Statistica Sinica Preprint No: SS-2022-0276	
Title	Unbiased Statistical Estimation and Valid Confidence Intervals Under Differential Privacy
Manuscript ID	SS-2022-0276
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0276
Complete List of Authors	Christian Covington, Xi He, James Honaker and Gautam Kamath
Corresponding Authors	Christian Covington
E-mails	ccovington@g.harvard.edu
Notice: Accepted version subject to English editing.	

Unbiased Statistical Estimation and Valid Confidence Intervals Under Differential Privacy

Christian Covington, Xi He, James Honaker, Gautam Kamath

Harvard University, University of Waterloo,

Harvard University & Meta, University of Waterloo

Abstract: We present a method for producing unbiased parameter estimates and valid confidence regions/intervals under the constraints of differential privacy, a formal framework for limiting individual information leakage from sensitive data. Prior work in this area is limited in that it is tailored to calculating confidence intervals for specific statistical procedures, such as mean estimation or simple linear regression. While other recent work can produce confidence intervals for more general sets of procedures, they either yield only approximately unbiased estimates, are designed for one-dimensional outputs, or assume significant user knowledge about the data-generating distribution. Our method induces distributions of mean and covariance estimates via the bag of little bootstraps (BLB) (Kleiner et al., 2014) and uses them to privately estimate the parameters' sampling distribution via a generalized version of the CoinPress estimation algorithm (Biswas et al., 2020). If the user can bound the parameters of the BLB-induced parameters and provide heavier-tailed families, the algorithm produces unbiased parameter estimates and valid confidence intervals which hold

with arbitrarily high probability. These results hold in high dimensions and for any estimation procedure which behaves nicely under the bootstrap.

Key words and phrases: Differential privacy

1. Introduction

1.1 Overview

The dramatic expansion of data collection and analysis has led to growing concerns around the role of privacy and security in the modern world. Our particular focus will be on statistical analysis and how it can leak information about individuals in a data set being analyzed.

Statistical agencies, in particular, have been concerned with *statistical disclosure limitation*, a broad set of techniques used to limit the leakage of sensitive information from statistical analyses. Duncan and Lambert (1986, 1989) and Reiter (2005) work on identifying and quantifying disclosure risk under different assumptions. There has also been substantial work developing various privacy definitions such as k-anonymity (Sweeney, 2002), t-closeness (Li et al., 2007), and l-diversity (Machanavajjhala et al., 2007) among others.

Dinur and Nissim (2003) developed a polynomial data reconstruction algorithm and used it to prove a result later coined in Dwork and Roth

(2014) as the *Fundamental Law of Information Recovery*. Roughly, this law states that an attacker can reconstruct a data set by asking a sufficiently large number of cleverly chosen queries of the data. This inspired the invention of a more formal privacy definition called *differential privacy* (DP) (Dwork et al., 2006). DP is a definition which requires that, for any two data sets differing in one row, the change in the distribution of answers to any possible query between the two data sets is minimal. We loosely define a "DP Algorithm" as a procedure that takes a statistical estimator and converts it into an estimator that satisfies DP (i.e. a "DP Estimator"); we define what it means to "satisfy DP" later.

DP has become a popular tool in many corners of industry but has not been widely applied to research in many fields that often analyze sensitive data (social sciences, medicine, etc.). We suggest that this is, in part, because of a lack of DP algorithms that effectively meet the needs of these fields. First, DP algorithms typically require the user to, *without looking at the data*, specify a domain to which the data will be censored/clipped. This is because privacy mechanisms often take the form of additive noise, with a variance parameter that scales with the sensitivity of the function being privatized (see Definition S3 and Lemma S1 for more details, and note that any theorem, algorithm, section, etc. whose number is preceded by an S

can be found in the online supplement). Many functions of interest, e.g. means and variances, are unbounded (and thus have infinite sensitivity) when defined over unbounded data domains, so we achieve finite sensitivity by bounding the data domain, i.e. clipping/censoring the data.

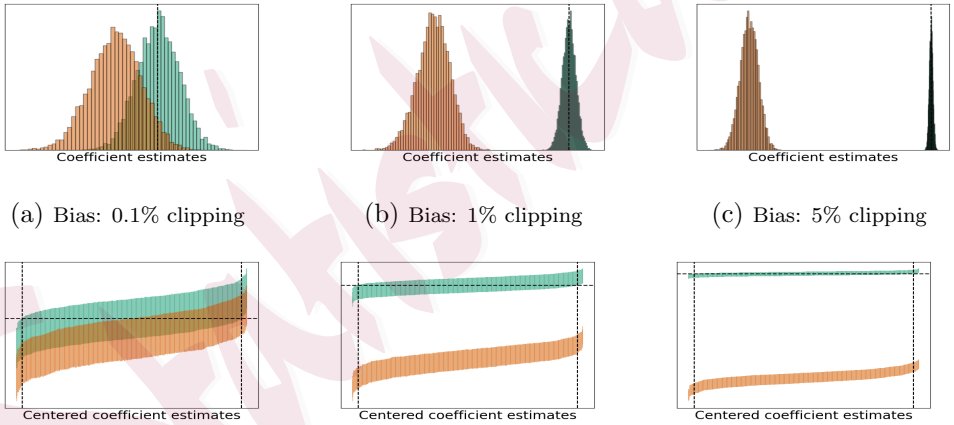
We claim that bounding the data domain effectively is a difficult problem in general, and potentially introduces substantial error into the DP pipeline which is difficult to account for. In particular, without assumptions that the data bounds are set well, DP estimators do not typically yield basic statistical guarantees that many applied researchers desire, namely unbiasedness and valid confidence intervals. Our goal is to provide a framework (i.e. a DP algorithm) for converting non-private estimators to DP estimators in a way that jointly addresses both of these concerns.

1.2 Problem Demonstration

Many DP mechanisms take the form "non-private statistic plus zero-mean noise" (e.g. the *Gaussian mechanism* defined in Lemma S1) and thus look like they ought to be unbiased and yield easily characterizable confidence regions. However, this doesn't take into account the necessary step of choosing a bounded data domain. In practice, analysts are required to specify this domain without looking at the data. The data are then projected into

the domain, and analysis proceeds on the projected data. This introduces a potential trade-off in the analyst's decision calculus. The analyst generally wants to specify a small domain, as this generally requires less noise addition to satisfy DP. However, if the analyst's domain is too small, they risk clipping the data, which can lead to biased estimates.

Consider the case where $X = \{X_1, \ldots, X_n\}$ with $X_i \sim N(0, 1)$ and $y = X\beta + \epsilon$ for $\beta = 100, \epsilon \sim N(0, 10^2)$. We estimate β and get associated 95% confidence intervals using OLS and test the effect of various levels of clipping on the estimates and confidence intervals. Specifically, we leave X unclipped and clip the top $\{0, 0.1, 1, 5\}$ percent of y.



(a) Bias: 0.1% clipping (b) Bias: 1% clipping (c) Bias: 5% clipping

(d) CI coverage: 0.1% clipping (e) CI coverage: 1% clipping (f) CI coverage: 5% clipping

Figure 1: Distribution of OLS coefficient estimates (a-c) and 95% confidence intervals (d-f) under different levels of clipping of y. Non-clipped distribution in green, clipped distribution in orange.

