Differentially Private Linear Regression and Synthetic Data Generation with Statistical Guarantees

Shurong Lin **Aleksandra Slavković** **Deekshith Reddy Bhoomireddy**
The Pennsylvania State University

Abstract

In the social sciences, small- to medium-scale datasets are common, and linear regression is canonical. In privacy-aware settings, much work has focused on differentially private (DP) linear regression, but mostly on point estimation with limited attention to uncertainty quantification. Meanwhile, synthetic data generation (SDG) is increasingly important for reproducibility studies, yet current DP linear regression methods do not readily support it. Mainstream DP-SDG approaches either are tailored to discrete or discretized data, making them less suitable for analyses involving continuous variables, or rely on deep learning models that require large datasets, limiting their use for the smaller-scale data typical in social science. We propose a method for linear regression with valid inference under Gaussian DP. It includes a bias-corrected estimator with asymptotic confidence intervals (CIs) and a general SDG procedure such that the corresponding regression on the synthetic data matches our DP linear regression procedure. Our approach is effective in small- to moderate-dimensional settings. Experiments show that our method (1) improves accuracy over existing methods for DP linear regression, (2) provides valid CIs, and (3) produces more reliable synthetic data for downstream statistical and machine learning tasks than current DP synthesizers.

1 INTRODUCTION

In the social, economic, and behavioral sciences, where small- to medium-scale datasets are common, linear regression (LR) and subsequent statistical inference are widely used to address important scientific questions. Data from these contexts readily contain sensitive information. The confidentiality protection methodology for sharing data and the results of statistical analyses has a long history drawing from many fields (e.g., Hundepool et al. (2012); Slavković and Seeman (2023)). The modern methods predominantly rely on differential privacy (DP) (Dwork et al., 2006) to ensure rigorous privacy guarantees. Numerous methods for fitting LR under DP have been proposed, including general approaches such as objective perturbation (Kifer et al., 2012; Zhang et al., 2012) and DP (stochastic) gradient descent (DP-SGD) (Bassily et al., 2014; Abadi et al., 2016), as well as LR-specific techniques like sufficient statistics perturbation (SSP) and its variants (Dwork et al., 2014; Wang, 2018). However, most methods focus on point estimation and provide statistical risk bounds but with limited support for uncertainty quantification. Valid statistical inference in LR settings under DP remains a challenge due to inadequate accounting for the noise added to satisfy the privacy guarantee. Sheffet (2017) derived theoretical inference results for the SSP and Johnson–Lindenstrauss Transform (JLT) mechanisms, but some of these rely on the strong assumption of a Gaussian design matrix. Moreover, the SSP and JLT approaches exhibit substantially larger error than our more flexible method and other baselines (see Figure 1), and therefore offer limited practical utility.

Meanwhile, reproducibility and replicability are important concepts in trustworthy social science research (National Academies of Sciences, Engineering, and Medicine, 2019; Webb et al., 2026). Researchers often want to conduct replication studies to verify or build upon prior analyses. In privacy-aware settings, however, typical DP methods only return model estimates, preventing others from revisiting or extending the analysis without access to the original data. Synthetic data generation (SDG) offers a possible solution. The basic idea was proposed in the early 1990s, but its broad adoption is still lacking (van Kesteren, 2024). Furthermore, most methods for generating synthetic data under DP either

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

rely on large datasets and complex models, such as deep learning–based approaches (Jordon et al., 2018; Xie et al., 2018; Xin et al., 2020, 2022), or discretize continuous variables and produce discretized synthetic data, thereby sacrificing continuity (McKenna et al., 2022; Zhang et al., 2021; Cai, Lei, Wei and Xiao, 2021). These limitations restrict their applicability in small- to medium-scale settings, particularly when preserving continuity is essential. Moreover, the statistical implications of conducting downstream analyses, including LR, on such DP synthetic data remain largely unexplored, especially with respect to inferential validity.

To address these challenges, we propose a novel unified method for LR and SDG under Gaussian DP that provides valid statistical inference through an effective and practical binning-aggregation strategy. We use an existing DP binning method as a preprocessing step to obtain a DP partition of the covariate domain for aggregation. Unlike approaches that rely on discretizing continuous variables, this step does not force the final synthetic data to lie on a discrete support or replace continuous variables by categorical values. The novelty lies in the binning-aggregation framework: by aggregating covariates and responses within bins, we reformulate LR as a weighted model, which supports valid statistical inference and provides a general procedure for SDG under DP. To our knowledge, this is the first work to reformulate DP-LR in this way, offering both inference guarantees and the ability to generate synthetic data within the same framework.

Main Contributions. (1) We propose a method for LR that satisfies Gaussian DP and mostly achieves the lowest estimation error among existing DP-LR algorithms, particularly on real datasets; see Algorithm 2 and Theorem 1. Our method requires minimal tuning and runs significantly faster than computationally intensive approaches. (2) We develop a DP statistical inference procedure based on the central limit theorem (CLT), analogous to the classical non-private regression, without requiring any assumptions on the covariate distribution. A CLT result for DP-LR has been missing from the literature, and our work provides the first such statement; see Theorem 2. (3) We introduce a SDG mechanism that provides a general procedure beyond LR and supports replication studies at no additional privacy cost; see Algorithm 3 and Theorem 1.

1.1 Related Work

Existing DP-LR methods include objective function perturbation (Kifer et al., 2012; Zhang et al., 2012), DP stochastic gradient descent (DP-SGD) (Bassily et al., 2014; Abadi et al., 2016; Cai, Wang and Zhang, 2021), and one posterior sampling (OPS) (Dimitrakakis et al., 2014; Minami et al., 2016). These approaches are general-purpose, but typically require careful hyperparameter tuning. Wang (2018) proposed modified versions of sufficient statistics perturbation (SSP) (Dwork et al., 2014; Sheffet, 2017) and OPS, namely AdaSSP and AdaOPS, respectively, by introducing adaptive regularization.

A few more recent works of interest have a relatively narrow focus. Alabi et al. (2022) proposed a DP method exclusively for simple linear regression that outputs predictions only at $x = 0.25$ and 0.75, assuming x and y are within $[0, 1]$. Varshney et al. (2022) proposed theory for a one-pass mini-batch SGD method for sub-Gaussian data via adaptive clipping, while Milionis et al. (2022) focused on LR with unbounded covariates but assumed the covariates are Gaussian; neither work includes numerical evaluations. Amin et al. (2023) proposed a bound-free method relying on a Propose-Test-Release check, which often fails when n is small. Their evaluations focused on datasets with $n \gtrsim 1000 \cdot d$, making the method unsuitable for the smaller datasets we are targeting. Dick et al. (2023) proposed a method to improve prediction accuracy through covariate selection rather than estimating the regression coefficient in the specified model.

While most of these methods provide statistical risk bounds, statistical inference has received much less attention. Unlike the non-private setting, where inference is well-established, DP-LR lacks broadly applicable methods for uncertainty quantification. Although empirical approaches such as bootstrapping can be used to construct confidence intervals, they often require generating many estimates, which in turn necessitates splitting the privacy budget across multiple runs, or incur additional privacy costs for estimating regression errors (Ferrando et al., 2022). Overall, analytical solutions for valid LR inference under DP remain limited. Sheffet (2017) studied inference for SSP and JLT mechanisms, but some results rely on Gaussian design assumptions. Lin et al. (2024) derived approximate variance formulas for DP-GD and SSP in the context of linked data, where LR appears as a special case.

2 PRELIMINARIES

In this section, we review some preliminaries on DP and an existing DP binning algorithm that is used as a preprocessing step in our method.

2.1 Differential Privacy

Two concepts central to DP are *neighboring relations* and *sensitivity*. Let $\mathcal{X}$ be some data space, and $D, D' \in \mathcal{X}^{\mathbb{N}}$ be two *neighboring datasets*, where one is obtained from the other by adding or removing a single record. This relation is denoted by $D \sim D'$. We refer to

it as *remove-one/add-one* neighboring relation. The sensitivity of a function is defined as follows.

Definition 1 (Sensitivity). *Consider the problem of privately releasing a statistic $\theta(D)$ of the dataset D. The sensitivity of θ is defined as*

$$\text{sens}(\theta) = \sup_{D \sim D'} |\theta(D) - \theta(D')|.$$

Definition 2 ((ε, δ)-DP, Dwork and Roth (2014)). *Let $\varepsilon > 0, \delta > 0$. An algorithm A is (ε, δ)-differentially private, if for every pair of neighboring datasets $D \sim D'$, and any possible output set S,*

$$\mathbb{P}(A(D) \in S) \leq e^{\varepsilon} \cdot \mathbb{P}(A(D') \in S) + \delta. \tag{1}$$

This notion is referred to as *approximate DP*. When $\delta = 0$, this notion is commonly denoted ε-DP and is called *pure DP* (Dwork et al., 2006).

In our work, we adopt Gaussian DP, a variant with a better statistical interpretation and a tighter composition property. Consider the hypothesis testing:

H_0 : the distribution is P vs. H_1 : the distribution is Q.

