.html`.

Ilya Mironov. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017. URL `http://arxiv.org/abs/1702.07476`.

S. Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In Howard J. Karloff and Toniann Pitassi (eds.), *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pp. 1285–1292. ACM, 2012. doi: 10.1145/2213977.2214090. URL `https://doi.org/10.1145/2213977.2214090`.

Seth V. Neel, Aaron L. Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 72–93, 2019. doi: 10.1109/FOCS.2019.00014.

Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *STOC '13*, 2013.

Lukman Olagoke, Salil Vadhan, and Seth Neel. Black-box training data identification in gans via detector networks, 2023.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2016. URL `https://arxiv.org/abs/1610.05755`.

Jack Pattee and Wei Pan. Penalized regression and model selection methods for polygenic scores on summary statistics. *PLOS Computational Biology*, 16(10):1–27, 10 2020. doi: 10.1371/journal.pcbi.1008271. URL `https://doi.org/10.1371/journal.pcbi.1008271`.

Florian Privé. Using the UK biobank as a global reference of worldwide populations: application to measuring ancestry diversity from GWAS summary statistics. *Bioinform.*, 38(13):3477–3480, 2022. doi: 10.1093/bioinformatics/btac348. URL `https://doi.org/10.1093/bioinformatics/btac348`.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls, 2009. URL `https://arxiv.org/abs/0910.2042`.

Reza Shokri, Marco Stronati, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *CoRR*, abs/1610.05820, 2016. URL `http://arxiv.org/abs/1610.05820`.

Justin Thaler, Jonathan R. Ullman, and Salil P. Vadhan. Faster algorithms for privately releasing marginals. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer (eds.), *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pp. 810–821. Springer, 2012. doi: 10.1007/978-3-642-31594-7_68. URL `https://doi.org/10.1007/978-3-642-31594-7_68`.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-CGAN: differentially private synthetic data and label generation. *CoRR*, abs/2001.09700, 2020. URL `https://arxiv.org/abs/2001.09700`.

Joel A. Tropp. Improved analysis of the subsampled randomized hadamard transform. *CoRR*, abs/1011.1595, 2010. URL `http://arxiv.org/abs/1011.1595`.

Jonathan R. Ullman and Salil P. Vadhan. Pcps and the hardness of generating private synthetic data. In Yuval Ishai (ed.), *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pp. 400–416. Springer, 2011. doi: 10.1007/978-3-642-19571-6_24. URL `https://doi.org/10.1007/978-3-642-19571-6_24`.

Salil P. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, 2017.

Roman Vershynin. *High-Dimensional Probability.* Cambridge University Press, 2019. URL `https://www.math.uci.edu/~rvershyn/papers/HDP-book/HDP-book.pdf`.

Duy Vu and Aleksandra Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 138–143, 2009. doi: 10.1109/ICDMW.2009.52.

Yu-Xiang Wang. Revisiting differentially private linear regression: optimal and adaptive prediction and estimation in unbounded domain, 2018. URL `https://arxiv.org/abs/1803.02596`.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. *CoRR*, abs/1808.00087, 2018. URL `http://arxiv.org/abs/1808.00087`.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1zk9iRqF7`.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

8 Appendix

8.1 Additional Related Work

Query Release. In Subsection 5 we show that privately computing the association term $\frac{1}{n}X^TY$ is equivalent to the problem of differentially privately releasing a set of $l \cdot d$ low-sensitivity queries Bassily et al. (2021). While less is known about the optimal l_2 error for arbitrary low-sensitivity queries, it is clear that geometric techniques based on factorization and projection do not (at least obviously) apply, since there is no corresponding notion of the query matrix $A_\mathcal{Q}$ Edmonds et al. (2019). In the case where X, Y are both private, this corresponds to releasing a subset of $l \cdot d$ 2-way marginal queries over $d + l$ dimensions, which is well-studied Dwork et al. (2015b); Thaler et al. (2012); Ullman & Vadhan (2011). In the related work section of the Appendix we discuss why applying existing algorithms for private release of marginals seems unlikely be able to both improve over the Gaussian Mechanism and achieve non-trivial error. However, in the less restrictive but still practically relevant setting where the labels Y are public, computing the association term $\frac{1}{n}X^TY$ is equivalent to the problem of differentially privately releasing a special class of dl “low-sensitivity" queries we term “inner product queries" (Definition 6). For this special class of queries we can adapt the projected Gaussian Mechanism of Aydöre et al. (2021) that is optimal for linear queries under l_2 error in the sparse regime. We are able to obtain greatly improved results over the naive Gaussian Mechanism in the regime where $l > \frac{n}{\sqrt{d}}$, as summarised in Figure 1 and presented in Subsection 5. This is a key regime of interest, particularly for genomic data, where d the number of SNPs could be in the hundreds of millions, n the population size in the hundreds of thousands ($\sim 400K$ for UK Biobank Privé (2022) one of the most popular databases), and l the number of recorded phenotypes could be in the thousands.