Figure 1 shows results from 10,000 simulations of this process. The top plots show coefficient distributions under each level of clipping and compare it to the condition of no clipping. Note that at even a moderate level of 1% clipping, the distributions of estimates are completely non-overlapping. The bottom plots show the absolute error of the estimates, arranged in increasing order on the x-axis, with vertical bars representing the 95% confidence interval for that estimate. The black dotted vertical lines on the left and right sides of each plot show the 0.025 and 0.975 quantiles, where we expect perfectly calibrated confidence intervals to cross the x-axis. At 0.1% clipping, approximately half of our confidence intervals do not contain the true parameter value; at higher levels of clipping, none of them do.

We chose a very simple regime for the experiments above: one-dimensional OLS with a Gaussian covariate, Gaussian error, clipping in only the outcome variable, and no attempt to satisfy DP (i.e. no noise addition). In more complex settings, the effect of clipping on the coefficients could be larger, in addition to being harder to predict and reason about. Thus, we argue that the data bounding step that precedes so many DP algorithms has potentially immense consequences for practical analysis and ought to be seriously considered.

1.3 Related Work

Differential privacy has grown in popularity in recent years, as has the literature exploring the intersection between statistics and DP. Dwork and Lei (2009) point out how a handful of common robust statistical estimators could be extended to satisfy differential privacy. Wasserman and Zhou (2010) compare DP mechanisms via convergence rates of distributions and densities from DP releases and frame DP in statistical language more broadly. Lei et al. (2016) explores model selection under DP, while Peña and Barrientos (2021) explores model uncertainty. Vu and Slavkovic (2009); Wang et al. (2015); Gaboardi et al. (2016); Canonne et al. (2019), and Awan and Slavkovic (2019) propose methods for DP hypothesis testing in various domains. Sheffet (2017); Wang (2018); Barrientos et al. (2019), and Alabi et al. (2020) all address the problem of differentially private linear regression.

Karwa and Vadhan (2017) gives nearly optimal confidence intervals for univariate Gaussian mean estimation with finite sample guarantees. Du et al. (2020) proposes their own algorithms for the same problem and finds superior practical performance in some domains, and Biswas et al. (2020) develop an algorithm that works well at reasonably small sample sizes and without strong assumptions on user knowledge, while also scaling well to

high dimensions. Drechsler et al. (2021) explores non-parametric confidence intervals for calculating medians. D'Orazio et al. (2015) gives confidence intervals for a difference of means. Avella-Medina et al. (2021) shows how to construct DP version of M-estimators, as well as associated confidence regions.

Our work continues in a line of recent work for constructing confidence intervals for more general classes of differentially private estimators. Brawner and Honaker (2018) shows how to combine estimates from additive functions that satisfy zCDP to get confidence intervals at no additional cost. Wang et al. (2019) provides confidence intervals for models trained with objective or output perturbation algorithms. These algorithms are quite general, but require solving the non-private ERM sub-problem optimally. Ferrando et al. (2021) presents a very general approach based on privately estimating parameters of the data-generating distribution and bootstrapping confidence intervals by repeatedly running the model of interest on samples from a distribution parameterized by the privately estimated parameters. This method is efficient with respect to its use of the privacy budget, but relies on significant knowledge of the structure of the data-generating process. With the exception of Karwa and Vadhan (2017), these works either ignore the issue of bounding the data domain $\mathcal{X}$ effec-

tively, or attempt to address it through bias-correction strategies which we believe are unlikely to work for complex problems. Barrientos et al. (2021) tests a variety of DP algorithms for various tasks (ranging from tabular statistics to OLS regression) on real data and argue that existing methods for performing DP regression, "would struggle to produce accurate regression estimates and confidence intervals" (Barrientos et al. (2021), p. 1).

Evans et al. (2019) is the closest existing work to ours. They also start with the Sample & Aggregate framework, and BLB algorithm to get k estimates of the parameters of the sampling distribution of their non-private estimator. The k estimates are then aggregated via a differentially private mean and confidence intervals are calculated using a differentially private variance estimate and CLT assumption. Because the estimates are projected into a bounded paramter domain to control the sensitivity of the mean, the resulting private mean is not necessarily unbiased. Evans et al. (2019) attempts to address this issue by privately estimating the proportion of the k estimates that are clipped by the projection and adjusting the private mean by the estimated clipping proportions. This method has the advantage of allowing users to specify overly tight clipping bounds in order to decrease the global sensitivity of their estimator, but is sensitive to how well the clipping proportions are estimated and, to our knowledge, has no

means of generalizing to multivariate parameter estimation.

1.4 Contributions

We introduce a general-purpose meta-algorithm that allows an analyst to take any estimator that is (1) unbiased in the non-private setting and (2) has "nice" properties under the bootstrap and produce a version that satisfies DP and, with high probability, is unbiased and produces valid confidence intervals or a valid confidence region. Our results hold under the central (or trusted curator) model of DP.

Our algorithm can be split into three distinct steps, each of which we explain in detail in Section 2. First, we use the Bag of Little Bootstraps (BLB) algorithm (Algorithm S1) developed by Kleiner et al. (2014) to produce estimates of the mean and covariance of the sampling distribution of the non-private estimator.

Second, we privately estimate the mean of each set of BLB estimates (both the means and covariances) using a modified version of the CoinPress private mean estimation algorithm (Algorithm S2 (Biswas et al., 2020)). For both the mean and covariance distributions induced by the BLB, the analyst must provide a distribution that is heavier-tailed (Definition S7 and Assumption 2), as well as give bounds on the mean (Assumption 3) and

covariance (Assumption 4) of the induced distribution. Under these conditions, this step produces parameter estimates which are unbiased and follow a multivariate Gaussian distribution with known covariance (Theorem 5). We argue that these conditions are natural and demonstrate how the properties of the CoinPress algorithm allow the analyst to get good performance even when they set the aforementioned bounds very conservatively.

Although our use of CoinPress in this way guarantees that DP is satisfied with respect to the BLB estimates, this guarantee also holds for the underlying sensitive data over which the BLB estimates were calculated. This fact follows from noting that combining the BLB with an aggregation step that satisfies DP (such as CoinPress) falls under the purview of the Sample & Aggregate framework developed in Nissim et al. (2007).

Third, we combine the estimated parameters using precision weighting to produce final mean and covariance estimates (Theorems 8 and 7), which are used to calculate a valid confidence region/intervals (Theorem 9). We do so via a multivariate extension of the precision-weighting technique to improve CoinPress' estimates (Theorem 6). While precision-weighting is a well-known technique in the meta-analysis literature Cochran (1954), we give, to the best of our knowledge, the first proof of multivariate optimality, which may be of independent interest. This step maintains the

privacy guarantees of the CoinPress algorithm because differential privacy is preserved under postprocessing (Lemma 2).

We believe that our framework is a promising step toward making differential privacy more practical for applied research. The problem of choosing good data bounds is significant in practice in that it is both generally difficult and that many DP algorithms are sensitive to poor choices. There is also currently an asymmetry in the failure modes, in that bounds that are too wide typically yield answers which are unbiased but very noisy, while bounds that are too narrow risk "silent failure", where the DP result looks precise but is not actually representative of the non-private answer. Our framework, through use of the CoinPress algorithm, gives users more leeway to err on the side of conservatism without introducing bias, thus mitigating the possibility of getting DP results that appear precise but are systematically incorrect.

Moreover, our framework is general enough to be applied to any estimator for which the BLB does a "good" job approximating the sampling distribution of the estimator. The bootstrap is broadly familiar to applied statisticians, and its properties for any particular estimator an analyst wishes to use can, in principle, be tested on non-sensitive data. Thus, answering the question of whether or not our algorithm will be useful in a

particular setting does not require significant knowledge of DP.