Let α_ϕ and β_ϕ denote Type I and II errors for rejection rule ϕ. Let $T(P, Q)(\alpha) := \inf_\phi\{\beta_\phi : \alpha_\phi \leq \alpha\}$.

Definition 3 (Gaussian DP, Dong et al. (2022)). *Let $\mu > 0$. An algorithm A is μ-Gaussian differentially private (μ-GDP), if for every neighboring $D \sim D'$ and any $\alpha \in (0, 1)$*

$$T(A(D), A(D'))(\alpha) \geq T(\mathcal{N}(0, 1), \mathcal{N}(\mu, 1))(\alpha).$$

In other words, testing "H_0: the underlying dataset is D" versus "H_1: the underlying dataset is D'" is at least as hard as testing "H_0: the distribution is $\mathcal{N}(0, 1)$" versus "H_1: the distribution is $\mathcal{N}(\mu, 1)$."

Gaussian DP and approximate DP are precisely mutually convertible, and pure DP implies Gaussian DP. We use the following conversion properties in our work.

Proposition 2.1 (Conversion). *Let $\Phi(\cdot)$ be the standard Gaussian cumulative distribution function.*

(i) (Corollary 2.13, Dong et al. (2022)) A mechanism is μ-GDP if and only if it is $(\varepsilon, \delta(\varepsilon))$-DP for all $\varepsilon > 0$, where $\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon}{\mu} + \frac{\mu}{2}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon}{\mu} - \frac{\mu}{2}\right)$.

(ii) (Theorem 5.1, Liu et al. (2022)) Any ε-DP algorithm is also μ-GDP for $\mu = -2\,\Phi^{-1}\left(\frac{1}{1+e^{\varepsilon}}\right) \leq \sqrt{\frac{\pi}{2}}$.

Like the classic notion of pure and approximate DP, Gaussian DP has the following fundamental properties and the Gaussian Mechanism (Dong et al., 2022).

Proposition 2.2 (Composition). *The n-fold composition of μ_i-GDP mechanisms is*

$$\sqrt{\mu_1^2 + \mu_2^2 + \cdots + \mu_n^2}.$$

Proposition 2.3 (Post-processing). *If an algorithm A is μ-GDP, then any post-processing function f, i.e., $f \circ A$, is also μ-GDP.*

Proposition 2.4 (Gaussian mechanism). *Define the Gaussian mechanism applied to a statistic θ on dataset D by*

$$A(D) = \theta(D) + \xi,$$

where $\xi \sim \mathcal{N}(0, \text{sens}(\theta)^2/\mu^2)$. Then $A(\cdot)$ satisfies μ-GDP.

2.2 A DP Binning Algorithm

The proposed method involves creating bins. If the binning strategy is data-independent, the bin boundaries are public and do not reveal sensitive information. For instance, one can set fixed bin widths before accessing sensitive data. However, data-independent strategies often lack adaptability and are prone to the curse of dimensionality. To address this, one may opt for a data-dependent binning method, such as recursive partitioning based on counts, which produces a more refined and representative histogram. This approach, however, incurs additional privacy cost, as the binning must itself be performed in a DP manner. In our work, we use the PrivTree algorithm of Zhang et al. (2016) to output private bins without counts.

The PrivTree algorithm builds a hierarchical, tree-structured partitioning of the data domain by recursively splitting nodes based on privately perturbed, down-biased counts. For a node v at depth $d = \text{depth}(v)$ (root has depth 0) with count $c(v)$, a penalized score is defined by subtracting a fixed amount τ per level so deeper nodes need stronger evidence to split:

$$b(v) = \max\{\, c(v) - d\,\tau,\ \theta - \tau \,\}.$$

Add Laplace noise to protect privacy,

$$\hat{b}(v) = b(v) + \text{Laplace noise},$$

and node v is split if $\hat{b}(v) > \theta$. A full algorithm and exact noise calibration can be found in Appendix C. The choice of τ is determined by the desired privacy level, while θ is a tunable hyperparameter. As discussed in Zhang et al. (2016), the negative bias helps ensure that setting the threshold to $\theta = 0$ typically results in sufficiently large point counts in each node.

In our implementation, the root node corresponds to the initial bin. We pass in a non-sensitive d-dimensional region represented as the Cartesian product $\Pi_{i=1}^{d}(L_i, U_i)$, where L_i and U_i denote the lower

and upper bounds of the i-th covariate, respectively. Each node is recursively split along its widest dimension. We use the resulting leaf nodes (final bins) in our algorithm design in Section 3. Since PrivTree satisfies ϵ-DP and our method adopts μ-GDP, we leverage Proposition 2.1 (ii) to convert the privacy guarantees.

3 METHODOLOGY

We present *BinAgg*, a framework with three algorithms: (1) the fundamental binning–aggregation step, (2) DP linear regression, and (3) DP synthetic data generation. A corresponding Python package is available on GitHub.

3.1 Aggregated Linear Model

Let X denote the $n \times d$ matrix of covariates and $\boldsymbol{y}$ be the n-dimensional response vector. The classic linear model is given by

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{e}, \quad \boldsymbol{e} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n). \tag{2}$$

We propose an alternative formulation given the observations of X are partitioned into K bins, represented as $\{(\mathcal{B}_k, c_k)\}_{k=1}^K$, where $\mathcal{B}_k$ denotes the kth bin and c_k is the number of observations in that bin. Let j be the index over data points. We aggregate the observations in each bin by defining

$$\boldsymbol{s}_k = \sum_{\boldsymbol{x}_j \in \mathcal{B}_k} \boldsymbol{x}_j, \quad t_k = \sum_{\boldsymbol{x}_j \in \mathcal{B}_k} y_j, \quad \eta_k = \sum_{\boldsymbol{x}_j \in \mathcal{B}_k} e_j.$$

In matrix form, we let $S = (\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \ldots, \boldsymbol{s}_K^\top)^\top$, $\boldsymbol{t} = (t_1, t_2, \ldots, t_K)^\top$, $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_K)^\top$. This aggregation leads to a weighted linear model:

$$\boldsymbol{t} = S\boldsymbol{\beta} + \boldsymbol{\eta}, \tag{3}$$

where $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 C)$ with $C = \mathrm{diag}(c_1, \ldots, c_K)$. Let $W = C^{-1}$. An unbiased and consistent estimator of $\boldsymbol{\beta}$ is given by the weighted least squares (WLS) estimator:

$$\hat{\boldsymbol{\beta}} = (S^\top W S)^{-1} S^\top W \boldsymbol{t}. \tag{4}$$

Remark 3.1. *Model (3) is equivalent to the averaged (weighted) model defined as* $\bar{\boldsymbol{y}} = \bar{X}\boldsymbol{\beta} + \bar{\boldsymbol{e}}$, *where* $\bar{X} \overset{def}{=} C^{-1}S$, $\bar{\boldsymbol{y}} \overset{def}{=} C^{-1}\boldsymbol{t}$, *and* $\bar{\boldsymbol{e}} \overset{def}{=} C^{-1}\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 C^{-1})$. *This model yields the same WLS estimator* $\hat{\boldsymbol{\beta}} = (\bar{X}^\top C \bar{X})^{-1} \bar{X}^\top C \bar{\boldsymbol{y}} = (S^\top W S)^{-1} S^\top W \boldsymbol{t}$ *as in (4). In this work, we proceed with Model (3) for DP algorithm design.*

Our approach that ensures DP for the original data $(X, \boldsymbol{y})$ consists of two major steps: (1) apply a DP algorithm to determine the bin structure $\{\mathcal{B}_k\}_{k=1}^K$, with PrivTree being one option, and (2) add noise to the aggregated statistics, specifically to $\boldsymbol{t}$, S, and the count matrix C. Then a DP aggregated model is

$$\widetilde{\boldsymbol{t}} = \widetilde{S}\boldsymbol{\beta} + \tilde{\boldsymbol{\eta}}, \tag{5}$$

where $\tilde{\boldsymbol{\eta}} \sim \mathcal{N}(0, \sigma^2 \widetilde{C})$. Let $\widetilde{W} = \widetilde{C}^{-1}$. A naive estimator is given by

$$\tilde{\boldsymbol{\beta}}_{\text{naive}} = (\widetilde{S}^\top \widetilde{W} \widetilde{S})^{-1} \widetilde{S}^\top \widetilde{W} \widetilde{\boldsymbol{t}}.$$

However, it does not account for the extra uncertainty introduced by injected noise. Instead, we propose a debiased estimator (see Theorem 2):

$$\tilde{\boldsymbol{\beta}} = (\widetilde{S}^\top \widetilde{W} \widetilde{S} - \widetilde{D})^{-1} \widetilde{S}^\top \widetilde{W} \widetilde{\boldsymbol{t}},$$

where $\widetilde{D}$ is a private bias-correction matrix, as specified in Algorithm 2.