Projection mechanisms and Algorithm 2. Since we make heavy use of projection mechanisms in Subsection 5 in the *public label* setting, we elaborate on the difference between the existing methods and our Algorithm 2. The most technically related work to our Algorithm 2 is Aydöre et al. (2021), which is itself a variant of the projection mechanism of Nikolov et al. (2013). There are several key differences in our analysis and application.The projection in Nikolov et al. (2013) is designed for linear queries over a discrete domain, and runs in time polynomial in the domain size. Algorithm 2 allows continuous $\mathcal{X}, \mathcal{Y}$ and runs in time polynomial in n, d, l. Like the relaxed projection mechanism Aydöre et al. (2021), when our data is discrete we relax our data domain to be continuous in order to compute the projection more efficiently. Unlike in their setting which attempts to handle general linear queries, due to the linear structure of inner product queries our projection can be computed in polynomial time via linear regression over the l_2 ball, as opposed to solving a possibly non-convex optimization problem. Moreover, due to this special structure we can use the same geometric techniques as in Nikolov et al. (2013) to obtain theoretical accuracy guarantees (Theorem 2).

Private release of 2-way marginals Consider the problem of privately releasing all 2-way marginals over n points in $\{0, 1\}^{d+l}$. Theorem 5.7 in Dwork et al. (2015b) gives a polynomial time algorithm based on relaxed projections that achieves mean squared error $\tilde{O}(n\sqrt{d + l})$, which matches the best known information theoretic upper bound Dwork et al. (2015b), although there is a small gap to the existing lower bound $\min(n, (d + l)^2)$. This relaxed projection algorithm outperforms the Gaussian Mechanism when $n\sqrt{d + l} < ld \implies n < \frac{dl}{\sqrt{d+l}} \implies d\sqrt{d + l} > n$. Since the mean squared error of Algorithm 1 that used this projection as a subroutine is at least $\frac{d\sqrt{d+l}}{n}$, this means that in the regime where the projection outperforms the Gaussian Mechanism, we do not achieve mean squared error < 1 in our regression.

This suggests that, at least using the existing error analysis of `SSP` from Wang (2018), it seems unlikely that by applying specialized algorithms for private release of 2-way marginals to compute the $\frac{1}{n}X^TY$ term subject to differential privacy in (X, Y), e.g. the label private setting, we can both improve over the Gaussian Mechanism and achieve non-trivial error.

Linear Queries under l_∞-loss. Beyond 2-way marginals, the problem of privately releasing large numbers of linear queries (Definition 4) has been studied extensively. It is known that the worst case error is bounded by $\min(\frac{\sqrt{\log(|\mathcal{Q}|)}(\log|\mathcal{X}|\log(1/\delta))^{1/4}}{\sqrt{n\epsilon}}, \frac{\sqrt{|\mathcal{Q}|\log(1/\delta)}}{\epsilon n})$. The first term, which dominates in the so-called low-accuracy or “sparse” (Nikolov et al. (2013)) regime, is achieved by the *PrivateMultiplicativeWeights* algorithm of Hardt

& Rothblum (2010), which is optimal over worst case workloads Bun et al. (2013). However, this algorithm has running time exponential in the data dimension, which is unavoidable Vadhan (2017) over worst case $\mathcal{Q}$. The second term, which dominates for $n \gg \frac{|\mathcal{Q}|}{\log|\mathcal{Q}|\log|\mathcal{X}|^{1/4}}$, the "high accuracy" regime, is achieved by the simple and efficient Gaussian Mechanism Dwork & Roth (2014), which is also optimal over worst-case sets of queries $\mathcal{Q}$ Bun et al. (2013).

Linear Queries under l_2-loss. For the l_2 error, in the high accuracy $n \gg |\mathcal{Q}|$ regime the factorization mechanism achieves error that is exactly tight for any workload of linear queries $\mathcal{Q}$ up to a factor of $\log(1/\delta)$, although it is not efficient (Theorem Edmonds et al. (2019)). In the low-accuracy regime, the algorithm of Nikolov et al. (2013) that couples careful addition of correlated Gaussian noise (akin to the factorization mechanism) with an l_1-ball projection step achieves error within log factors of what is (a slight variant) of a quantity known as the *hereditary discrepancy* $\text{opt}_{\epsilon,\delta}(A, n)$ (Theorem Nikolov et al. (2013)). This quantity is a known lower bound on the error of any (ϵ, δ) mechanism for answering linear queries Muthukrishnan & Nikolov (2012), and so the upper bound is tight up to log factors in $|\mathcal{Q}|, |\mathcal{X}|$. Theorem 21 in Nikolov et al. (2013) analyzes the simple projection mechanism that adds independent Gaussian noise and projects rather than first performing the decomposition step that utilizes correlated Gaussian noise, achieving error $O(nd\log(1/\delta)\sqrt{\log|X|}/\epsilon)$, which matches the best known (worst case over $\mathcal{Q}$) upper bound for the sparse $n < d$ case Gupta et al. (2011). In our Theorem 2 we give such a universal upper bound, rather than one that depends on the hereditary discrepancy of the matrix Y. While the bound can of course be improved for a specific set of outcomes Y by the addition of the decomposition step to the projection algorithm, we omit this step in favor of a simpler algorithm with more directly comparable bounds to existing private regression algorithms.

