Differentially Private Federated Learning: Servers Trustworthiness, Estimation, and Statistical Inference

Zhe Zhang* Ryumei Nakada† Linjun Zhang‡

April 26, 2024

Abstract

Differentially private federated learning is crucial for maintaining privacy in distributed environments. This paper investigates the challenges of high-dimensional estimation and inference under the constraints of differential privacy. First, we study scenarios involving an untrusted central server, demonstrating the inherent difficulties of accurate estimation in high-dimensional problems. Our findings indicate that the tight minimax rates depends on the high-dimensionality of the data even with sparsity assumptions. Second, we consider a scenario with a trusted central server and introduce a novel federated estimation algorithm tailored for linear regression models. This algorithm effectively handles the slight variations among models distributed across different machines. We also propose methods for statistical inference, including coordinate-wise confidence intervals for individual parameters and strategies for simultaneous inference. Extensive simulation experiments support our theoretical advances, underscoring the efficacy and reliability of our approaches.

1 Introduction

1.1 Overview

Federated learning is an efficient approach for training machine learning models on distributed networks, such as smartphones and wearable devices, without moving data to a central server [26, 25, 30]. Since its proposal in [32], federated learning has gained significant attention in both practical and theoretical machine learning communities. One of the key attractions of federated learning is its ability to provide a certain level of data privacy by keeping raw data on local machines. However, without specific design choices, there are no formal privacy guarantees. To fully exploit the benefits of federated learning, researchers have introduced the concept of differential privacy [1, 12, 13, 14, 15]

*Rutgers University. Email: zzres0131@gmail.com.
†Rutgers University. Email: rn375@stat.rutgers.edu.
‡Rutgers University. Email: lz412@stat.rutgers.edu.

to quantify the exact privacy level in federated learning. A series of research papers have focused on federated learning with differential privacy, applying various algorithms and methods [23, 37, 39]. Despite these efforts, there remains a significant gap between practical usage and statistical guarantees, particularly in the high-dimensional setting with sparsity assumptions, where theoretical results for the optimal rate of convergence and statistical inference results are largely missing.

In this paper, we focus on studying the estimation and inference problems in the federated learning setting under differential privacy, particularly in the high-dimensional regime. In federated learning, there are several local machines containing data sets from different sources, and a central server to coordinate all local machines to train learning models collaboratively. We present our key results in two major settings for privacy and federated learning. In the first setting, we consider an untrusted central server [31, 39, 23] where each machine sends only privatized information to the central server. For example, when using smartphones, where users may not fully trust the server and do not want their personal information to be directly updated on the remote central server. In the second setting, we consider a trusted central server where each machine sends raw information without making it private. [33, 19, 34] For example, in different hospitals, patient data may not be shared among hospitals to protect patient privacy, but they can all report their data to a central server, such as a non-profit organization or an institute, to gain more information and publish statistics on certain diseases.

In the first part of our paper, we demonstrate that under the assumption that the central server is untrusted, the optimal rate of convergence for mean estimation is $\mathrm{O}(sd/(mn\epsilon^2))$, where m is the number of local machines and each containing n data points, d is the parameter of interest, s is the sparsity level, and ϵ is the privacy parameter. As commonly assumed in high-dimensional settings where the dimension is comparable or even larger than the number of data, such an optimality result shows the incompatibility of untrusted central server setting and high-dimensional statistics. As a result, we can only hope to get a good estimation under the trusted central server setting in the high-dimensional regime.

In the second part of the paper, we consider the case of a trusted central server and design algorithms that allow for accurate estimations and obtain a near-optimal rate of convergence up to logarithm factors. We also present statistical inference results, including the construction of coordinate-wise confidence intervals with privacy guarantees, and the solution to conduct simultaneous inference privately. This will assist in hypothesis testing problems and construction of confidence intervals for a given subset of indices of a vector simultaneously in high-dimensional settings. We emphasize that our algorithms for estimation and inference are suited for practical purposes, considering its capacity to (1) leverage data from multiple devices to improve machine learning models and (2) draw accurate conclusions about a population from a sample while preserving individual privacy. For instance, in healthcare, we could combine patient data from multiple hospitals to develop more accurate models for disease diagnosis and treatment, while ensuring that patient privacy is protected. We summarize our major contributions as follows:

- For the untrusted central server setting, we provably show that federated learning is

not suited for high-dimensional mean estimation problems by providing the optimal rate of convergence under the untrusted central server constraints. This suggests us to consider a trusted central server setting to utilize federated learning for such problems.

- For the trusted central server setting, we design novel algorithms to achieve private estimation with federated learning. We first consider the estimation in homogeneous federated learning setting and then we extend it to a more complicated heterogeneous federated learning setting. We also provide a sharp rate of convergence for our algorithm in both settings.