Although the aggregated model involves K bins as the effective sample size rather than n individual observations, this does not necessarily imply a substantial loss of statistical efficiency. After accounting for the weights, the variance of each bin-level summary shrinks with its count c_k. In other words, aggregation reduces the effective size but simultaneously reduces variability of the effective random error. Moreover, adaptive DP partitioning (e.g., PrivTree) mitigates extreme or highly unbalanced binning structure by allocating finer partitions in dense regions and coarser ones elsewhere, helping preserve utility in practice.

3.2 DP Binning-Aggregation Algorithms

Motivated by Section 3.1, we propose the binning–aggregation (BinAgg) framework, as captured by Algorithm 1. The novelty of BinAgg lies in converting raw data into a set of DP bin-level summaries sufficient for both LR and SDG. Given a DP partition of the covariate space, Algorithm 1 aggregates records within each bin to form counts and per-bin sums of covariates and responses. It releases privatized bins and counts under task-specific privacy budgets, while the per-bin sums of covariates and responses are privatized in later algorithms. The partition requires a prespecified, public domain $\mathcal{X}$ for X, provided by the analyst (e.g., based on survey design or known variable ranges). If such bounds are not naturally available or are too conservative, they may be obtained via standard clipping based on domain knowledge or privately estimated using a small portion of the privacy budget. The same assumption applies to the response variable y, for which a public bound is also specified. By injecting privacy noise to bin-level sums, rather than to per-record quantities or sufficient statistics, the framework (i) preserves the joint $(X, \boldsymbol{y})$ structure and (ii) reduces effective sensitivity: each coordinate is confined to its

bin range, and bins with small (privatized) counts can be discarded. The contribution is a novel coupling of DP binning with aggregation to yield a weighted linear model for valid inference and a general SDG mechanism simultaneously.

Algorithm 1 DP BinAgg Preparation

Input: Dataset $(X, \boldsymbol{y})$, domain for X, privacy budgets for binning and counts: μ_{bin} and μ_c.

1: Create a list of μ_{bin}-GDP bins for X (e.g., via PrivTree) : $\{\mathcal{B}_k\}_{k=1}^K$ where $\mathcal{B}_k = \prod_{i=1}^d (L_{ki}, U_{ki})$
2: For each bin $\mathcal{B}_k$, compute: $c_k = \sum_{\boldsymbol{x}_j \in \mathcal{B}_k} 1$
3: **for** $k = 1$ to K **do**
4: Privatize count:

$$\tilde{c}_k = \text{round}(c_k + \xi^c), \quad \xi^c \sim \mathcal{N}(0, 1/\mu_c^2)$$

5: **if** $\tilde{c}_k < 2$ **then**
6: Discard bin $\mathcal{B}_k$.
7: **end if**
8: **end for**
9: Reset K to be the number of bins after discarding.
10: For each bin $\mathcal{B}_k$, compute:

$$\boldsymbol{s}_k = \sum_{\boldsymbol{x}_j \in \mathcal{B}_k} \boldsymbol{x}_j, \quad t_k = \sum_{x_j \in \mathcal{B}_k} y_j$$

11: Compute sensitivity vector $\boldsymbol{\Delta}_k = (\Delta_{k1}, \ldots, \Delta_{kd})^\top$, where $\Delta_{ki} = \max(|L_{ki}|, |U_{ki}|)$

Output: Privatized bins and counts $\{(\mathcal{B}_k, \tilde{c}_k)\}_{k=1}^K$; bin-wise aggregates $\{(\boldsymbol{s}_k, t_k)\}_{k=1}^K$ (to be privatized) with sensitivity vectors $\{\boldsymbol{\Delta}_k\}_{k=1}^K$ for $\boldsymbol{s}_k$.

Privacy Model. Throughout this paper, we adopt the unbounded notion of DP, under which neighboring datasets differ by adding or removing a single record (i.e., the remove-one/add-one relation), whereas bounded DP uses the replace-one neighboring relation. The unbounded notion is common in the DP synthetic data literature and is often preferred when either definition is acceptable, since it yields lower sensitivity than bounded DP under the same privacy parameters and therefore requires less noise (McKenna et al., 2022). That said, the two notions are qualitatively different, as they are based on different neighboring relations, so the same privacy parameters do not have the same interpretation. Our framework can also be adapted to bounded DP by recalibrating the sensitivities and the corresponding Gaussian noise under the replace-one neighboring relation.

Sensitivity of $\boldsymbol{x}$ and y. Under the remove-one/add-one neighboring relation, a differing record affects only one bin. Let i index the covariate dimensions in the kth bin, denoted by $\mathcal{B}_k = \prod_{i=1}^d (L_{ki}, U_{ki})$. For the per-bin sum of covariates, the coordinate-wise sensitivity is $\Delta_{ki} = \max\{|L_{ki}|, |U_{ki}|\}$, and for the per-bin sum of responses it is given by the bound on the response variable, denoted by B_y. We collect $\boldsymbol{\Delta}_k = (\Delta_{k1}, \ldots, \Delta_{kd})^\top$ to calibrate the Gaussian mechanisms in Algorithms 2 and 3.

Algorithm 2 DP BinAgg for Linear Regression

Input: Output from Algorithm 1, response variable bound B_y, privacy budgets for covariates and response: μ_s and μ_t

1: **for** $k = 1$ to K **do**
2: Privatize $\boldsymbol{s}_k$, and t_k:

$$\tilde{\boldsymbol{s}}_k = \boldsymbol{s}_k + \boldsymbol{\xi}^s, \quad \boldsymbol{\xi}^s \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Delta}_k^2/\mu_s^2),$$
$$\tilde{t}_k = t_k + \xi^y, \quad \xi^y \sim \mathcal{N}(0, B_y^2/\mu_t^2).$$

3: **end for**
4: Let

$$\widetilde{W} = \text{diag}(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_K), \quad \tilde{w}_k = 1/\tilde{c}_k$$
$$\widetilde{S} = (\tilde{\boldsymbol{s}}_1^\top, \tilde{\boldsymbol{s}}_2^\top, \ldots, \tilde{\boldsymbol{s}}_K^\top)^\top$$
$$\widetilde{\boldsymbol{t}} = (\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_K)^\top,$$

5: Calculate matrix $\widetilde{D} = \frac{1}{K} \sum_{k=1}^K \tilde{w}_k D_k$ where $D_k = \text{diag}(\boldsymbol{\Delta}_k^2/\mu_s^2)$.

Output: Private estimator

$$\tilde{\boldsymbol{\beta}} = (\widetilde{S}^\top \widetilde{W} \widetilde{S} - \widetilde{D})^{-1} \widetilde{S}^\top \widetilde{W} \widetilde{\boldsymbol{t}}$$

and a DP-CI for each coordinate j,

$$\tilde{\beta}_j \pm z_{\alpha/2} \, \text{se}(\tilde{\beta}_j),$$

where $\text{se}(\tilde{\beta}_j)$ is defined in Section 4.2 and $z_{\alpha/2}$ is the $(1 - \alpha/2)$ quantile of $\mathcal{N}(0, 1)$.

This BinAgg framework instantiates two procedures: (1) LR in Algorithm 2, which implements the weighted model induced by bin-level aggregation and provides valid CIs as established in Theorem 2; and (2) SDG in Algorithm 3, which generates samples directly from the same bin-level summaries to support reproducibility and broader downstream analyses. These procedures target different use cases: Algorithm 2 performs DP-LR and provides DP-CIs, whereas Algorithm 3 produces a reusable DP synthetic dataset supporting residual analysis, visualization, and fitting alternative models. Algorithm 3 is more general and subsumes Algorithm 2, since applying the weighted/aggregated linear model to the synthetic data yields an estimator that matches Algorithm 2 in distribution (see Corollary 1). This consistency is essential for reproducible scientific research. It allows for synthetic data sharing and valid LR inference without discrepancies or additional privacy cost. We remark that applying the usual unweighted lin-

ear regression directly to the synthetic data does not yield the debiased estimator or the associated CIs provided by our method. Practitioners who want both synthetic data and the debiased estimator may embed Algorithm 2 within Algorithm 3; we provide this option in our software package.

Algorithm 3 DP BinAgg for Synthetic Data

Input: Output from Algorithm 1, response variable bound B_y, privacy budgets for covariates and response: μ_s and μ_t.