While the information-theoretic l_∞ or l_2 error achievable for linear queries is well-understood Edmonds et al. (2019); Nikolov et al. (2013); Bun et al. (2013), as synthetic data algorithms like *PrivateMultiplicativeWeights* and *MedianMechanism*, or the factorization or projection mechanisms are in general inefficient, there are many open problems pertaining to developing efficient algorithms for specific query classes, or heuristic approaches that work better in practice. Examples of these approaches along the lines of the factorization mechanism McKenna et al. (2018; 2019), efficient approximations of the projection mechanism Aydöre et al. (2021); Dwork et al. (2015b), and using heuristic techniques from distribution learning in the framework of iterative synthetic data algorithms Yoon et al. (2019); Torkzadehmahani et al. (2020); Beaulieu-Jones et al. (2019); Neel et al. (2019).

Sub-sampled Linear Regression. In Subsection 6.2 we analyze `SSP` where we first sub-sample a random set of s points without replacement, and use this sub-sample to compute the noisy covariance matrix. Sub-sampled linear regression has been studied extensively absent privacy, where it is known that uniform sub-sampling is sub-optimal in that it produces biased estimates of the OLS estimator, and performs poorly in the presence of high-leverage points Derezinski et al. (2018). To address these shortcomings, techniques based on leverage score sampling Drineas et al. (2006), volume-based sampling Derezinski & Warmuth (2017)Derezinski et al. (2018), and spectral sparsification Lee & Sun (2015) have been developed. Crucially, in these methods the probability of a point being sub-sampled is data-dependent, and so they are (not obviously) compatible with differential privacy.

8.2 Definitions

Throughout the paper we make heavy use of common matrix and vector norms. For a vector $v \in \mathbb{R}^d$ and matrix $A \in \mathbb{R}^{d\times d}$, $||v||_A^2 := v^t A v, ||A||_2^2 = \lambda_{\max}(A^T A), ||v||_2^2 = \sum_{k=1}^d v_k^2, ||A||_F^2 = \sum_{i,j\in[d]} a_{ij}^2$.

Definition 4. *Statistical Query Kearns (1993) Let $D \in \mathcal{X}^n$ a dataset. A linear query is a function $q : \mathcal{X} \to [0, 1]$, where $q(D) := \frac{1}{n}\sum_{x_i \in D}^n q(x_i)$.*

Definition 5. *Kasiviswanathan et al. (2010) Let $\mathcal{X} = \{0, 1\}^m$, and $\mathcal{Q}_k = \{q_{ij}\}_{1\le i_1<i_2<i_k\le m}$, where $q_{ij}(x) := \prod_{u=1}^k x_{i_u}$. Then the class $\mathcal{Q}_k$ are called $k-$way marginals.*

Lastly, we define a new class of "low-sensitivity" queries Bassily et al. (2021) we call "inner product" queries that we will make heavy use of in Subsection 5.

Definition 6. *Inner Product Query. Let $\mathcal{X} \subset \mathbb{R}^d$ and $X \in \mathcal{X}^n \subset \mathbb{R}^{d \times n}$ a dataset with column j corresponding to $x_j \in \mathcal{X}$. Let $y \in \mathbb{R}^n$, and $i \in [d]$. Then the pair (i, y) defines an inner product query $q_{(i,y)} : \mathcal{X}^n \to \mathbb{R}$,*

$$q_{(i,y)}(X) := \frac{1}{n}\sum_{j=1}^{n} X_{ij} y_j = \frac{1}{n} e_i^T X y,$$

where e_i is the i^{th} basis vector in $\mathbb{R}^d$.

Changing one row j of X, changes at most one summand $\frac{1}{n} X_{ij} y_j$ of $q_{(i,y)}(X)$, and hence the l_2 sensitivity $\Delta_2(q_{(i,y)}) \leq \frac{1}{n}|X_{ij}||y_j| \leq \frac{|\mathcal{X}||\mathcal{Y}|}{n}$, a fact we will use throughout.