- In addition, we consider statistical inference problems in both homogeneous and heterogeneous federated learning settings. We provide algorithms for coordinate-wise and simultaneous confidence intervals, which are two common inference problems in high-dimensional statistics. It is worth mentioning that our proposed methods for high-dimensional differentially private inference problems are novel and unique, which has not been developed even for the single-source and non-federated learning setting. Theoretical results show that our proposed confidence intervals are asymptotically valid, supported by simulations.

1.2 Related Work

In the literature, several works focused on designing private algorithms in federated learning/distributed learning based on variants of stochastic gradient decent algorithms. [3] proposed a communication efficient algorithm, CP-SGD algorithm for learning models with local differential privacy (LDP). [17] proposed a distributed LDP gradient descent algorithm by applying LDP on gradients with ESA framework [6]. [20] extended works on LDP approach for federated learning and proposed a distributed communication-efficient LDP stochastic gradient descent algorithm through shuffled model and analyzed the upper bound of the convergence rate. However, the trade-off between statistical accuracy and the privacy cost has not been considered in these works.

In the distributed settings, the trade-off between statistical accuracy and information constraints has been discussed in various papers. Two common types of information constraints are communication constraints and privacy constraints. We refer to [43, 7, 21, 5, 18] for more discussions on communication constraints, considering the situation where the bits of the information during communication have constraints.

A series of work discusses the trade-off between accuracy and privacy in high-dimensional and non-federated learning problems, including top-k selection [35], sparse mean estimation [8], linear regression [8, 36], generalized linear models [9], latent variable models [46]. However, the discussion on privacy constraints in the distributed settings are still largely lacking. Among the existing works, most of them focus on the local differential privacy (LDP) constraint. In [4], the mean estimation under ℓ_2 loss for Gaussian and sparse Bernoulli distributions are discussed. [11] discussed the lower bounds under LDP constraints in the blackboard communication model for mean estimation of product Bernoulli

distributions and sparse Gaussian distributions. [2] proposed a more general approach to combine both communication constraints and privacy constraints. Compared with previous works, we focus on the problem where there are n data points on each machine. Our interest lies in the (ϵ, δ)-DP instead of LDP, which is a weaker constraint containing broader settings. We further note that, compared with the blackboard communication model [7, 18], in the federated learning setting, we assume that the existence of a central server and that each server is only allowed to communicate with the central server. This setting enables us to enhance more privacy.

When we are finalizing this paper[1], we realized an independent and concurrent work [28]. [28] also considers differentially private federated transfer learning under high-dimensional sparse linear regression model. Namely, they proposed a notion of federated differential privacy that allows multiple rounds of (ϵ, δ)-differentially private transmissions between local machines and the central server, and provides algorithms to filter out irrelevant sources, and exploit information from relevant sources to improve the performance of estimation of target parameters. We differntiate our research with their paper as follows: (1) While they consider differentially private federated learning under untrusted server setting, we deal with both trusted and untrusted server settings. We also highlight a fundamental difficulty of pure ϵ-differentially private estimation under untrusted central server settings in federated learning by establishing a tight minimax lower bound, and resort to trusted server settings for estimation and inference problems. (2) While their investigation centers on differentially private estimation within a federated transfer learning framework—specifically focusing on parameter estimation for a target distribution using similar source data—our work focuses on private estimation and *inference* for parameters that are either common across all participating machines, or vary across different machines.

We also cite papers that provided us inspirations for the design of our proposed algorithms and methods. [24] introduces a de-biasing produce for the statistical inference problems. [29] considers the transfer learning problem in high-dimensional settings, which enables us to combine information from other sources to benefit the estimation problems. Such idea could be adopted in the hetergeneous federated learning problems. For the simultaneous inference problems, we refer to [42, 41], which discussed how to conduct simultaneous inference for high-dimensional problems.

Notation. We introduce several notations used throughout the paper. Let $\boldsymbol{v} = (v_1, v_2, \ldots, v_d)^\top \in \mathbb{R}^d$ represent a vector. Given a set of indices $\mathcal{S}$, $\boldsymbol{v}_\mathcal{S}$ refers to the components of $\boldsymbol{v}$ corresponding to the indices in $\mathcal{S}$. The ℓ_q norm of $\boldsymbol{v}$, for $1 \leq q \leq \infty$, is given by $\|\boldsymbol{v}\|_q$, whereas $\|\boldsymbol{v}\|_0$ represents the number of non-zero elements in $\boldsymbol{v}$, also called as its sparsity level.

We use m to indicate the number of machines, n for the number of samples per machine, d for the dimensionality of vectors, and s for their sparsity level. The total number of samples across all machines is denoted by $n_0 = m \cdot n$. Additionally, we define the truncation function $\Pi_T : \mathbb{R}^d \to \mathbb{R}^d$, which projects a vector onto the ℓ_∞ ball of radius T centered at the origin.