1: **for** $k = 1$ to K **do**
2: **for** $i = 1$ to $\tilde{c}_k$ **do**
3: Sample synthetic covariates:

$$\tilde{\boldsymbol{x}}^{(k,i)} = \frac{\boldsymbol{s}_k + \boldsymbol{\xi}^x}{\tilde{c}_k}, \quad \boldsymbol{\xi}^x \sim \mathcal{N}(\mathbf{0}, \tilde{c}_k \boldsymbol{\Delta}_k^2/\mu_s^2),$$

4: Sample synthetic response:

$$\tilde{y}^{(k,i)} = \frac{t_k + \xi^y}{\tilde{c}_k}, \quad \xi^y \sim \mathcal{N}(0, \tilde{c}_k B_y^2/\mu_t^2).$$

5: **end for**
6: **end for**

Output: A DP synthetic dataset:

$$\mathcal{D}_{\text{syn}} = \left\{ (\tilde{\boldsymbol{x}}^{(k,i)}, \tilde{y}^{(k,i)}) \mid k = 1, \ldots, K;\ i = 1, \ldots, \tilde{c}_k \right\}.$$

Corollary 1 (Equivalence of Two Algorithms)**.** *In Algorithm 3, aggregate the synthetic data points in each bin by letting* $\tilde{\boldsymbol{s}}'_k = \sum_{i=1}^{\tilde{c}_k} \tilde{\boldsymbol{x}}^{(k,i)}$ *and* $\tilde{t}'_k = \sum_{i=1}^{\tilde{c}_k} \tilde{y}^{(k,i)}$. *Then,*

$$\tilde{\boldsymbol{s}}'_k \overset{d}{=} \tilde{\boldsymbol{s}}_k \sim \mathcal{N}(\boldsymbol{s}_k, \boldsymbol{\Delta}_k^2/\mu_s^2), \quad \tilde{t}'_k \overset{d}{=} \tilde{t}_k \sim \mathcal{N}(t_k, B_y^2/\mu_t^2),$$

where $\tilde{\boldsymbol{s}}_k$ *and* $\tilde{t}_k$ *are defined in Algorithm 2.*

A natural alternative for SDG is to post-process a private regression fit (e.g., $\boldsymbol{y} = X\boldsymbol{\beta}^{\text{priv}}$ or $\boldsymbol{y} = X\boldsymbol{\beta}^{\text{priv}} + \boldsymbol{e}$), but this either collapses variability onto a hyperplane or requires a private estimate of the residual variance, which demands extra privacy budget and sensitivity analysis. In contrast, Algorithm 3 generates samples by using the DP bin-level summaries, thereby preserving variability while remaining consistent with the regression model, without additional privacy cost.

Remark on Synthetic Sample Size. Under unbounded DP, neighboring datasets differ in size by one record, so the sample size could be protected. Accordingly, Algorithm 3 does not require the synthetic dataset to have exactly the same size as the original data. Instead, its size is determined by the privatized counts, namely, $\tilde{n} = \sum_{k=1}^{K} \tilde{c}_k$, after discarding bins with $\tilde{c}_k < 2$. Thus, the released synthetic data do not reveal the original sample size n. In our implementation, the threshold $\tilde{c}_k < 2$ typically reduces the synthetic sample size only slightly.

If a synthetic dataset of exact size m is desired (for example, $m = n$ when n is treated as public information), one may post-process the noisy counts by rescaling $\tilde{c}_k \leftarrow \tilde{c}_k \cdot \frac{m}{\tilde{n}}$ and applying an integer apportionment rule such as the largest remainder (Hamilton) method or randomized rounding to obtain integer counts summing to m, with no additional privacy cost. We provide this option in our software package.

Remark on Binning Dependence. While BinAgg requires a binned structure (i.e., the bin boundaries) as its foundation, it is not inherently tied to the use of any particular binning algorithm. This flexibility allows practitioners to tailor the binning procedure to their data characteristics and privacy constraints, while still retaining the theoretical guarantees of our method. We adopt PrivTree in our implementation because it is a practical, data-dependent binning method that adapts to the data density and has been successfully combined with DP-SDG in prior evaluation studies (Tao et al., 2022). However, other binning approaches, either data-independent (e.g., uniform partitioning) or alternative data-dependent methods, can also be used in conjunction with BinAgg, provided that the bin boundaries are released in a DP manner. Importantly, during the binning step, BinAgg allocates privacy budget only for constructing the bin structure, not for any additional statistics that some binning algorithms may output.

4 THEORETICAL RESULTS

This section presents two-fold theoretical results that capture the privacy-utility tradeoff: (1) privacy guarantees, and (2) statistical inference guarantees. All proofs are provided in Appendix D.

4.1 Privacy Guarantees

Theorem 1 (Gaussian DP Guarantees)**.** *Algorithms 2 and 3 satisfy* $\sqrt{\mu_{bin}^2 + \mu_s^2 + \mu_t^2 + \mu_c^2}$*-GDP.*

The GDP guarantee follows directly from the design of our mechanisms and the composition properties of DP. Specifically, the binning step employs the PrivTree algorithm, which in our implementation we show ensures μ_{bin}-GDP. The remaining components, including the sum of covariates $\boldsymbol{x}$, the sum of response variable y, and the bin count, are each released using Gaussian mechanisms calibrated to privacy budgets μ_s, μ_t, and μ_c, respectively. By composition, their privacy losses combine, yielding the overall guarantee of $\sqrt{\mu_{\text{bin}}^2 + \mu_s^2 + \mu_t^2 + \mu_c^2}$-GDP.

Algorithm 2 and Algorithm 3 both inherit the same overall privacy bound, and one may choose one or the other depending on the use case and analysis goals. In particular, by the post-processing property of DP, any subsequent analysis, including LR, performed on the synthetic data does not incur additional privacy cost.

4.2 Statistical Inference

In this section, we establish the asymptotic normality of the proposed private estimator and use it to construct asymptotic CIs, accounting for noise added for privacy protection.

For the non-private weighted linear model in (3), the classical theory gives

$$\Sigma^{-1/2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(0, \boldsymbol{I}_d), \quad \Sigma = \sigma^2 (S^\top W S)^{-1}.$$

In the private model (5), a naive plug-in estimator takes the form $\widetilde{\Sigma}_{\text{naive}} = \sigma^2(\widetilde{S}^\top \widetilde{W} \widetilde{S})^{-1}$. However, the naive covariance estimator does not account for DP-induced uncertainty properly, leading to undercoverage of the resulting CI; see Table 1. Instead, we provide the following theoretical result for the proposed DP bias-corrected estimator.

Theorem 2 (Asymptotic Normality). *As $K \to \infty$ with $n \to \infty$, assume there exists a constant $c_0 > 0$ such that $\min_{1 \le k \le K} c_k \ge c_0$, and that the injected DP noises have finite variances. Define*

$$\widetilde{M} := \frac{1}{K}\Big(\widetilde{S}^\top \widetilde{W} \widetilde{S}\Big) - \widetilde{D},$$

and assume $\widetilde{M} \xrightarrow{p} M$ for some finite, nonsingular matrix M. The bias-corrected estimator

$$\tilde{\boldsymbol{\beta}} := (\widetilde{S}^\top \widetilde{W} \widetilde{S} - \widetilde{D})^{-1} \widetilde{S}^\top \widetilde{W} \widetilde{\boldsymbol{t}}$$

satisfies

$$\sqrt{K}\,(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}\big(\mathbf{0},\ M^{-1} H M^{-1}\big),$$

where

$$H := \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \mathrm{Var}\big(\boldsymbol{Q}_k(\boldsymbol{\beta})\big),$$
$$\boldsymbol{Q}_k(\boldsymbol{\beta}) := \tilde{\boldsymbol{s}}_k \tilde{w}_k \big(\tilde{t}_k - \tilde{\boldsymbol{s}}_k^\top \boldsymbol{\beta}\big) + \tilde{w}_k D_k \boldsymbol{\beta}.$$

Moreover, a consistent and private estimator of $\mathrm{Var}(\tilde{\boldsymbol{\beta}})$ *is given by*

$$\widetilde{\Sigma} = \frac{1}{K}\, \widetilde{M}^{-1} \widetilde{H}\, \widetilde{M}^{-1},$$

where

$$\widetilde{H} =: \frac{1}{K-d} \sum_{k=1}^{K} \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{Q}}_k^\top,$$
$$\widetilde{\boldsymbol{Q}}_k =: \tilde{\boldsymbol{s}}_k \tilde{w}_k (\tilde{t}_k - \tilde{\boldsymbol{s}}_k^\top \tilde{\boldsymbol{\beta}}) + \tilde{w}_k D_k \tilde{\boldsymbol{\beta}}.$$

Therefore, an asymptotic $(1-\alpha)$ confidence interval for β_j is

$$\tilde{\beta}_j \ \pm\ z_{\alpha/2} \sqrt{[\widetilde{\Sigma}]_{jj}}, \qquad j = 1, \ldots, d.$$

The asymptotic normality follows from the central limit theorem, as in the non-private LR setting. The proposed CI is differentially private because it is constructed entirely from privatized quantities. It also accounts for the uncertainty introduced by the injected DP noise, as reflected in the construction of the covariance estimator $\widetilde{\Sigma}$.

5 EXPERIMENTS

In this section, we compare our algorithms with existing DP-LR and DP-SDG methods, and assess the validity of the private CIs from Theorem 2.