8.3 Lemmas

Lemma 8 (Raskutti et al. (2009))**.** *Let $K \subset \mathbb{R}^d$ be a symmetric convex body, let $g \in K$, and $\tilde{g} = g + w$ for some $w \in \mathbb{R}^d$. Then if $\hat{g} = argmin_{g' \in K} ||\tilde{g} - g'||_2^2$, then*

$$||\hat{g} - g||_2^2 \leq \min\{4||w||_2^2, 4||w||_{K^\circ}\}$$

Lemma 9 (Vershynin (2019))**.** *Let C^T an $m \times n$ matrix, and $X \sim N(0,1)^n \in \mathbb{R}^n$. Then for a fixed constant $c > 0$:*

$$P(||C^T X||_F - ||C||_F > t) \leq 2e^{\frac{-ct^2}{||C||_F}}$$

Lemma 10. *Tropp (2010) Let $Z_i \in \mathbb{S}_d^+$, and sample $Z_1, \ldots Z_s$ without replacement from $\{Z_1, \ldots Z_n\}$. Suppose $\mathbb{E}[Z_i] = I_d$, and $\max_{i \in [n]} \lambda_{\max}(Z_i) \leq B$. Then for $\delta = \sqrt{\frac{2B\log(2d/\rho)}{s}}$, with probability $1 - \rho$:*

$$\lambda_{\max}(\frac{1}{s}\sum_{i=1}^{s} Z_i) < 1 + \delta, \quad \lambda_{\min}(\frac{1}{s}\sum_{i=1}^{s} Z_i) > 1 - \delta$$

Lemma 11 (folklore e.g. Wang et al. (2018))**.** *Given a dataset $\mathcal{X}^n$ of n points and an $(\epsilon, \delta)-DP$ mechanism M. Let the procedure **subsample** take a random subset of s points from $\mathcal{X}^n$ without replacement. Then if $\gamma = s/n$, the procedure $M \circ$ **subsample** is $(O(\gamma\epsilon), \gamma\delta)-DP$ for sufficiently small ϵ.*

The following lemma is used repeatedly in analyzing the accuracy of all `SSP` variants.

Lemma 12. *Let A, B invertible matrices in $\mathbb{R}^{n \times n}$, and v, c vectors $\in \mathbb{R}^n$.*

Then

$$A^{-1}v - (A+B)^{-1}(v+c) = (A+B)^{-1}BA^{-1}v - (A+B)^{-1}c$$

Proof. Expanding we get that:

$$A^{-1}v - (A+B)^{-1}(v+c) = (A^{-1} - (A+B)^{-1})v - (A+B)^{-1}c,$$

so it suffices to show that:

$$(A^{-1} - (A+B)^{-1})v = (A+B)^{-1}BA^{-1}v$$

Now the Woodbury formula tells us that $(A+B)^{-1} = A^{-1} - (A + AB^{-1}A)^{-1}$, hence

$$(A^{-1} - (A+B)^{-1})v = (A + AB^{-1}A)^{-1}v = (A(I + B^{-1}A))^{-1}v$$

Then since:

$$(A(I + B^{-1}A))^{-1} = (I + B^{-1}A)^{-1}A^{-1} = (B^{-1}(B+A))^{-1A^{-1}} = (B+A)^{-1}BA^{-1},$$

we are done. ☐

8.4 Proofs from Subsection 5

Computing the association term. Consider entry (k, j) of $\frac{1}{n}X^TY$ for $k \in [d], j \in [l]$, $a_{kj} = (\frac{1}{n}X^TY)_{kj} = e_k\frac{1}{n}X^Ty_j = q_{(k,y_j)}(X)$, $q_{(k,y_j)}$ is the inner product query of Definition 6. Thus privately computing $\frac{1}{n}X^TY$, is equivalent to privately releasing answers to dl inner product queries $\{q_{(k,y_j)}(X)\}_{k\in[d],j\in[l]}$.

It will be convenient for us to write $q_{(k,y_j)}(X)$ as a single inner product. Let $\text{vec}(X) = (x_{11}, \ldots, x_{1n}, x_{21}, \ldots x_{d1}, \ldots x_{dn}) \in \mathbb{R}^{nd}$, and given $y = \text{vec}(X)$, let $\text{mat}(y) = X$. Denote by $c_{kj} \in \mathbb{R}^{nd}$ the vector that has all zeros except in positions $(k-1)n+1, \ldots kn$ it contains $\frac{1}{n}y_{1j}, \ldots \frac{1}{n}y_{nj}$. Then it is clear that $q_{(k,y_j)}(X) = c_{kj}^T\text{vec}(X)$, so if we let $C \in (\frac{1}{n}\mathcal{Y})^{dl\times dn}$ be the matrix with row $kj \in [dl]$ equal to c_{kj}, then $\frac{1}{n}X^TY = C\cdot\text{vec}(X)$.