2. Algorithm Overview

2.1 DP Preliminaries

Throughout this work we use a particular notion of DP called zero-concentrated DP. Suppose we have a data domain $\mathcal{X}$ and data set $X \in \mathcal{X}^n$.

Definition 1 (Zero-concentrated differential privacy (zCDP) (Bun and Steinke, 2016)). Let $\mathcal{M} : \mathcal{X}^n \to \Omega$ be a randomized algorithm where $(\Omega, \Sigma, \mathbb{P})$ is a probability space and $\rho \geq 0$. We say that $\mathcal{M}$ satisfies ρ-zCDP with respect to a data set X if, for all $(X', X^*) \in \mathcal{X}_n$ and $\alpha \in (1, \infty)$: $H_\alpha \left(\mathcal{M}(X') \| \mathcal{M}(X^*)\right) \leq \rho\alpha$, where H_α is the α-Rényi divergence.

The parameter ρ represents an upper bound on the amount of information $\mathcal{M}$ leaks about the underlying data X. Larger ρ implies more information leakage, or *privacy loss*, but also allows for the statistics returned by $\mathcal{M}$ to be more accurate.

zCDP (like other standard notions of DP) has two properties which are very useful for reasoning about how DP guarantees operate within a full data analysis pipeline.

Lemma 1 (Composition of zCDP (Bun and Steinke, 2016)). *Let* $\mathcal{M} : \mathcal{X}^n \to$

$\mathcal{Y}$ and $\mathcal{M}' : \mathcal{X}^n \to \mathcal{Z}$ such that $\mathcal{M}$ satisfies ρ-zCDP and $\mathcal{M}'$ satisfies ρ'-zCDP. Define $\mathcal{M}'' : \mathcal{X}^n \to \mathcal{Y} \times \mathcal{Z}$ by $\mathcal{M}''(x) = (\mathcal{M}(x), \mathcal{M}'(x))$. Then $\mathcal{M}''$ satisfies $(\rho + \rho')$-zCDP.

We use zCDP because its privacy parameters ρ compose additively, which is convenient for algorithms, like CoinPress, that contain multiple private releases. zCDP implies the more familiar notion of (ϵ, δ)-DP (see Proposition 3 of Bun and Steinke (2016)), so any zCDP guarantees in this paper can be converted to (ϵ, δ)-DP if an analyst prefers.

Lemma 2 (Postprocessing of zCDP (Bun and Steinke, 2016)). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ and $f : \mathcal{Y} \to \mathcal{Z}$ such that $\mathcal{M}$ satisfies ρ-zCDP. Define $\mathcal{M}' : \mathcal{X}^n \to \mathcal{Z}$ such that $\mathcal{M}'(x) = f(\mathcal{M}(x))$. Then $\mathcal{M}'$ satisfies ρ-zCDP.*

This postprocessing property of zCDP states that if some output satisfies DP with respect to some data X, functions of that output also satisfy DP with respect to X (provided that the functions do not take X as input).

2.2 Algorithm Step 1: Bag of Little Bootstraps and Sample & Aggregate

In Step 1 of our algorithm, the algorithm takes the analyst's non-private estimator $\hat{\theta}$ and uses the Bag of Little Bootstraps to try to approximate the sampling distribution of $\hat{\theta}$. In particular, the BLB partitions the data

into k disjoint subsets, repeatedly runs the estimator over each subset, and produces k estimates of its mean, $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$, and covariance, $\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]}$.

Let $\mathcal{X}$ be our data domain, $\mathcal{D}$ a distribution over the domain, and $X = \{x_1, \ldots, x_n\}$ be our sensitive data set where x_i are drawn i.i.d. from $\mathcal{D}$. For shorthand, we say that $X \in \mathcal{X}^n$ and $X \sim \mathcal{D}^n$. We say that the analyst wants to run some model, which has an associated parameter vector $\theta \in \mathbb{R}^d$. The analyst specifies the estimator they would have liked to run in the non-private setting $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$. Our goal is eventually to approximate $\hat{\theta}$ in a manner that satisfies DP with respect to X. We assume that n is "public knowledge" and does not need to be privately estimated. If this is not true, we can generate a DP estimate of n, call it $\tilde{n}$, and then X could be subsampled or augmented with rows of synthetic data until it has $\tilde{n}$ rows, creating a new data set $\tilde{X}$ over which we can apply our algorithm.

As stated earlier, DP algorithms typically require specification of the global sensitivity of the function whose outputs are being privatized (see Definition S3 and Lemma S1). This can become arbitrarily complex for complicated models, even after assuming a bounded input domain. The Sample & Aggregate framework introduced in Nissim et al. (2007) provides a strategy for estimating such functions without specifying the global sensitivity. First run the function of interest non-privately over k disjoint subsets

of the data, bound the outputs, and then aggregate the results using a function with a sensitivity that is easier to reason about (for our purposes, we assume the aggregation function is the mean). We then privately estimate the mean of these k results. Because each element in the original data contributes to one subset of the partition, its effect on the aggregation is localized to one of its k inputs, and so a mean estimation algorithm that satisfies DP with respect to those k elements also satisfies DP with respect to the underlying data. A more thorough treatment of this framework can be found in Nissim et al. (2007) and Chapter 7 of Dwork and Roth (2014).

The Bag of Little Bootstraps algorithm (developed in (Kleiner et al., 2014) and reproduced in Section S2) randomly partitions the data X into k disjoint subsets $\{X_1, \ldots, X_k\}$, scales the subset back up to an effective sample size that matches that of the original data (via multinomial sampling), and runs $\hat{\theta}$ r times, producing estimates $\{\hat{\theta}_{i,a}\}_{a\in[r]}$. It then aggregates these into an arbitrary assessment of estimator quality. We use the mean and covariance, so for each $i \in [k]$ we get $\hat{\theta}_i^{BLB} = \frac{1}{r}\sum_{a=1}^{r}\hat{\theta}_{i,a}$ and $\hat{\Sigma}_i^{BLB} = \text{Cov}\left(\{\hat{\theta}_{i,a}\}_{a\in[r]}\right)$. Our approach allows us to find a single confidence region for the entire parameter vector jointly. However, we acknowledge that many analysts will prefer separate confidence intervals for each element in their parameter vector. This preference is advantageous from a privacy perspec-

tive, as confidence intervals will require less noise to privatize than a full confidence region and will thus be more accurate.

We present methods and results for the general case of finding a confidence region, but also show how this could be adapted to instead find confidence intervals. We include the setup for finding confidence intervals in the main text with results in the supplement.

These sets of estimates are now empirical approximations to the theoretical distributions of the BLB estimates, which we assume are themselves good approximations of the actual parameters of interest. We make this last assumption explicit as follows.

Assumption 1. *Let the estimator* $\hat{\theta} \sim G(\theta, \Sigma)$ *where the marginals of* G *each belong to a location-scale family. For* $\hat{\theta}_i^{BLB}$ *and* $\hat{\Sigma}_i^{BLB}$ *generated by applying BLB to our estimator* $\hat{\theta}$*, let* $\hat{\theta}^{BLB} = \frac{1}{k}\sum_{i=1}^{k} \hat{\theta}_i^{BLB}$ *and* $\hat{\Sigma}^{BLB} = \frac{1}{k}\sum_{i=1}^{k} \hat{\Sigma}_i^{BLB}$*. We assume that* $\mathbb{E}\left(\hat{\theta}^{BLB}\right) = \mathbb{E}\left(\hat{\theta}\right) = \theta$ *and* $\mathbb{P}\left(\hat{\Sigma}^{BLB} \succeq \hat{\Sigma}\right) = 1$.