5.1 Simulation Studies

We compare our Algorithm 2 to two popular algorithms: DP-(S)GD and AdaSSP (the only two methods that are used for comparison in Amin et al. (2023)). Given our focus on small-scale datasets, we use the variant DP-GD for its more stable performance without the need to tune batch size. In addition, we also compare it to SSP (Dwork et al., 2014; Sheffet, 2017) and JLT (Sheffet, 2017) that also give CIs. The non-private OLS estimator is displayed for benchmarking purposes.

Covariates are drawn from Uniform($[0,1]^d$), and the true coefficients are drawn from Uniform($[1,2]^d$). The regression error scale is set to $\sigma = 1$. Details on the hyperparameter settings are provided in Appendix A.

We evaluate five methods under three settings. Figure 1 shows the estimation error across varying sample sizes in three settings with dimensions $d = 1, 5, 10$. The y-axis displays the averaged relative ℓ_2 error over 100 repetitions, defined as $\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 / \|\boldsymbol{\beta}\|_2$ where $\boldsymbol{\beta}$ denotes the true coefficients and $\tilde{\boldsymbol{\beta}}$ is any estimator. Precise conversion in Proposition 2.1 (i) is used for other (ε, δ)-DP methods by setting $\delta = 1/n^{1.1}$. The performance of the five algorithms falls into two tiers: SSP and JLT perform poorly, while the other three methods exhibit better accuracy. Among them, BinAgg and DP-GD perform comparably well overall, with AdaSSP slightly worse. Notably, BinAgg outperforms all others on the smallest-scale datasets with $d = 1$. At first glance, DP-GD may appear to perform slightly better in some settings (e.g., $d = 5$). However, achieving this typically requires extensive hyperparameter tuning, which is *computationally expensive* and can implicitly *leak privacy* through the tuned parameters. In practice, identifying an effective tuning grid is harder than in

simulation. By contrast, in our real-data evaluations (Section 5.2.1), DP-SGD mostly underperforms BinAgg. Moreover, neither DP-GD nor AdaSSP provide CIs.

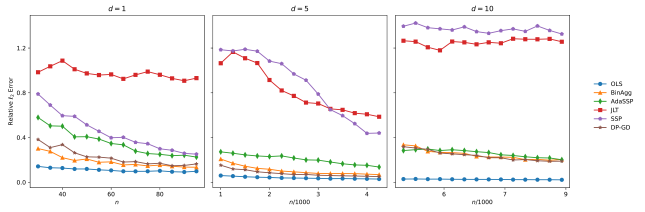

Figure 1: Coefficient estimation error across different (n, d) over 100 repetitions with $\mu = 1$

In addition, BinAgg is less sensitive to loose covariate bounds than AdaSSP. Figure 2 shows the error under varying bounds for $d = 5$. In AdaSSP, the covariate bounds directly determine sensitivity and thus the noise level. In contrast, BinAgg uses the bounds only to initialize bin boundaries, and bins with low noisy counts are discarded, mitigating the effect of loose initial bounds. More aggressive filtering of such low-count bins can be achieved by increasing the threshold.

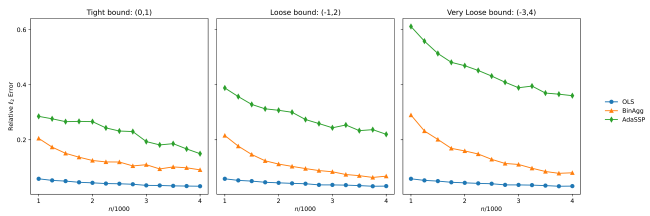

Figure 2: Coefficient estimation error using different covariate bounds with $d = 5$ and $\mu = 1$.

We also evaluate DP-CIs based on Theorem 2, with results shown in Table 1; SD stands for standard deviation. The naive CI discussed in Section 4 is included to illustrate that ignoring extra uncertainty due to DP underestimates the variance of the estimator, and the resulting CI significantly undercovers. Given that the JLT and SSP methods consistently give significantly poor performance, we do not include their CIs. Although they support uncertainty quantification, the estimation error is too large to be useful in practice. From Table 1, our theoretical standard deviations are very close to the empirical ones, and the empirical coverage is around the nominal level of 95%.

Table 1: BinAgg: Gaussian DP 95% CI over 2000 repetitions with $d = 5$, $n = 1000$, $\mu = 1$.

Avg. bias	Empirical SD	Avg. theor. SD	Naive theor. SD	Coverage	Naive coverage
-0.012	0.252	0.255	0.113	0.953	0.637
-0.008	0.262	0.268	0.117	0.950	0.654
-0.002	0.271	0.271	0.119	0.947	0.637
-0.004	0.298	0.307	0.129	0.947	0.637
0.004	0.503	0.521	0.194	0.957	0.597

5.2 Real Data Applications

5.2.1 Linear Regression

We apply Algorithm 2 to several small- to mid-scale real datasets accessible in the UCI Machine Learning Repository, covering application domains across various scientific areas. The data vary in size from $n = 182$ to $n = 21{,}263$ and in dimensionality from $d = 4$ to $d = 81$. For brevity, we denote these datasets as D1–D9 throughout the text; additional details and references are provided in Appendix B. Given the significantly worse performance of SSP and JLT, we focus our comparison on three methods: BinAgg, AdaSSP, and DP-GD, using non-private OLS as the benchmark. To evaluate prediction accuracy, in Table 2 we report the relative mean squared error (MSE), defined as $\|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_2^2 / \|\boldsymbol{y}\|_2^2$, where $\tilde{\boldsymbol{y}}$ denotes the predicted values. All methods on each dataset satisfy μ-GDP with $\mu = 1$.

Table 2: Relative MSE of prediction across datasets over 100 repetitions. Lowest error per dataset in **bold**; second lowest underlined. All DP methods satisfy μ-GDP with $\mu = 1$.

Dataset	Size (n, d)	OLS	BinAgg	AdaSSP	DP-GD
D1	(182, 4)	0.038	**0.095**	0.690	0.677
D2	(345, 6)	0.084	0.151	0.229	**0.102**
D3	(2043, 8)	0.023	**0.035**	0.669	0.693
D4	(4177, 10)	0.044	**0.059**	0.082	0.059
D5	(5875, 21)	0.011	**0.016**	0.080	0.062
D6	(6497, 12)	0.016	**0.022**	0.203	0.120
D7	(9357, 12)	0.441	**0.463**	0.682	0.852
D8	(19735, 27)	0.438	0.507	**0.500**	0.546
D9	(21263, 81)	0.131	**0.203**	0.429	0.420

For DP-GD, we use a grid search setup similar to Amin et al. (2023). For all other applicable methods, we use common non-private bounds computed from the observed dataset. These bounds are used solely for controlled comparison, following prior implementations that use non-private dataset-specific bounds or normalization (Amin et al., 2023; Wang, 2018). Further details and discussion are given in Appendix A. In practice, when public bounds are unavailable, one may instead use clipping based on domain knowledge or privately estimate bounds using a small privacy budget. While conservative bounds can reduce utility, BinAgg demonstrates greater robustness to loose bounds than AdaSSP; see Figure 2.

Table 2 shows that BinAgg outperforms the other two methods on most datasets, and at least achieves the second-lowest relative MSE. This demonstrates that BinAgg is *both effective and robust in practice*, with its performance advantage more pronounced than in the controlled simulation settings. In contrast, AdaSSP performs the worst overall. Although DP-GD occasionally approaches BinAgg, it does not surpass it on most

datasets. Moreover, DP-GD's performance is highly sensitive to hyperparameter tuning. Achieving competitive results requires an extensive search, which comes at a substantially higher computational cost.

5.2.2 Synthetic Data Generation

Our private synthetic data supports both scientific reproducibility with privacy guarantees and broader downstream tasks. To further assess downstream utility, we evaluate performance on additional widely used machine learning models for regression. In this setting, we compare Algorithm 3 against five DP-SDG approaches: AIM (Amin et al., 2023), BinAgg, DP-GAN (Xie et al., 2018), PATE-GAN (Jordon et al., 2018), and their enhanced variants DP-CTGAN and PATE-CTGAN (Xu et al., 2019). Implementation of all five algorithms is available in the open-source SmartNoise library (SmartNoise, 2021). AIM is a marginal-based method that requires discretizing the data, with the synthetic data inheriting the discretized structure. It has been shown to be the strongest marginal-based baseline (Chen et al., 2022). DP-GAN and PATE-GAN, together with their CTGAN extensions, represent approaches based on complex generative adversarial networks (GANs).

For each method, we generate 10 synthetic datasets and then train four regression models: XGBoost, Random Forest, Support Vector Regression (SVR), and Multilayer Perceptrons (MLPs). MLPs are excluded for datasets with fewer than 500 samples. "N/A" indicates inapplicability due to sample size constraints (PATE-GAN requires $n \geq 1000$) or computational limitations (in our experiments, AIM required 8 hours for one synthetic dataset, whereas the other methods completed within seconds or minutes). Competing methods are formulated under (ϵ, δ)-DP. To ensure fair comparison, we apply Proposition 2.1 to convert our μ-GDP guarantee into (ϵ, δ)-DP. For each dataset, we set $\epsilon = 1$ and $\delta = 1/n^{1.1}$. Predictive performance is measured in terms of relative MSE, averaged across 10 synthetic datasets; see Table 3.