Proof of Theorem 4:

Proof. We start with our usual expansion of $f(\tilde{w}_i) - f(w_{i*})$, up until Equation 10, we have with probability $1-\rho$ for every $i \in [l]$:

$$n \cdot f(\tilde{w}_i) - f(w_{i*}) = \tilde{O}\left(\frac{d}{\lambda_{\min}+\lambda}||w_{i*}||_2^2(||\mathcal{X}||_2^4/\epsilon^2)\log(2d^2/\rho) + \lambda||w_{i*}||_2^2 + \frac{1}{\lambda_{\min}+\lambda}||E_{2i}||_2^2\right) \tag{24}$$

Aggregating over i and rearranging gives:

$$\frac{n}{l}\sum_{i=1}^{l} f(\tilde{w}_i) - f(w_{i*}) = \tilde{O}\left(\frac{d}{\lambda_{\min}+\lambda}||\tilde{w}||_2^2(||\mathcal{X}||_2^4/\epsilon^2)\log(2d^2/\rho) + \lambda||\tilde{w}||_2^2 + \frac{1}{\lambda_{\min}+\lambda}\frac{1}{l}\sum_{i=1}^{l}||E_{2i}||_2^2\right), \tag{25}$$

where $||\tilde{w}||_2^2 = \frac{1}{l}\sum_{i=1}^{l}||w_{i*}||_2^2 = \frac{1}{l}||W^*||_F^2$. Then by Theorem 2, we have with $\frac{1}{l}\sum_{i=1}^{l}||E_{2i}||_2^2 = O\left(c(\epsilon,\delta)\sqrt{\log(2/\rho)}n\sqrt{d}|\mathcal{Y}|^2||\mathcal{X}||_2^2\right)$ with probability $1-\rho$. So with probability $1-2\rho$, we have $\frac{n}{l}\sum_{i=1}^{l} f(\tilde{w}_i) - f(w_{i*}) =$

$$\tilde{O}\left(\frac{d}{\lambda_{\min}+\lambda}||\tilde{w}||_2^2(||\mathcal{X}||_2^4/\epsilon^2)\log(2d^2/\rho) + \lambda||\tilde{w}||_2^2 + \frac{1}{\lambda_{\min}+\lambda}c(\epsilon,\delta)\sqrt{\log(2/\rho)}n\sqrt{d}|\mathcal{Y}|^2||\mathcal{X}||_2^2\right) \tag{26}$$

Finally optimizing over λ gives the desired result:

$$\alpha = \tilde{O}\left(||\tilde{w}||_2\sqrt{\frac{d||\tilde{w}||_2^2(||\mathcal{X}||_2^4/\epsilon^2)\log(2d^2/\rho)}{n^2} + \frac{c(\epsilon,\delta)\sqrt{\log(2/\rho)}\sqrt{d}|\mathcal{Y}|^2||\mathcal{X}||_2^2}{n}}\right)$$

□

8.5 Proofs from Subsection 6.2

Proof of Theorem 6:

Proof. Our analysis will hinge on the case where $l = 1$ e.g. that of standard private linear regression, which we will extend to the PRIMO case by our choice of ϵ as in the proof of Theorem 1. The fact that the Algorithm is $(O(\epsilon), \delta)$ private follows immediately from the Gaussian mechanism, and the secrecy of the sub-sample lemma (Lemma 11), which is why we can set $\epsilon_1 = \frac{n}{s}\epsilon/2$ in Line 2. We proceed with the accuracy analysis.

Define:

- $w_{i*} = (X^TX)^{-1}X^TY$: the least squares estimator as before
- $w_{i*}^{\lambda} = (\frac{1}{n}X^TX + \lambda I)^{-1}\frac{1}{n}X^TY$: the ridge regression estimator

- $w_s = (\frac{1}{s}X_S^T X_S + \lambda I)^{-1}(\frac{1}{n}X^T Y)$: the sub-sampled least squares estimator
- $\tilde{w}_s = (\frac{1}{s}X_S^T X_S + E_1 + \lambda I)^{-1}(\frac{1}{n}X^T Y + E_2)$: differentially private estimate of w_s

We note that $0 \leq f(w_{i*}^{\lambda}) - f(w_{i*}) \leq \lambda \left(||w_{i*}||_2^2 - ||w_{i*}^{\lambda}||_2^2\right) \leq \lambda||\mathcal{W}||_2^2$. Then by the Lemma 2 and Cauchy-Schwartz with respect to the norm $|| \cdot ||_{\frac{X^TX}{n}}$:

$$\begin{aligned}|f(\tilde{w}_s) - f(w_{i*})| = ||\tilde{w}_s - w_{i*}||_{\frac{X^TX}{n}}^2 \leq \\ 3||w_{i*} - w_{i*}^{\lambda}||_{\frac{X^TX}{n}}^2 + 3||w_{i*}^{\lambda} - w_s||_{\frac{X^TX}{n}}^2 + 3||w_s - \tilde{w}_s||_{\frac{X^TX}{n}}^2 \leq \\ 3\lambda||\mathcal{W}||_2^2 + 3||w_s - w_{i*}^{\lambda}||_{\frac{X^TX}{n}}^2 + 3||\tilde{w}_s - w_s||_{\frac{X^TX}{n}}^2\end{aligned} \tag{27}$$

Lemmas 13, 14 bound these terms with high probability.