We take $\succeq$ to mean "greater/equal in Löwner order". In particular, $A \succeq B \iff A - B$ is a PSD matrix, in which case we call A a Löwner upper bound on B.

2.3 Algorithm Step 2: Generating Private Parameter Estimates With CoinPress

Now that we have BLB estimates $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$ and $\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]}$, we can privately estimate their empirical means $\hat{\theta}^{BLB}$ and $\hat{\Sigma}^{BLB}$ using the CoinPress mean estimation algorithm (Section S3.1). We try to provide intuition here for how and why CoinPress works, but interested readers should consult Biswas et al. (2020) for a more complete treatment.

For generality, we say that we have data $\{y_i\}_{i\in[k]}$ with empirical mean $\hat{\mu}$, where each y_i is an i.i.d. instantiation of a random variable Y with mean μ_Y and covariance Σ_Y. Our goal is to estimate μ_Y in a manner that satisfies DP. In order to state these results generally, we will assume that $Y \in \mathbb{R}^{d'}$ for some d'. When y_i stands in for $\hat{\theta}_i^{BLB}$, we have $d' = d$, the dimension of our parameter vector of interest. The same is true when y_i stands in for $\hat{\Sigma}_i^{BLB}$ and we care only about univariate confidence intervals such that $\hat{\Sigma}_i^{BLB} = \hat{V}_i^{BLB} I_d$ for some $\hat{V}_i^{BLB}$. When we care about a joint confidence region, we need to use the entire upper triangular of $\hat{\Sigma}_i^{BLB}$, so $d' = \frac{d(d+1)}{2}$. In the actual implementation (and proofs), these covariance estimates are flattened into a vector of the appropriate dimension before estimation and unflattened at the end of estimation to produce full covariance matrices again. We assume that this is going on behind the scenes and, for notational

clarity, keep calling them $\hat{\Sigma}_i^{BLB}$.

The high-level idea behind CoinPress is to make a series of mean estimates, where each estimate (probabilistically) improves upon the last, making only a few requirements of the analyst.

Assumption 2. *The analyst provides a location-scale family of distributions $Q_Y(\mu, \Sigma)$ with heavier tails than the distribution of Y as described in Definition S7.*

Assumption 3. *The analyst provides $\tilde{\mu}_0 \in \mathbb{R}^d$ and $r_0 \in \mathbb{R}$ such that $\mu_Y \in B_2(\tilde{\mu}_0, r_0)$, the ℓ_2 ball centered at $\tilde{\mu}_0$ with radius r_0.*

Assumption 4. *The analyst provides $\Sigma_Y^U \in \mathbb{R}^{d\times d}$ such that $\Sigma_Y \preceq \Sigma_Y^U$.*

These three assumptions provide the backbone of the iterative improvement in CoinPress. Recall that the noise added to privatize an estimator defined over an input domain generally scales with the size of the domain. Over a series of t steps, CoinPress attempts to find a small domain that, with high probability, contains all the $\{y_i\}_{i\in[k]}$. Assumption 3 ensures that the algorithm starts with sufficiently conservative bounds that contain μ_X. Assumptions 2 and 4 then allow the algorithm to convert bounds on μ_X to high-probability bounds on the individual data points y_i. For the remainder of the section, we assume that our three assumptions hold.

2.3 Algorithm Step 2: Generating Private Parameter Estimates With CoinPress

At step $m \in [t]$ of the algorithm, CoinPress takes the ℓ_2 ball and expands it outward based on the assumed distribution. If you have a distribution of a known family with bounded mean and covariance, you can, with high probability, upper bound the ℓ_2 norm of an arbitrary number of draws from said distribution. In the context of our case, it specifies a ball that, with high probability, contains all of the $\{y_i\}_{i\in[k]}$. In this case, using the ball as our data domain and projecting our data into this domain will not clip any of the data. The global sensitivity of the mean is calculated using this ball and CoinPress adds noise scaled to the sensitivity using the Gaussian mechanism (Lemma S1), which adds zero-mean noise from a multivariate Gaussian with diagonal covariance to get a private estimate $\tilde{\mu}_m$. Because we are, with high probability, not clipping any of the $\{y_i\}_{i\in[k]}$, our use of the Gaussian mechanism implies that the form of our private estimator is "empirical mean + zero-mean noise". Thus, based on the scale of the noise, we can produce a new ℓ_2 ball which, with high probability, contains the true empirical mean. This ball becomes the bounding set for the mean at the next step and the process continues. The guarantee that CoinPress does not clip any points ensures (with high probability) that for all $m \in [t]$, the $\tilde{\mu}_m$ are unbiased estimates of the empirical mean of the y_i.

Moreover, assuming no clipping, the variances (induced by clipping

and our use of the Gaussian mechanism) of $\tilde{\mu}_m$ are just the diagonal of the covariance parameter used in the Gaussian mechanism, which we call $\vec{\sigma}^2_{\tilde{\mu},m} \in \mathbb{R}^{d'}_+$. In other words, under no clipping, the additional error incurred by the general privacy mechanism is simply the error from the Gaussian mechanism.

In general, applying the CoinPress algorithm to a data set $\{y_i\}_{i\in[k]}$ satisfies ρ-zCDP with respect to $\{y_i\}_{i\in[k]}$ (Biswas et al., 2020). In our case $\{y_i\}_{i\in[k]}$ stands in for either of the sets of BLB estimates $\{\hat{\theta}_i^{BLB}\}_{i\in[k]}$ and $\{\hat{\Sigma}_i^{BLB}\}_{i\in[k]}$. So, CoinPress satisfies zCDP with respect to each set of BLB estimates and, by the extension implied by our use of the Sample & Aggregate framework, also satisfies zCDP with respect to the original data X. Moreover, CoinPress comes with a high-probability guarantee on the form of the private estimates it produces.

Theorem 5. *Under Assumptions 2, 3, and 4, CoinPress produces t mean estimates $\{\tilde{\mu}_m\}_{m\in[t]}$ and associated privacy variances $\{\vec{\sigma}^2_{\tilde{\mu},m}\}_{m\in[t]}$ such that*

$$\mathbb{P}\left[\forall m \in [t] : \tilde{\mu}_m \sim N\left(\hat{\mu}, \vec{\sigma}^2_{\tilde{\mu},m} I_{d'}\right)\right] \geq 1 - \beta^{\tilde{\mu}}.$$

We now recall that $\hat{\mu}$ is the empirical mean of the $\{y_i\}_{i\in[k]}$, where y_i stands in for either $\hat{\theta}_i^{BLB}$ or $\hat{\Sigma}_i^{BLB}$. That is, we use CoinPress to privately estimate the mean of the distributions induced by the BLB. This yields t es-

timates of both the parameter means and covariances (both with associated privacy variances) $\{\tilde{\theta}_m, \vec{\sigma}^2_{\tilde{\theta},m}\}_{m\in[t]}$ and $\{\tilde{\Sigma}_m, \vec{\sigma}^2_{\tilde{\Sigma},m}\}_{m\in[t]}$.

2.4 Algorithm Step 3: Get Final Parameter Estimates and Confidence Intervals Via Postprocessing

We now use our t DP mean and covariance estimates $\{\tilde{\theta}_m, \tilde{\Sigma}_m\}_{m\in[t]}$ to get a single DP estimate of the mean and covariance, which we'll call $\tilde{\theta}$ and $\tilde{\Sigma}$.