Table 3: Average relative MSE across datasets using different DP-SDG methods. Lowest error per dataset in **bold**; second lowest underlined. All DP methods satisfy (ϵ, δ)-DP with $\epsilon = 1; \delta = 1/n^{1.1}$.

Dataset	Original	AIM	BinAgg	DP-CTGAN	PATE-CTGAN	DP-GAN	PATE-GAN
D1	0.286	1.400	**0.939**	1.091	1.344	2.398	N/A
D2	1.073	1.326	**1.015**	1.126	1.054	1.984	N/A
D3	0.875	**1.169**	2.208	17.298	4.325	12.619	8.789
D4	0.487	0.898	**0.731**	1.289	1.001	3.522	7.775
D5	0.033	1.058	**0.677**	1.139	1.000	2.527	2.241
D6	0.628	**0.908**	1.195	1.381	1.120	2.008	2.307
D7	0.489	0.614	**0.584**	1.231	1.039	1.488	1.753
D8	0.683	**1.079**	1.490	1.727	1.087	2.395	3.256
D9	0.240	N/A	**0.910**	39.621	2.538	8.024	2.460

Across these settings, BinAgg achieves the best overall performance: it attains the lowest error on 6 out of 9 datasets and, when not the best, performs very close to the top method. Its performance is followed by AIM and then PATE-CTGAN. Beyond accuracy, BinAgg is also markedly faster than the competing methods. AIM requires 221.08 seconds per synthetic dataset on average across datasets D1–D8, whereas BinAgg requires only 0.13 seconds and PATE-CTGAN 6.24 seconds over the same datasets. This translates to BinAgg being approximately 1,700× faster than AIM, and 48× faster than PATE-CTGAN. Details of the running times and computing environments are provided in Appendix B. These results highlight that BinAgg delivers both high utility and exceptional efficiency, making it particularly well-suited for regimes where both accuracy and computational efficiency are critical.

6 DISCUSSION

We propose BinAgg, a unified DP framework for LR with valid inference and regression-aware SDG. Binning preserves the covariate–response distribution for statistically faithful synthetic data, while aggregation maintains linear relationships necessary for inference in linear models.

Since BinAgg preserves information at the covariate–bin level, it can potentially support regression and inference on arbitrary subsets of covariates without additional privacy cost. Thus, promising future directions include the exploration of model selection procedures such as the F-test, AIC, and BIC when DP uncertainty is properly accounted for. Treatment effect comparisons, a cornerstone of randomized controlled trials, can be carried out directly on synthetic data, offering a path toward privacy-preserving causal inference. Beyond linear models, the released synthetic datasets enable analyses using other statistical or ML methods.

Our approach is primarily designed for small- to moderate-dimensional settings where binning remains computationally feasible and statistically stable. Extending BinAgg to high-dimensional regimes, where the curse of dimensionality affects partition-based methods, is a natural direction for future work. The method also assumes a prespecified bounded domain for covariates and responses. While such bounds are standard in many applications, developing principled and utility-preserving procedures for privately estimating or adapting bounds is another important direction. In addition, while we establish asymptotic guarantees for estimation and inference, sharper non-asymptotic results would provide a more precise understanding of when the asymptotic normality approximation is appropriate. Finally, improved binning strategies with optimized privacy allocation and adaptive tuning may further enhance accuracy and broaden applicability.

Acknowledgments

This work was supported at Penn State by the Huck Institutes of the Life Sciences through the Dorothy Foehr Huck and J. Lloyd Huck Chair in Data Privacy and Confidentiality, and by a 2025–2026 Rising Researcher Grant from the Institute for Computational and Data Sciences (RRID: SCR_025154).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K. and Zhang, L. (2016), 'Deep learning with differential privacy', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)* pp. 308–318.

Alabi, D., McMillan, A., Sarathy, J., Smith, A. D. and Vadhan, S. P. (2022), 'Differentially private simple linear regression', *Proceedings on Privacy Enhancing Technologies* **2022**(1), 184–204.

Amin, K., Joseph, M., Ribero, M. and Vassilvitskii, S. (2023), 'Easy differentially private linear regression', *International Conference on Learning Representations (ICLR)* .

Bassily, R., Smith, A. D. and Thakurta, A. (2014), 'Private empirical risk minimization: Efficient algorithms and tight error bounds', *55th IEEE Symposium on Foundations of Computer Science (FOCS)* pp. 464–473.

Cai, K., Lei, X., Wei, J. and Xiao, X. (2021), 'Data synthesis via differentially private markov random fields', *Proceedings of the VLDB Endowment* **14**(11), 2190–2202.

Cai, T. T., Wang, Y. and Zhang, L. (2021), 'The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy', *The Annals of Statistics* **49**(5), 2825–2850.

Candanedo, L. (2017), 'Appliances Energy Prediction', UCI Machine Learning Repository. DOI: 10.24432/C5VC8G.

Chen, K., Li, X., McKenna, R., Wang, T. et al. (2022), 'Benchmarking differentially private tabular data synthesis algorithms', *Workshop: Will Synthetic Data Finally Solve the Data Access Problem?* .

Cortez, P., Cerdeira, A., Almeida, F., Matos, T. and Reis, J. (2009), 'Wine Quality', UCI Machine Learning Repository. DOI: 10.24432/C56S3T.

Dick, T., Gillenwater, J. and Joseph, M. (2023), 'Better private linear regression through better private feature selection', *Advances in Neural Information Processing Systems* .

Dimitrakakis, C., Nelson, B., Mitrokotsa, A. and Rubinstein, B. I. P. (2014), 'Robust and private bayesian inference', *Algorithmic Learning Theory* **8776**.

Dong, J., Roth, A. and Su, W. J. (2022), 'Gaussian differential privacy', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **84**(1), 3–37.

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

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

Dwork, C., Talwar, K., Thakurta, A. and Zhang, L. (2014), 'Analyze gauss: Optimal bounds for privacy-preserving principal component analysis', *Symposium on Theory of Computing (STOC)* pp. 11–20.

Ferrando, C., Wang, S. and Sheldon, D. (2022), 'Parametric bootstrap for differentially private confidence intervals', *International Conference on Artificial Intelligence and Statistics (AISTATS)* pp. 1598–1618.

Hamidieh, K. (2018), 'Superconductivity Data', UCI Machine Learning Repository. DOI: 10.24432/C53P47.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K. and De Wolf, P.-P. (2012), *Statistical Disclosure Control*, Wiley, New York.

Jordon, J., Yoon, J. and van der Schaar, M. (2018), 'Pate-gan: Generating synthetic data with differential privacy guarantees', *International Conference on Learning Representations (ICLR)* .

Kifer, D., Smith, A. and Thakurta, A. (2012), 'Private convex empirical risk minimization and high-dimensional regression', *Proceedings of the 25th Annual Conference on Learning Theory (COLT)* **23**, 25.1–25.40.

Lin, S., Paquette, E. and Kolaczyk, E. D. (2024), 'Differentially private linear regression with linked data', *Harvard Data Science Review* **6**(3). https://hdsr.mitpress.mit.edu/pub/4if53bjq.

Liu, Y., Sun, K., Kong, L. and Jiang, B. (2022), 'Identification, β amplification and measurement: A bridge to gaussian differential privacy', *arXiv* **abs/2210.09269**.

Liver Disorders (2016), UCI Machine Learning Repository. DOI: 10.24432/C54G67.

McKenna, R., Mullins, B., Sheldon, D. and Miklau, G. (2022), 'AIM: An adaptive and iterative mechanism for differentially private synthetic data', *Proceedings of the VLDB Endowment* **15**(11), 2599–2612.

Milionis, J., Kalavasis, A., Fotakis, D. and Ioannidis, S. (2022), 'Differentially private regression with unbounded covariates', *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)* **151**, 3242–3273.

Minami, K., Arai, H., Sato, I. and Nakagawa, H. (2016), 'Differential privacy without sensitivity', *Advances in Neural Information Processing Systems 29* .

Nash, W., Sellers, T., Talbot, S., Cawthorn, A. and Ford, W. (1994), 'Abalone', UCI Machine Learning Repository. DOI: 10.24432/C55C7W.

National Academies of Sciences, Engineering, and Medicine (2019), *Reproducibility and Replicability in Science*, The National Academies Press, Washington, DC.

Ordoni, E., Bach, J. and Fleck, A.-K. (2022), 'Auction Verification', UCI Machine Learning Repository. DOI: 10.24432/C52K6N.

Sheffet, O. (2017), 'Differentially private ordinary least squares', *Proceedings of the 34th International Conference on Machine Learning (ICML)* **70**, 3105–3114.

Singh, A. (2022), 'LT-FS-ID: Intrusion Detection in WSNs', UCI Machine Learning Repository. DOI: 10.3390/s22031070.