Lemma 13. *Under the assumption* $||\mathcal{X}||_2 = O(n\lambda)$, *then w.p.* $1 - \rho/2$:

$$||w_{i*}^{\lambda} - w_s||_{\frac{X^TX}{n}} = O\left(\frac{||\mathcal{X}||_2|\mathcal{Y}|\log(2d/\rho)}{\lambda s}\right)$$

Lemma 14. *Under the assumption* $||\mathcal{X}||_2 = O(n\lambda)$, *then w.p.* $1 - \rho/2$:

$$||w_s - \tilde{w}_s||_{\frac{X^TX}{n}} = O(\frac{||X||_2^4}{n^2\epsilon^2}||\mathcal{W}||_2^2 \cdot \frac{d}{\lambda}\frac{\log(2d/\rho)}{s} + \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2} \cdot \frac{d}{\lambda}\frac{\log(2d/\rho)}{s}) \tag{28}$$

Then by Equation 27 and Lemmas 13, 14, we get that w.p. $1 - \rho$:

$$\begin{aligned}&f(\tilde{w}_s) - f(w_{i*}) = \\ &O\left(\lambda||w_{i*}||_2^2 + \frac{||\mathcal{X}||_2|\mathcal{Y}|}{\lambda}(\frac{\log(2d/\rho)}{s}) + \frac{||X||_2^4 n^2}{s^4\epsilon^2}||\mathcal{W}||_2^2 \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s}) \;+\; \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2} \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s})\right),\end{aligned} \tag{29}$$

Summing over i and minimizing over λ we set

$$\lambda = \frac{\sqrt{||\mathcal{X}||_2|\mathcal{Y}|(\frac{\log(2d/\rho)}{s}) + \frac{||X||_2^4 n^2}{s^4\epsilon^2}||\mathcal{W}||_2^2 \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s}) \;+\; \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2} \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s})}}{\sqrt{1}l||W||_F},$$

which completes the result. ☐

Proof of Lemma 13:

Proof. Now:

$$||w_{i*}^{\lambda} - w_s||_{\frac{1}{n}X^TX}^2 = n||w_{i*}^{\lambda} - w_s||_{\frac{1}{n}X^TX}^2 \leq n||w_{i*}^{\lambda} - w_s||_{\frac{1}{n}X^TX+\lambda I}^2$$

We will focus on $||w_{i*}^{\lambda} - w_s||_{\frac{1}{n}X^TX+\lambda I}^2$. Let $\Sigma = \frac{1}{n}X^TX + \lambda I$, $\Sigma_s = \frac{1}{s}\sum_{j\in S} x_s x_s^T + \lambda I$, and $v = \frac{1}{n}X^TY$. Expanding $||w_{i*}^{\lambda} - w_s||_{\Sigma} =$

$$(\Sigma_s^{-1}v - \Sigma^{-1}v)^T\Sigma(\Sigma_s^{-1}v - \Sigma^{-1}v) = v^T \overbrace{(\Sigma_s^{-1} - \Sigma^{-1})\Sigma(\Sigma_s^{-1} - \Sigma^{-1})}^{A} v = v^T A v \tag{30}$$

Now since A is Hermitian, we know $||v||_A \leq ||v||||A||_2$. Since $||v|| \leq ||\mathcal{X}||_2|\mathcal{Y}|$, it suffices to bound $||A||_2$ with high probability. Noting that $A = (\Sigma_s^{-1} - \Sigma^{-1})\Sigma(\Sigma_s^{-1} - \Sigma^{-1}) = \Sigma_s^{-1}\Sigma^{1/2}(I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2})(I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2})\Sigma^{1/2}\Sigma_s^{-1}$, we have by the sub-multiplicativity of the operator norm:

$$||A||_2 \leq \left(||\Sigma_S^{-1}\Sigma^{1/2}||_2^2\right) \cdot \left(||I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2}||_2^2\right) \leq \frac{\lambda_{\max}(\Sigma)}{\lambda^2} \cdot \left(||I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2}||_2^2\right) \quad (31)$$

Now consider $\Sigma^{-1/2}\Sigma_s\Sigma^{-1/2} = \Sigma^{-1/2}\frac{1}{s}\sum_{j\in S}(x_s x_s^T + \lambda I)\Sigma^{-1/2} = \frac{1}{s}\sum_{j\in s} Z_i$. Then note that $\mathbb{E}[Z_i] = \Sigma^{-1/2}(\frac{1}{n}\mathbb{E}[x_i x_i^T] + \lambda I)\Sigma^{-1/2} = \Sigma^{-1/2}\Sigma\Sigma^{-1/2} = I$, and that $\lambda_{max}(Z_i) \leq ||\Sigma^{-1}||_2||\frac{1}{n}x_i x_i^T + \lambda I||_2 \leq 1 + \frac{||\mathcal{X}||_2}{n\lambda}$. Now we can bound $||I - \frac{1}{s}\sum_{j\in s} Z_i||_2$ by Theorem 2.2 in Tropp (2010):