Final Parameter Estimates We start by presenting a multivariate version of the precision weighting argument which gives the minimal covariance way to combine a set of unbiased estimators.

Theorem 6. *For a parameter τ, say we are given a series of independent estimates $\{\hat{\tau}_m\}_{m\in[t]}$ such that $\mathbb{E}(\hat{\tau}_m) = \tau$ and $Cov(\hat{\tau}_m) = S_m$ for some positive definite S_m. Then the minimum covariance unbiased linear weighting of the $\{\hat{\tau}_m\}_{m\in[t]}$ is given by $\hat{\tau} = (\Sigma_{m=1}^t S_m^{-1})^{-1} (\Sigma_{m=1}^t S_m^{-1}\hat{\tau}_m)$, which has $\mathbb{E}(\hat{\tau}) = \tau$ and $Cov(\hat{\tau}) = (\Sigma_{m=1}^t S_m^{-1})^{-1}$.*

Specifically, by "minimum covariance" we mean that any other unbiased linear weighting $\hat{\tau}'$ of the $\{\hat{\tau}_m\}_{m\in[t]}$ will have $\mathrm{Cov}(\hat{\tau}) \preceq \mathrm{Cov}(\hat{\tau}')$.

Recall from Theorem 5 that, with high probability, both our $\{\tilde{\theta}_m\}_{m\in[t]}$ and $\{\tilde{\Sigma}_m\}_{m\in[t]}$ are sets of estimates which are independent, unbiased, and

have a known covariance structure. Thus, they both meet the criteria to be combined into precision-weighted estimators in the style of Theorem 6. Because the precision weighting step is a function of the DP estimates, it retains the privacy guarantees from Step 2 by the postprocessing property of zCDP (Lemma 2).

We start by using the precision-weighting idea to find an estimator for the covariances, $\tilde{\Sigma}$. Unlike the mean estimation setting, where our goal is to produce an unbiased private estimator, our goal for our private covariance estimator is to find a private estimator that will reliably overestimate the empirical covariance, and thus yield valid confidence intervals. To get a sense for why this is necessary, consider the one-dimensional case where we have a sample variance s and a privatized version $\tilde{s}$. In the non-private setting, we simply use the estimate s to calculate confidence intervals, but in order to use $\tilde{s}$ for confidence intervals we need to understand the relationship between s and $\tilde{s}$. An important first step is to ensure that there is no clipping in the construction of $\tilde{s}$, so we know that $\tilde{s}$ equals s plus zero-mean noise; this is the same motivation as in the mean estimation case. An extra complication for variances is that zero-mean noise addition has an asymmetric impact on confidence interval coverage; if we were to use $\tilde{s}$ for confidence intervals, we would underestimate the true variance 50% of

the time. We can avoid this problem by increasing $\tilde{s}$ to the point that we know, with high probability, that it is at least as large as s.

Theorem 7. *Given covariance estimates and privacy variances* $\{\tilde{\Sigma}_m, \vec{\sigma}^2_{\Sigma,m}\}_{m\in[t]}$, *let* $\tilde{S}_m \in \mathbb{R}^{d'}$ *be the flattened version of* $\tilde{\Sigma}_m$. *We can construct a precision-weighted estimator* $\tilde{S}$: $\tilde{S} := \frac{\sum_{m=1}^t \tilde{S}_m/\vec{\sigma}^2_{\Sigma,m}}{\sum_{m=1}^t 1/\vec{\sigma}^2_{\Sigma,m}}$.

Let $\tilde{\Sigma}'$ *be the unflattened* $d \times d$ *version of* $\tilde{S}$ *and* b *be the unflattened* $d \times d$ *matrix where* $b^2_{ij} = Var\left(\tilde{\Sigma}'_{ij}\right)$ *(i.e. the diagonal values of the covariance matrix of the flattened precision-weighted estimator). For* $\beta^{ub} \in (0,1)$, *define*

$$c = \min_{\epsilon\in(0,1/2]} (1+\epsilon)\left(2\max_{i\in[d]}\|b_{\cdot,j}\|_2 + \frac{6\sqrt{\log d}}{\log(1+\epsilon)}\max_{i,j\in[d]\times[d]}|b_{ij}|\right) + \sqrt{\frac{\ln(1/\beta^{ub})}{4\max_{ij} b^2_{ij}}}.$$

Then, for $\tilde{\Sigma} = \tilde{\Sigma}' + cI_d$ *we have* $\mathbb{P}\left(\hat{\Sigma} \preceq \tilde{\Sigma}\right) \geq 1 - \beta^{\tilde{\Sigma}} - \beta^{ub}$.

We use a similar strategy to convert our private parameter estimates $\{\tilde{\theta}_m\}_{m\in[t]}$ into a final private parameter estimate $\tilde{\theta}$. Because we want an unbiased estimator of $\hat{\theta}$, we don't need to be conservative like we did with the covariance; we simply use the precision-weighted estimator.

Theorem 8. *Given parameter estimates and privacy variances* $\{\tilde{\theta}_m, \vec{\sigma}^2_{\theta,m}\}_{m\in[t]}$, *we define the precision-weighted estimator* $\tilde{\theta}$ *as* $\tilde{\theta} := \frac{\sum_{m=1}^t \tilde{\theta}_m/\vec{\sigma}^2_{\theta,m}}{\sum_{m=1}^t 1/\vec{\sigma}^2_{\theta,m}}$. *This estimator has expectation* $\hat{\theta}$ *and covariance* $\Sigma_{\tilde{\theta}} = \frac{1}{\sum_{m=1}^t 1/\vec{\sigma}^2_{\theta,m}} I_d$. *In particular, we say that* $\mathbb{P}\left(\tilde{\theta} \sim N\left(\hat{\theta}, \Sigma_{\tilde{\theta}}\right)\right) \geq 1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$.

Confidence Region We now have private estimates $\tilde{\theta}$ and $\tilde{\Sigma}$ such that $\tilde{\theta} \sim N\left(\hat{\theta}, \Sigma_{\tilde{\theta}}\right)$ and $\hat{\Sigma} \preceq \tilde{\Sigma}$ with probability $1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$. Going back to Assumption 2, we also have a distribution $Q_{\hat{\theta}}$ we assume to be heavier tailed than that of $\hat{\theta}$. We can represent our approximation of the sampling distribution of our estimator as the compound distribution $Q_{\hat{\theta}}\left(\hat{\theta} + N\left(0, \Sigma_{\tilde{\theta}}\right), \tilde{\Sigma}\right)$.

Theorem 9 (Confidence Region (valid with high probability))**.** *Let Z be a d-dimensional random variable such that $Z \sim Q_{\hat{\theta}}\left(\hat{\theta} + N\left(0, \Sigma_{\tilde{\theta}}\right), \tilde{\Sigma}\right)$. Suppose C is a d-dimensional ellipsoid such that $\mathbb{P}(Z \in C) \geq 1 - \alpha$ for some $\alpha \in (0, 1)$. Then, with probability $1 - \beta^{\tilde{\Sigma}} - \beta^{ub} - \beta^{\tilde{\theta}}$: $\mathbb{P}\left(\theta \in C\right) \geq 1 - \alpha$.*

It is always trivial to find such an ellipsoid C (e.g. take $C = \mathbb{R}^d$), but finding one with coverage close to $1 - \alpha$ analytically could be difficult in general. However, in typical scenarios it is likely to be much more nicely behaved. Most notable is the case where $Q_{\hat{\theta}}$ is multivariate Gaussian (e.g. the analyst is comfortable assuming that the CLT has kicked in for the BLB estimates). The resulting compound distribution is $N\left(\tilde{\theta}, \tilde{\Sigma} + \Sigma_{\tilde{\theta}}\right)$; a Gaussian random variable with a Gaussian random variable as its location parameter is still Gaussian. This becomes even simpler if the analyst is interested in univariate confidence intervals, in which case they can use the fact that $\forall j \in [d] : \tilde{\theta}_j \sim N\left(\hat{\theta}_j, \tilde{\Sigma}_{j,j} + \Sigma_{\tilde{\theta}_{j,j}}\right)$, and can calculate confidence

intervals directly from the CDF of the univariate Gaussian. In more complicated scenarios, an analyst can always get approximate quantiles using Monte Carlo methods.