Slavković, A. and Seeman, J. (2023), 'Statistical data privacy: A song of privacy and utility', *Annual Review of Statistics and Its Application* **10**, 189–218.

SmartNoise (2021), 'Implementation of dpgan, dp-ctgan, pategan, and pate-ctgan', `https://github.com/opendp/smartnoise-sdk/tree/a99f004732d7779f082a09037c5204165a94e81e/sdk/opendp/smartnoise/synthesizers/pytorch/nn`. [released 13-Jul-2021].

Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A. and Miklau, G. (2022), 'Benchmarking differentially private synthetic data generation algorithms', *arXiv preprint arXiv:2112.09238* .

Tsanas, A. and Little, M. (2009), 'Parkinsons Telemonitoring', UCI Machine Learning Repository. DOI: 10.24432/C5ZS3N.

van Kesteren, E.-J. (2024), 'To democratize research with sensitive data, we should make synthetic data more accessible', *Patterns* **5**(9), 101049.

Varshney, P., Thakurta, A. and Jain, P. (2022), '(Nearly) optimal private linear regression for subgaussian data via adaptive clipping', *Proceedings of the 35th Conference on Learning Theory (COLT)* **178**, 1126–1166.

Vito, S. (2008), 'Air Quality', UCI Machine Learning Repository. DOI: 10.24432/C59K5F.

Wang, Y.-X. (2018), 'Revisiting differentially private linear regression: Optimal and adaptive prediction & estimation in unbounded domain', *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)* .

Webb, K. R., Mukherjee, S., Mustafi, A., Slavković, A. and Vilhuber, L. (2026), 'Assessing utility of differential privacy for rcts'.
URL: *https://arxiv.org/abs/2309.14581*

Xie, L., Lin, K., Wang, S., Wang, F. and Zhou, J. (2018), 'Differentially private generative adversarial network', *arXiv preprint arXiv:1802.06739* .

Xin, B., Geng, Y., Hu, T., Chen, S., Yang, W., Wang, S. and Huang, L. (2022), 'Federated synthetic data generation with differential privacy', *Neurocomputing* **468**, 1–10.

Xin, B., Yang, W., Geng, Y., Chen, S., Wang, S. and Huang, L. (2020), 'Private fl-gan: Differential privacy synthetic data generation based on federated learning', *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 2927–2931.

Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K. (2019), 'Modeling tabular data using conditional GAN', *Advances in Neural Information Processing Systems 33* .

Zhang, J., Xiao, X. and Xie, X. (2016), 'Privtree: A differentially private algorithm for hierarchical decompositions', *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)* .

Zhang, J., Zhang, Z., Xiao, X., Yang, Y. and Winslett, M. (2012), 'Functional mechanism: Regression analysis under differential privacy', *Proceedings of the VLDB Endowment* **5**(11), 1364–1375.

Zhang, Z., Wang, T., Li, N., Honorio, J., Backes, M., He, S., Chen, J. and Zhang, Y. (2021), 'PrivSyn: Differentially private data synthesis', *30th USENIX Security Symposium (USENIX Security 21)* pp. 929–946.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

A Hyperparameter Detail

Simulation in Section 5.1. For hyperparameters of DP-GD, we use grid search over 252 combinations, with learning rate from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, the clipping norm from $\{1, 5, 10, 20, 50, 100\}$, and the number of epochs from $\{1, 5, 10, 20, 50, 100\}$. For all other methods, we pre-specify bounds on the covariates and response variable for clipping before data generation: $(0, 1)$ for $\boldsymbol{x}$ and $(0, 2)$, $(0, 7)$, $(0, 15)$ for y when $d = 1$, 5, and 10, respectively. For BinAgg, the privacy budget is allocated as $\mu_{\text{bin}} : \mu_c : \mu_s : \mu_t = 1 : 3 : 3 : 3$. We use the default value $\theta = 0$ in all experiments, noting that negative θ values lead to more aggressive splitting.

Simulation in Section 5.2. For DP-GD, we use a grid search with learning rates from $\{10^{-6}, 10^{-5}, 10^{-4}, \ldots, 1\}$, clipping norms from $\{10^{-6}, 10^{-5}, 10^{-4}, \ldots, 10^{6}\}$, and number of epochs from $\{1, 5, 10, 20, 50\}$, resulting in a total of 455 combinations. This setup is similar to that used in Amin et al. (2023) for evaluating real datasets. A caveat is that the grid search is conducted on the real data. For all other methods, we use the non-private data bounds. We acknowledge that these bounds are not DP, but this is acceptable for comparison purposes. In practice, applications are typically conducted by domain experts who have the knowledge to determine appropriate bounds and perform clipping before regression. If tight bounds are difficult to specify, one may either choose conservative bounds or use privatized ones. While conservative bounds often lead to reduced utility, BinAgg demonstrates greater robustness to loose bounds than AdaSSP, as shown in Figure 2.

B Additional Results for Experiments

Table 4: Datasets used in experiments with size and references.

ID	Dataset	Size (n, d)	Reference
D1	Intrusion Detection	(182, 4)	(Singh, 2022)
D2	Liver Disorders	(345, 6)	(*Liver Disorders*, 2016)
D3	Auction Verification	(2043, 8)	(Ordoni et al., 2022)
D4	Abalone Age	(4177, 10)	(Nash et al., 1994)
D5	Parkinson's Telemonitoring	(5875, 21)	(Tsanas and Little, 2009)
D6	Wine Quality	(6497, 12)	(Cortez et al., 2009)
D7	Air Quality	(9357, 12)	(Vito, 2008)
D8	Appliances Energy	(19735, 27)	(Candanedo, 2017)
D9	Superconductivity	(21263, 81)	(Hamidieh, 2018)

Table 5: Average runtime per synthetic dataset (seconds).

Dataset	AIM	BinAgg	DP-CTGAN	PATE-CTGAN	DP-GAN	PATE-GAN
Intrusion Detection	28.6317	**0.0097**	2.1385	1.9045	1.3252	N/A
Liver Disorders	48.7492	**0.0127**	1.0844	1.8050	0.7650	N/A
Auction Verification	82.9825	**0.0282**	1.8096	2.6069	0.6688	3.0788
Abalone Age	130.1068	**0.0662**	5.8339	4.7986	1.3709	4.2810
Parkinson's Telemonitoring	407.7839	**0.0862**	13.3774	6.6898	3.1690	6.0159
Wine Quality	210.8847	**0.0566**	11.9014	10.8458	2.0291	6.9002
Air Quality	236.7886	**0.1444**	24.6687	5.2088	2.7882	5.4930
Appliance Energy	622.7202	**0.6674**	113.1582	16.0563	11.4593	12.6062
Superconductivity	N/A	**2.5845**	968.1989	35.1800	163.6625	24.0455

Computing environment: All experiments were conducted on a laptop with an 8-core AMD Ryzen 7 8845HS CPU and 16 GB of RAM. All synthesizers were run on the same CPU.

C Algorithm: PrivTree

Algorithm 4 PrivTree (D, λ, θ, τ) (Zhang et al., 2016)

Initialize a tree $\mathcal{T}$ with a root node v_1
Set domain$(v_1) = \Omega$, and mark v_1 as `unvisited`
while there exists an `unvisited` node v **do**
    Mark v as `visited`
    Compute a biased point count for v with decaying factor τ:
        $b(v) = c(v) - \text{depth}(v) \cdot \tau$
    Adjust $b(v)$ if it is excessively small:
        $b(v) = \max\{b(v), \theta - \tau\}$
    Compute a noisy version of $b(v)$: $\hat{b}(v) = b(v) + \text{Lap}(\lambda)$
    if $\hat{b}(v) > \theta$ **then**
        Split v and add its children to $\mathcal{T}$
        Mark the children of v as `unvisited`
    end if
end while
return $\mathcal{T}$ with all point counts removed

Lemma (Corollary 1, Zhang et al. (2016)). *Let κ be the branching factor of tree $\mathcal{T}$. PrivTree satisfies ε-differential privacy if*

$$\lambda \geq \frac{2\kappa - 1}{\kappa - 1} \cdot \frac{1}{\varepsilon} \quad \textit{and} \quad \tau = \lambda \cdot \ln \kappa.$$

D Proofs

Proof of Corollary 1. For each bin k, let $\tilde{\boldsymbol{x}}^{(k,i)}$ denote the privatized version of the i-th sample within the bin. Given that $\tilde{c}_k$ is fixed, by construction of the Gaussian mechanism, each privatized vector has expectation

$$\mathbb{E}\Big(\tilde{\boldsymbol{x}}^{(k,i)}\Big) = \frac{\boldsymbol{s}_k}{\tilde{c}_k},$$

and variance

$$\text{Var}\Big(\tilde{\boldsymbol{x}}^{(k,i)}\Big) = \frac{\boldsymbol{\Delta}_k^2}{\tilde{c}_k\, \mu_s^2}.$$

Because the $\tilde{\boldsymbol{x}}^{(k,i)}$'s are independent Gaussian random variables, their sum is also Gaussian, with mean equal to the sum of means and variance equal to the sum of variances. Consequently,

$$\tilde{\boldsymbol{s}}'_k = \sum_{i=1}^{\tilde{c}_k} \tilde{\boldsymbol{x}}^{(k,i)} \sim \mathcal{N}\left(\boldsymbol{s}_k, \frac{\boldsymbol{\Delta}_k^2}{\mu_s^2}\right).$$

This distribution matches exactly the law of $\tilde{\boldsymbol{s}}_k$, the directly privatized statistic. Hence, the two constructions are distributionally equivalent. An identical argument, applied to the statistics $\tilde{t}'_k$ and $\tilde{t}_k$, shows the same distributional equivalence holds for the response terms. □

Proof of Theorem 1(Algorithm 2). The algorithm privatizes the counts c_k, the sums of covariates s_k, and the sums of responses t_k. By Proposition 2.4, the Gaussian mechanism applied to each of these statistics ensures Gaussian DP (GDP) with parameters μ_c, μ_s, and μ_t, respectively.