Lemma 15. *Tropp (2010) Let $Z_1, \ldots Z_s$ sampled without replacement from $\{Z_1, \ldots Z_n\}$. Then if $Z_i \in \mathbb{S}_d^+, \mathbb{E}[Z_i] = I_d$, and $\max_{i\in[n]} \lambda_{\max}(Z_i) \leq B$ w.p. $1-\rho$, for $\delta = \sqrt{\frac{2B\log(2d/\rho)}{s}}$:*

$$\lambda_{\max}(\frac{1}{s}\sum_{i=1}^{s} Z_i) < 1 + \delta, \;\; \lambda_{\min}(\frac{1}{s}\sum_{i=1}^{s} Z_i) > 1 - \delta$$

So by Lemma 15, we know that with probability $1-\rho$: $|\lambda_{\min}(I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2})| = 1 - \lambda_{\max}(\Sigma^{-1/2}\Sigma_s\Sigma^{-1/2}) \leq \delta$, and similarly $|\lambda_{\max}(I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2})| \leq \delta$, thus $||I - \Sigma^{-1/2}\Sigma_s\Sigma^{-1/2})||_2 \leq \delta$. Substituting this all into Equation 31 and noting $\lambda_{\max}(\Sigma) \leq \frac{||\mathcal{X}||_2}{n} + \lambda$ we get with probability $1-\rho$:

$$||w_{i*}^{\lambda} - w_s||_{\frac{1}{n}X^TX} \leq ||\tilde{w}_s - w_s||_{\Sigma} \leq ||\mathcal{X}||_2|\mathcal{Y}|n||A||_2 \leq ||\mathcal{X}||_2|\mathcal{Y}|\frac{\lambda_{\max}(\Sigma)}{\lambda^2}\delta^2 = \quad (32)$$

$$2||\mathcal{X}||_2|\mathcal{Y}|\frac{(\frac{||\mathcal{X}||_2}{n} + \lambda)^2}{\lambda^3}\frac{\log(2d/\rho)}{s}, \quad (33)$$

which under the assumption $||\mathcal{X}||_2 = O(n\lambda)$ gives $O(\frac{||\mathcal{X}||_2|\mathcal{Y}|\log(2d/\rho)}{\lambda s})$. as desired. ☐

Proof of Lemma 14:

Proof. By Lemma 12, with $A = \frac{1}{s}X_S^TX + \lambda I$, $B = E_1, c = E_2, v = \frac{1}{n}^TY$, we get $w_s - \tilde{w}_s = (\frac{1}{s}X_S^TX + \lambda I + E_1)E_1 w_s - (\frac{1}{s}X_S^TX + \lambda I + E_1)^{-1}E_2$, and so

$$||w_s - \tilde{w}_s||^2_{\frac{X^TX}{n}} \leq 2||(\frac{1}{s}X_S^TX + \lambda I + E_1)^{-1}E_1 w_s||^2_{\frac{X^TX}{n}} + 2||(\frac{1}{s}X_S^TX + \lambda I + E_1)^{-1}E_2||^2_{\frac{X^TX}{n}}$$

Under the assumption $||E_1||_2 \leq \lambda/2$, this becomes

$$||w_s - \tilde{w}_s||^2_{\frac{X^TX}{n}} = O\left(||E_1 w_s||_{(\frac{1}{s}X_S^TX_S+\lambda I)^{-1}(X^TX/n+\lambda I)(\frac{1}{s}X_S^TX_S+\lambda I)^{-1}}\right) + O\left(||E_2||_{(\frac{1}{s}X_S^TX_S+\lambda I)^{-1}(X^TX/n+\lambda I)(\frac{1}{s}X_S^TX_S+\lambda I)^{-1}}\right) \quad (34)$$

Now to apply Lemma 4, we need to bound

$$\mathrm{Tr}((\frac{1}{s}X_S^TX_S + \lambda I)^{-1}(X^TX/n + \lambda I)(\frac{1}{s}X_S^TX_S + \lambda I)^{-1}) \leq d\lambda_{\max}(\Sigma_S)^{-1}(\Sigma)(\Sigma_S)^{-1}) = d\lambda_{\max}(\Sigma^{-1/2}(\Sigma^{1/2}\Sigma_S^{-1}\Sigma^{1/2})^2\Sigma^{-1/2}) \leq d\lambda_{\max}(\Sigma^{-1})\frac{1}{\lambda_{\min}(\Sigma^{-1/2}\Sigma^{1/2}\Sigma_S^{-1}\Sigma^{1/2})}^2 \leq \frac{d}{\lambda}(\frac{1}{1-\delta})^2, \quad (35)$$

where the last inequality follows from Lemma 15. Applying Lemma 4 we get that with probability $1-2\rho$:

$$||w_{i*}^{\lambda} - w_s||_{X^TX/n} = O\left(\sigma_1^2 \cdot \frac{d}{\lambda}(\frac{1}{1-\delta})^2 \cdot ||w_s||_2^2 \log(2d^2/\rho) + \sigma_2^2 \cdot \frac{d}{\lambda}(\frac{1}{1-\delta})^2 \log(2d^2/\rho)\right), \quad (36)$$