2.5 Full Algorithm Statement

In Algorithm 1, we finally present our algorithm in whole. We omit some hyperparameters in the subroutines to make it easier to focus on the core pieces that change between them. Let $\xi : a_{1:k} \mapsto \frac{1}{k}\sum_{i=1}^{k} a_i$ and either $\xi' : a_{1:k} \mapsto \text{Cov}\left(\{a_i\}_{i\in[k]}\right)$ or $\xi' : a_{1:k} \mapsto \text{diag}\left(\text{Cov}\left(\{a_i\}_{i\in[k]}\right)\right)$, depending on whether the analyst desires a joint confidence region or separate confidence intervals.

In summary, our private estimates $\tilde{\theta}$ and $\tilde{\Sigma}$ have statistical guarantees relative to the true parameters of the sampling distribution of $\hat{\theta} \sim G(\theta, \Sigma)$ via the following lines of reasoning:

$$\mathbb{E}\left(\tilde{\theta}\right) \underset{\text{CoinPress + postprocessing}}{\overset{w.h.p.}{=}} \mathbb{E}\left(\hat{\theta}^{BLB}\right) \underset{\text{Assumption 1}}{=} \mathbb{E}\left(\hat{\theta}\right) = \theta$$

$$\tilde{\Sigma} \underset{\text{CoinPress + postprocessing}}{\overset{w.h.p.}{\succeq}} \hat{\Sigma}^{BLB} \underset{\text{Assumption 1}}{\succeq} \Sigma$$

3. Empirical Evaluation

We provide empirical demonstrations of our core result, showing that we can produce unbiased parameter estimates and valid confidence intervals

Algorithm 1 General Valid DP (GVDP)

Input: data set $X \in \mathbb{R}^{n \times m}$, estimator $\hat{\theta} : \mathcal{X}^n \to \mathbb{R}^d$ families of distributions $Q_{\hat{\theta}}, Q_{\hat{\Sigma}}$, privacy budgets $\rho^{\tilde{\theta}}, \rho^{\tilde{\Sigma}} > 0$, failure probabilities $\beta^{\tilde{\theta}}, \beta^{\tilde{\Sigma}}, \beta^{ub} \in (0, 1)$

Output: parameter estimate $\tilde{\theta}$ and associated confidence intervals/region C which satisfy $\left(\rho^{\tilde{\theta}} + \rho^{\tilde{\Sigma}}\right)$-zCDP and have desired unbiased/validity properties with probability $1 - \beta^{\tilde{\theta}} - \beta^{\tilde{\Sigma}} - \beta^{ub}$

procedure GVDP($X, \hat{\theta}, Q_{\hat{\theta}}, Q_{\hat{\Sigma}}, \rho^{\tilde{\theta}}, \rho^{\tilde{\Sigma}}$)

  $\{\hat{\Sigma}_i^{BLB}\}_{i \in [k]} = \text{BLB}\left(X, \hat{\theta}, \xi', \ldots\right)$ ▷ Algorithm S1 – get BLB estimates of parameter covariance

  $\{\hat{\theta}_i^{BLB}\}_{i \in [k]} = \text{BLB}\left(X, \hat{\theta}, \xi, \ldots\right)$ ▷ Algorithm S1 – get BLB estimates of parameter means

  $\{\tilde{\Sigma}_m\}_{m \in [t]} = \text{MVMREC}\left(\{\hat{\Sigma}_i^{BLB}\}_{i \in [k]}, \ldots, Q_{\hat{\Sigma}}, \ldots, \rho^{\tilde{\Sigma}}, \beta^{\tilde{\Sigma}}\right)$ ▷ Algorithm S2 – privately estimate parameter covariance at $\rho^{\tilde{\Sigma}}$-zCDP level

  Combine $\{\tilde{\Sigma}_m\}_{m \in [t]}$ via precision-weighting to get $\tilde{\Sigma}$ ▷ Theorem 6

  $\{\tilde{\theta}_m\}_{m \in [t]} = \text{MVMREC}\left(\{\hat{\theta}_i^{BLB}\}_{i \in [k]}, \ldots, Q_{\hat{\theta}}, \ldots, \rho^{\tilde{\theta}}, \beta^{\tilde{\theta}}\right)$ ▷ Algorithm S2 – privately estimate parameter means at $\rho^{\tilde{\theta}}$-zCDP level

  Combine $\{\tilde{\theta}_m\}_{m \in [t]}$ via precision-weighting to get $\tilde{\theta}$ ▷ Theorem 6

  Use $\tilde{\theta}, \tilde{\Sigma}, Q_{\hat{\theta}}$, and β^{ub} to get confidence intervals/region C. ▷ Theorem 9

  return $\{\tilde{\theta}, C\}$

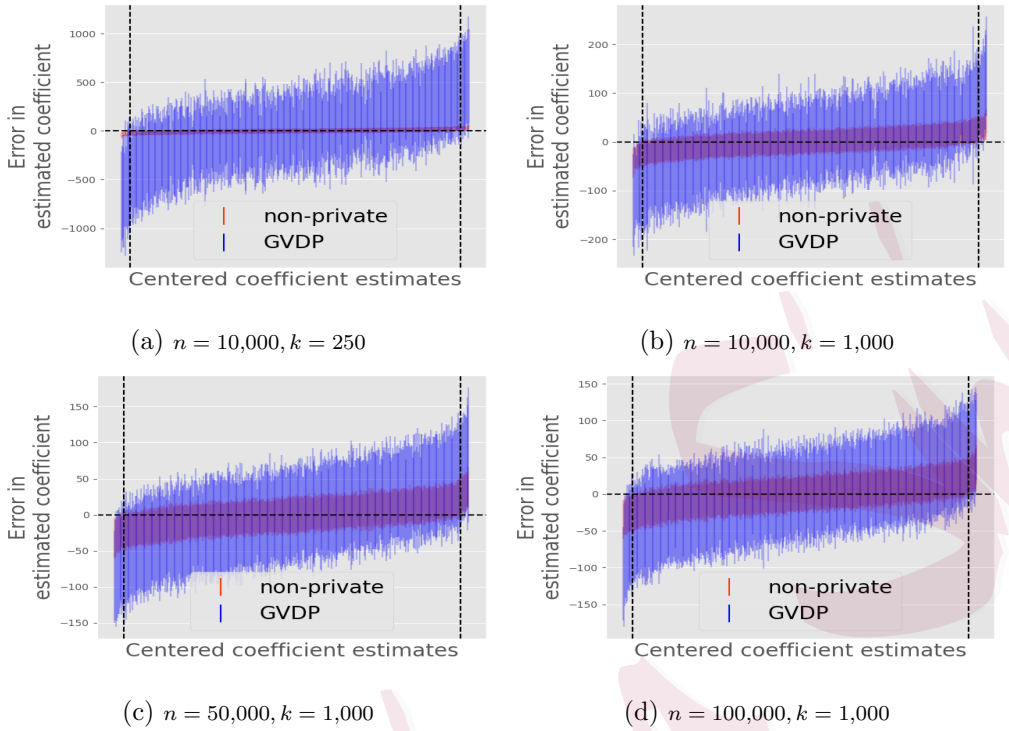
when the requisite assumptions hold. For every evaluation, we aim to get valid confidence intervals for each element of the parameter vector rather than a single valid confidence region, as we expect this to be the dominant use case in practice.