The initial binning procedure itself incurs a privacy cost quantified by μ_{bin}. By composition of GDP (Proposition 2.2), the overall privacy is given by $\sqrt{\mu_{\text{bin}}^2 + \mu_s^2 + \mu_t^2 + \mu_c^2}$-GDP. By the post-processing property (Proposition 2.3), all subsequent steps of the algorithm satisfy the same privacy guarantee. □

Proof of Theorem 1 (Algorithm 3). First, the procedure privatizes the bin counts c_k. By Proposition 2.4, this step satisfies μ_c-GDP. Next, the algorithm generates $\tilde{c}_k$ synthetic records per bin k. Each synthetic feature vector $\tilde{\boldsymbol{x}}^{(k,i)}$ is generated by adding Gaussian noise calibrated to ensure $\mu_s/\sqrt{\tilde{c}_k}$-GDP, while each synthetic label $\tilde{y}^{(k,i)}$ is privatized with $\mu_t/\sqrt{\tilde{c}_k}$-GDP. Because there are $\tilde{c}_k$ such records, the composition property implies that the total privacy cost across all synthetic samples in a given bin is

$$\sqrt{\tilde{c}_k\left(\frac{\mu_s^2}{\tilde{c}_k}+\frac{\mu_t^2}{\tilde{c}_k}\right)}=\sqrt{\mu_s^2+\mu_t^2}.$$

Combining this contribution with the binning privacy loss and the count privatization, and by composition and post-processing, the overall privacy guarantee of Algorithm 3 is

$$\sqrt{\mu_{\text{bin}}^2+\mu_s^2+\mu_t^2+\mu_c^2}\text{-GDP}.$$

□

Proof of Theorem 2. The result concerns the asymptotic distribution of the bias-corrected estimator $\tilde{\boldsymbol{\beta}}$. The argument follows from the general theory of estimating equations.

We begin by recalling the noise model. Let $\xi_{\boldsymbol{s}_k}\sim\mathcal{N}(\mathbf{0},\Sigma_{\boldsymbol{s}_k})$ denote the Gaussian noise added to the bin-level summary vector $\boldsymbol{s}_k$, and let $\xi_{t_k}\sim\mathcal{N}(0,\sigma_t^2)$ denote the noise added to t_k. Then the privatized quantities are

$$\tilde{\boldsymbol{s}}_k=\boldsymbol{s}_k+\xi_{\boldsymbol{s}_k},\qquad \tilde{t}_k=t_k+\xi_{t_k}=\boldsymbol{s}_k^\top\boldsymbol{\beta}+\eta_k+\xi_{t_k},$$

where $\eta_k\sim\mathcal{N}(0,\sigma^2c_k)$ is the aggregated regression noise.

The estimator $\tilde{\boldsymbol{\beta}}$ is defined implicitly as the solution to the estimating equation

$$\boldsymbol{Q}(\boldsymbol{b})=\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{Q}_k(\boldsymbol{b})=\mathbf{0},$$

with

$$\boldsymbol{Q}_k(\boldsymbol{b})=-\tilde{\boldsymbol{s}}_k\tilde{w}_k\big(\tilde{t}_k-\tilde{\boldsymbol{s}}_k^\top\boldsymbol{b}\big)-\tilde{w}_kD_k\boldsymbol{b},\qquad D_k=\mathbb{E}(\xi_{\boldsymbol{s}_k}\xi_{\boldsymbol{s}_k}^\top).$$

At the true parameter $\boldsymbol{\beta}$, we have

$$\begin{aligned}\mathbb{E}[\boldsymbol{Q}_k(\boldsymbol{\beta})]&=-\mathbb{E}\Big[(\boldsymbol{s}_k+\xi_{\boldsymbol{s}_k})\tilde{w}_k\big(\boldsymbol{s}_k^\top\boldsymbol{\beta}+\eta_k+\xi_{t_k}-(\boldsymbol{s}_k+\xi_{\boldsymbol{s}_k})^\top\boldsymbol{\beta}\big)\Big]-\mathbb{E}[\tilde{w}_kD_k\boldsymbol{\beta}]\\&=\mathbb{E}[\tilde{w}_k\xi_{\boldsymbol{s}_k}\xi_{\boldsymbol{s}_k}^\top]\boldsymbol{\beta}-\mathbb{E}[\tilde{w}_k]D_k\boldsymbol{\beta}\\&=0,\end{aligned}$$

due to the independence between $\xi_{\boldsymbol{s}_k}$ and $\tilde{w}_k$. Hence, the estimating equation is unbiased.

We next analyze the asymptotic distribution. Because $\boldsymbol{Q}(\boldsymbol{b})$ is linear in $\boldsymbol{b}$, its Jacobian with respect to $\boldsymbol{b}$ is constant:

$$\widetilde{M}=\frac{\partial\boldsymbol{Q}(\boldsymbol{b})}{\partial\boldsymbol{b}}=\frac{1}{K}\sum_{k=1}^{K}\big(\tilde{w}_k\tilde{\boldsymbol{s}}_k\tilde{\boldsymbol{s}}_k^\top-\tilde{w}_kD_k\big)=\frac{1}{K}\,\widetilde{S}^\top\widetilde{W}\widetilde{S}-\widetilde{D},$$

where $\widetilde{D}=\frac{1}{K}\sum_{k=1}^{K}\tilde{w}_kD_k$. A first-order Taylor expansion of $\boldsymbol{Q}(\cdot)$ around $\boldsymbol{\beta}$ yields

$$\mathbf{0}=\boldsymbol{Q}(\tilde{\boldsymbol{\beta}})=\boldsymbol{Q}(\boldsymbol{\beta})+\widetilde{M}(\tilde{\boldsymbol{\beta}}-\boldsymbol{\beta}),$$

which rearranges to

$$\tilde{\boldsymbol{\beta}}-\boldsymbol{\beta}=-\widetilde{M}^{-1}\boldsymbol{Q}(\boldsymbol{\beta}).$$

Since the bins $\{B_k\}$ are disjoint, the regression errors $\{\eta_k\}$ are independent, and the DP noises $\{(\xi_{\boldsymbol{s}_k}, \xi_{t_k})\}$ are generated independently across bins, the vectors $\{\boldsymbol{Q}_k(\boldsymbol{\beta})\}_{k=1}^K$ are independent with finite second moments, so the multivariate CLT applies. We then have

$$\sqrt{K}\,\boldsymbol{Q}(\boldsymbol{\beta}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} \boldsymbol{Q}_k(\boldsymbol{\beta}) \;\xrightarrow{d}\; \mathcal{N}(\mathbf{0}, H), \qquad H = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathrm{Var}\big(\boldsymbol{Q}_k(\boldsymbol{\beta})\big).$$

If $\widetilde{M} \xrightarrow{p} M$ with M nonsingular, Slutsky's theorem yields

$$\sqrt{K}\,(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}) \;\xrightarrow{d}\; \mathcal{N}\big(\mathbf{0},\; M^{-1} H M^{-1}\big).$$

A consistent estimator of H is given by its corrected sample analogue

$$\widetilde{H} = \frac{1}{K-d} \sum_{k=1}^{K} \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{Q}}_k^\top, \quad \widetilde{\boldsymbol{Q}}_k = \tilde{\boldsymbol{s}}_k \tilde{w}_k (\tilde{t}_k - \tilde{\boldsymbol{s}}_k^\top \tilde{\boldsymbol{\beta}}) + \tilde{w}_k D_k \tilde{\boldsymbol{\beta}},$$

so a consistent estimator of $\mathrm{Var}(\tilde{\boldsymbol{\beta}})$ is

$$\widetilde{\Sigma} = \frac{1}{K}\, \widetilde{M}^{-1} \widetilde{H}\, \widetilde{M}^{-1}.$$

This establishes asymptotic normality of the bias-corrected estimator, along with consistent covariance estimation. □