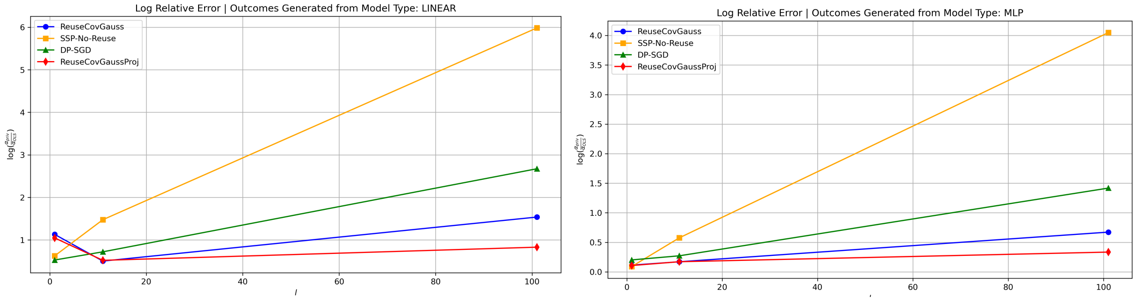
From Lemma 15, $\delta = \sqrt{\frac{2(1+\frac{||\mathcal{X}||_2}{n\lambda})\log(2d/\rho)}{s}}$, which under the assumption $||\mathcal{X}||_2 = O\left(n\lambda\right)$ gives $\frac{1}{(1-\delta)}^2 = O(1 + \frac{\log(2d/\rho)}{s})^2$. Substituting in the value of σ_1, σ_2 gives:

$$\frac{||X||_2^4 n^2}{s^4\epsilon^2}||\mathcal{W}||_2^2 \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s}) \; + \; \frac{l||\mathcal{X}||_2^2||\mathcal{Y}||_2^2}{n^2\epsilon^2} \cdot \frac{d}{\lambda}(1 + \frac{\log(2d/\rho)}{s}),$$

as desired. $\square$

8.6 Additional Experimental Results

Realizability of outcomes has a minimal effect.



(a) (b)

Figure 5: Both (a) and (b) show the log of the ratio of the squared loss of the private estimator to the square loss of the OLS estimator, but in (a) the outcomes are synthetically generated from a linear model, whereas in (b) the outcomes are generated from a 2-layer neural network, to test how the algorithms perform when the outcomes are not generated by a linear model. All values are averaged over 5 iterations, over $l = (1, 11, 101)$ with $(\epsilon, \delta) = (5, \frac{1}{n^2})$ and $n = 5000, d = 25$ on a synthetic dataset with Gaussian features.

DP-SGD outperforms `ReuseCov` for $l < \frac{n}{\sqrt{d}}$. We construct our dataset by sub-sampling $n = 5000$ images of dimension $d = 784$ from MNIST Deng (2012), and generating $l = 1, 11, 101$ outcomes from a noisy linear model with unit norm. We implement DP-SGD using the Opacus Yousefpour et al. (2021) library from Meta, and do privacy accounting across regressions by composing in zCDP and then converting back to $(\epsilon, \delta) - \text{DP}$. Figure 6 shows that `DP-SGD` outperforms our `ReuseCov` algorithms when $l \in (1, 11, 101)$, although, in agreement with the theory, this gap shrinks as l increases.

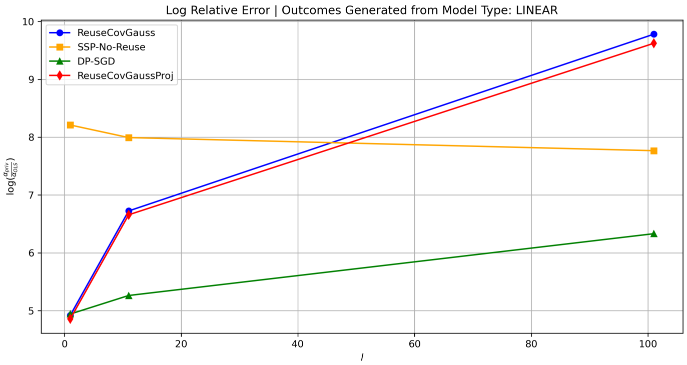


Figure 6: We construct our MNIST dataset by sampling $n = 5000$ random training images ($d = 784$), and then generating synthetic labels from a random linear model. All values are averaged over 5 iterations, over $l = (1, 11, 101)$ with $(\epsilon, \delta) = (5, \frac{1}{n^2})$.