All results satisfy zCDP at the $\rho = 0.1$ level and, inside the GVDP algorithm, we always run CoinPress for $t = 5$ iterations. Additionally, we assume that the analyst chooses bounds that satisfy Assumptions 3 and 4, but are larger than the tightest possible bounds by a factor of ≈ 100. For

example, if the analyst had an estimator with a $N(\mu = 1, \sigma^2 = 1)$ sampling distribution we assume their prior knowledge to be that $\mu \in [-100, 100]$ and $\sigma^2 \leq 100$. Additional empirical results, including comparisons to existing methods, are described in Section S6.

OLS Regression Demonstration We begin by testing parameter estimation for a properly specified OLS model with $d = 5$ parameters of interest. In a single iteration of our experiment, we generate data from a linear model $y = X\beta + \epsilon$ with Gaussian covariates, Gaussian error, and correlation structure such that the effective rank of the resulting data is $\approx d - 1$. We increase the underlying noise in the data as n increases such that the non-private confidence intervals are essentially constant across values of n. This allows us to better demonstrate the effects of changes in k and n on our algorithm's performance. We then privately estimate the values of the d coefficients and their associated standard errors. We run this entire experiment 100 times. We assume the sampling distribution of the coefficients is multivariate Gaussian and imagine that the user sets all upper bounds ≈ 100 times larger than the tightest possible upper bounds. We present these results in Figure 2.

Each plot consists of coefficient estimates centered around their true



(a) $n = 10{,}000, k = 250$ (b) $n = 10{,}000, k = 1{,}000$

(c) $n = 50{,}000, k = 1{,}000$ (d) $n = 100{,}000, k = 1{,}000$

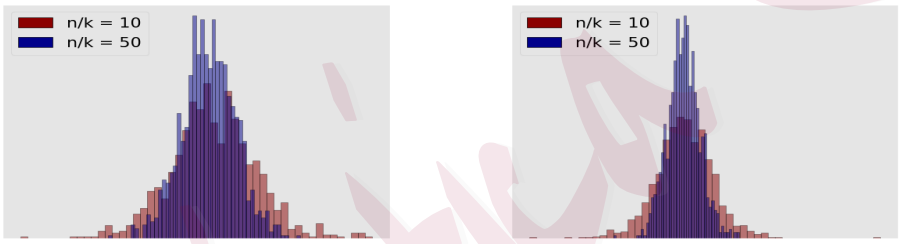
Figure 2: OLS: Distribution of coefficient estimates and 95% confidence intervals.

values and presented in increasing order, with vertical bars representing the 95% confidence interval for that estimate. We expect properly calibrated confidence intervals to cross the x-axis at the vertical dotted black lines, placed at the 2.5^{th} and 97.5^{th} quantiles, which is the behavior we observe in each plot.

Additionally, Figure 2 demonstrates the principle that the noise due to privacy in our algorithm scales with k rather than n. The private confidence intervals are significantly tighter in plot (b) than in plot (a), while plots (c)

and (d) are essentially identical.

One complicating factor to this story is that plot (c) looks better than plot (b), even though they use the same k. This is because the variance of the BLB estimates will tend to decrease as $\frac{n}{k}$ increases, up to the point where $\frac{n}{k}$ is large enough that the BLB estimates have converged to the sampling distribution of the estimator. Figure 3 shows the distribution of the BLB estimates of two of our estimated coefficients at the different levels of $\frac{n}{k}$ used in plots (b) and (c).



(a) Coefficient 1 (b) Coefficient 3

Figure 3: OLS: BLB estimates from a single run

Ideally, the analyst should attempt to choose k to be as large as possible, subject to the constraint that the BLB estimates, calculated on subsets of size $\frac{n}{k}$, converge to the true sampling distribution of the estimator.

One possible strategy for doing this is for the analyst to test the BLB estimation for their estimator of interest on non-sensitive data and use a

k that performs well on the non-sensitive data. In particular, an analyst could start with $k = 1$ and increase k until the distribution of BLB estimates starts to look substantially different (i.e. different beyond Monte Carlo error). Of course, this approach works only if the analyst can find or generate non-sensitive data that are similar enough to their sensitive data that the BLB will perform similarly on each. We discuss the issue further in Section S7.4.

3.1 Comparison with UnbiasedPrivacy (Evans et al., 2019)

In Table 1, we compare our algorithm to the UnbiasedPrivacy (UP) algorithm from Evans et al. (2019). UP is designed for estimators with a univariate Gaussian sampling distribution, so we focus on that setting here.

While the goal of GVDP is to let users set very conservative bounds (which the algorithm improves upon) and not clip any points in the aggregation step, UP requires that users set reasonably good bounds upfront. UP never tightens bounds that are too loose, but it will attempt to bias-correct the results if the analyst's chosen bounds clip some of the data (which our algorithm does not).

We test UP against GVDP across five scenarios using the implementation provided in the original paper. As suggested in Evans et al. (2019), we

split the privacy budget evenly between the mean estimation task and the estimation of the proportion of points that are clipped. We consider two cases in which we expect UP to perform bias correction, clipping the top 20% and 10% of the BLB estimates, and three in which we don't; clipping bounds set as tightly as possible with no clipping, bounds set three times larger than the true values of the parameter, and bounds set 1,000 times larger than the true value.

We run 1,000 simulations, each of which involves generating $n = 50{,}000$ data points $Y_i \sim N(0, 250)$ and using UP and GVDP to generate a private OLS estimator, using $k = 500$ as the number of subsets for the BLB for each method. In Table 1, we provide the average ℓ_1 error in the coefficient estimate, average standard error, and empirical 95% confidence interval coverage for each method with various levels of clipping bound.

In the 10% clipping and tight bound (no clipping) settings, UP mostly delivers as advertised; it gives (approximately) valid confidence intervals and does so with smaller standard errors than GVDP does under any bounds. However, it is unable to achieve this when the top 20% of BLB estimates are clipped, yielding highly biased coefficient estimates and poor CI coverage. Moreover, even in the 10% and tight bound settings, UP does not appear to achieve truly unbiased coefficient estimates. We believe

Bounds	Method	Avg Coef Err	Avg SE	CI Cov
Top 20% Clipped	UP	0.222	0.0872	0.237
	GVDP	0.007	0.207	0.970
Top 10% Clipped	UP	0.080	0.108	0.932
	GVDP	0.003	0.207	0.970
Tightest bounds (no clipping)	UP	0.129	0.145	0.973
	GVDP	0.001	0.208	0.975
3 times too large	UP	0.004	0.370	0.968
	GVDP	0.004	0.218	0.973
1,000 times too large	UP	3.239	121.090	0.950
	GVDP	0.005	0.701	0.961

Table 1: Comparison of UP and GVDP: Average Coefficient Estimate, Average Standard Error, and Empirical Coverage of 95% Confidence Intervals

this is because of the error in the bias correction step of UP, created when privately estimating the proportion of clipped data points.

In the cases where the bounds are too conservative, we see that GVDP outperforms UP, as GVDP is designed to improve conservative bounds whereas UP is not. It's notable that in the 10% and 20% clipping where GVDP gives no guarantees, it appears to provide unbiased coefficient estimates and valid CI coverage. This is because of GVDP's variance estimates are conservative by design (to ensure valid coverage with high probability), so even when the initial data bounds are set too narrowly it is possible that GVDP's overly conservative variances compensate inside of CoinPress such that no BLB estimates end up being clipped.

4. Discussion

We believe that whether or not our method (GVDP) is effective relative to other approaches will generally come down to a few different factors.

First, we suggest that GVDP be considered primarily when the analyst is not confident in their ability to set "good" bounds on their underlying data domain $\mathcal{X}$ for their given estimator. This is a function of both analyst knowledge of $\mathcal{X}$ and the properties of their estimator, as some estimators will be robust even if the analyst sets bounds that clip small proportions of the data, while others will not be (see our demonstration in Section 1.2).

Second, GVDP is likely to work well only for reasonably large n. Recall that GVDP partitions the data X into k subsets of size $\frac{n}{k}$ and bootstrapping the estimator over each subset. This creates a tradeoff between the plausibility of our assumptions and the required noise addition to satisfy DP. As k increases, the sensitivity of our aggregator decreases and so too does the variance of the noise in our privacy mechanism.

However, we require that the BLB estimates be good estimates of the sampling distribution of our estimator in the non-private setting (Assumption 1). Similar to other bootstrap methods, the BLB's

guarantees are asymptotic (see Section 3 of Kleiner et al. (2014)), and so it is difficult to know how reasonable Assumption 1 is when $\frac{n}{k}$ is small.

Finally, we believe GVDP can be useful in settings where the dimension of the estimand of interest is significantly lower than the dimension of the data. Higher dimensional domains create two potential problems for differentially private estimation that are not present in non-private estimation. First, setting clipping bounds, generally speaking, gets more difficult as the dimension increases. Second, the function sensitivity, and thus variance of the noise in our privacy mechanism, increases with the dimension of the function's input domain (again see Lemma S1). The function being privatized in GVDP takes the BLB parameter estimates as input rather than the full data, so these two problems are minimized if the estimand is low-dimensional relative to the original data.

Acknowledgements

CC was supported by an NSERC Discovery Grant, a Graduate Excellence Award in Computer Science, a Dr. Derick Wood Graduate Scholarship, and a David R. Cheriton Graduate Scholarship. XH was supported by an NSERC Discovery Grant. GK was supported by an NSERC Discovery Grant and a University of Waterloo startup grant.

References

Alabi, D., A. McMillan, J. Sarathy, A. Smith, and S. Vadhan (2020). Differentially private simple linear regression. *arXiv preprint arXiv:2007.05157*.

Avella-Medina, M., C. Bradshaw, and P.-L. Loh (2021). Differentially private inference via noisy optimization.

Awan, J. and A. Slavkovic (2019). Differentially private inference for binomial data.

Barrientos, A. F., J. P. Reiter, A. Machanavajjhala, and Y. Chen (2019). Differentially private significance tests for regression coefficients. *Journal of Computational and Graphical Statistics 28*(2), 440–453.

Barrientos, A. F., A. R. Williams, J. Snoke, and C. M. Bowen (2021). A feasibility study of differentially private summary statistics and regression analyses for administrative tax data.

Biswas, S., Y. Dong, G. Kamath, and J. Ullman (2020). Coinpress: Practical private mean and covariance estimation. *Advances in Neural Information Processing Systems 33*.

Brawner, T. and J. Honaker (2018). Bootstrap inference and differential privacy: Standard errors for free.

Bun, M. and T. Steinke (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer.

Canonne, C. L., G. Kamath, A. McMillan, A. Smith, and J. Ullman (2019). The structure

of optimal private tests for simple hypotheses. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 310–321.

Cochran, W. G. (1954). The combination of estimates from different experiments. *Biometrics 10*(1), 101–129.

Dinur, I. and K. Nissim (2003). Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, New York, NY, USA, pp. 202–210. Association for Computing Machinery.

D'Orazio, V., J. Honaker, and G. King (2015). Differential privacy for social science inference. *Sloan Foundation Economics Research Paper* (2676160).

Drechsler, J., I. Globus-Harris, A. McMillan, J. Sarathy, and A. D. Smith (2021). Non-parametric differentially private confidence intervals for the median. *CoRR abs/2106.10333*.

Du, W., C. Foot, M. Moniot, A. Bray, and A. Groce (2020). Differentially private confidence intervals.

Duncan, G. and D. Lambert (1989). The risk of disclosure for microdata. *Journal of Business & Economic Statistics 7*(2), 207–217.

Duncan, G. T. and D. Lambert (1986). Disclosure-limited data dissemination. *Journal of the American Statistical Association 81*(393), 10–18.

REFERENCES

Dwork, C. and J. Lei (2009). Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 371–380.

Dwork, C., F. McSherry, K. Nissim, and A. Smith (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer.

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

Evans, G., G. King, M. Schwenzfeier, and A. Thakurta (2019). Statistically valid inferences from privacy protected data.

Ferrando, C., S. Wang, and D. Sheldon (2021). Parametric bootstrap for differentially private confidence intervals.

Gaboardi, M., H. Lim, R. Rogers, and S. Vadhan (2016). Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *International conference on machine learning*, pp. 2111–2120. PMLR.

Karwa, V. and S. Vadhan (2017). Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*.

Kleiner, A., A. Talwalkar, P. Sarkar, and M. I. Jordan (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, 795–816.

Lei, J., A.-S. Charest, A. Slavkovic, A. Smith, and S. Fienberg (2016). Differentially private model selection with penalized and constrained likelihood. *arXiv preprint*

arXiv:1607.04204.

Li, N., T. Li, and S. Venkatasubramanian (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115. IEEE.

Machanavajjhala, A., D. Kifer, J. Gehrke, and M. Venkitasubramaniam (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD) 1*(1), 3–es.

Nissim, K., S. Raskhodnikova, and A. Smith (2007). Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 75–84.

Peña, V. and A. F. Barrientos (2021). Differentially private methods for managing model uncertainty in linear regression models. *arXiv preprint arXiv:2109.03949*.

Reiter, J. P. (2005). Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association 100*(472), 1103–1112.

Sheffet, O. (2017, 06–11 Aug). Differentially private ordinary least squares. In D. Precup and Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, Volume 70 of *Proceedings of Machine Learning Research*, pp. 3105–3114. PMLR.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*(05), 557–570.

REFERENCES

Vu, D. and A. Slavkovic (2009). Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 138–143. IEEE.

Wang, Y., D. Kifer, and J. Lee (2019, Mar.). Differentially private confidence intervals for empirical risk minimization. *Journal of Privacy and Confidentiality 9*(1).

Wang, Y., J. Lee, and D. Kifer (2015). Revisiting differentially private hypothesis tests for categorical data. *arXiv preprint arXiv:1511.03376*.

Wang, Y.-X. (2018). Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain.

Wasserman, L. and S. Zhou (2010). A statistical framework for differential privacy. *Journal of the American Statistical Association 105*(489), 375–389.

Christian Covington, Harvard University

E-mail: ccovington@g.harvard.edu

Xi He, University of Waterloo

E-mail: xihe@uwaterloo.ca

James Honaker, Meta & Harvard University

E-mail: james@hona.kr

Gautam Kamath, University of Waterloo

E-mail: gautam.c.kamath@uwaterloo.ca