[1] An initial draft of this paper was published as a Ph.D. dissertation in 2023 [45].

For a matrix Σ, $\max_{\|\boldsymbol{v}\|_2=1,\|\boldsymbol{v}\|_0\leq s} \boldsymbol{v}^\top\Sigma\boldsymbol{v}$ and $\min_{\|\boldsymbol{v}\|_2=1,\|\boldsymbol{v}\|_0\leq s} \boldsymbol{v}^\top\Sigma\boldsymbol{v}$ denote the largest and smallest s-restricted eigenvalues of Σ, denoted as $\mu_s(\Sigma)$ and $\nu_s(\Sigma)$, respectively.

For sequences a_n and b_n, $a_n = o(b_n)$ implies $a_n/b_n \to 0$ as n grows, $a_n = O(b_n)$ signifies that a_n is upper bounded by a constant multiple of b_n, and $a_n = \Omega(b_n)$ indicates that a_n is lower bounded by a constant multiple of b_n, where constants are independent of n. The notation $a_n \asymp b_n$ denotes that a_n is both upper and lower bounded by constant multiples of b_n.

In this work, we often use symbols $c_0, c_1, m_0, m_1, C, C', K, K'$ to represent universal constants. Their specific values may vary depending on the context, but they are independent from other tunable parameters.

2 Preliminaries

2.1 Differential Privacy

We start form the basic concepts and properties of differential privacy [13]. The intuition behind differential privacy is that a randomized algorithm produces similar outputs even when an individual's information in the dataset is changed or removed, thereby preserving the privacy of individual data. The formal definition of differential privacy is given below.

Definition 2.1 (Differential Privacy [13]) *Let $\mathcal{X}$ be the sample space for an individual data, a randomized algorithm $M : \mathcal{X}^n \to \mathbb{R}$ is (ϵ, δ)-differentially private if and only if for every pair of adjacent data sets $\boldsymbol{X}, \boldsymbol{X}' \in \mathcal{X}^n$ and for any $S \subseteq \mathbb{R}$, the inequality below holds:*

$$\mathbb{P}\left(M(\boldsymbol{X}) \in S\right) \leq e^{\varepsilon} \cdot \mathbb{P}\left(M(\boldsymbol{X}') \in S\right) + \delta,$$

where we say that two data sets $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ and $\boldsymbol{X}' = \{\boldsymbol{x}_i'\}_{i=1}^n$ are adjacent if and only if they differ by one individual datum.

In the above definition, the two parameters ϵ, δ control the privacy level. From the definition, with smaller ϵ and δ, the outcomes given adjacent $\boldsymbol{X}$ and $\boldsymbol{X}'$ become closer, making it harder for an adversary to distinguish if the original dataset is $\boldsymbol{X}$ or $\boldsymbol{X}'$, indicating the privacy constraint becomes more stringent. Furthermore, when $\delta = 0$, we could use ϵ-differentially private as the abbreviation of $(\epsilon, 0)$-differentially private.

In the rest of this section, we introduce several useful properties of differential privacy and how to create a differential private algorithm from non-private counterparts. One common strategy is through noise injection. The scale of noise is characterized by the sensitivity of the algorithm:

Definition 2.2 *For any algorithm $f : \mathcal{X}^n \to \mathbb{R}^d$ and two adjacent data sets $\boldsymbol{X}$ and $\boldsymbol{X}'$, the ℓ_p-sensitivity of f is defined as:*

$$\Delta_p(f) = \sup_{\boldsymbol{X},\boldsymbol{X}'\in\mathcal{X}^n \ adjacent} \|f(\boldsymbol{X}) - f(\boldsymbol{X}')\|_p.$$

We then introduce two mechanisms. For algorithms with finite ℓ_1-sensitivity, we add Laplace noises to achieve differential privacy, while for ℓ_2-sensitivity, we inject Gaussian noises.

Proposition 2.3 (The Laplace Mechanism [13, 14]) *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a deterministic algorithm with $\Delta_1(f) < \infty$. For $\boldsymbol{w} \in \mathbb{R}^d$ with coordinates $w_1, w_2, \cdots, w_d$ be i.i.d samples drawn from Laplace$(\Delta_1(f)/\epsilon)$, $f(\boldsymbol{X}) + \boldsymbol{w}$ is $(\epsilon, 0)$-differentially private.*

Proposition 2.4 (The Gaussian Mechanism [13, 14]) *Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a deterministic algorithm with $\Delta_2(f) < \infty$. For $\boldsymbol{w} \in \mathbb{R}^d$ with coordinates $w_1, w_2, \cdots, w_d$ be i.i.d samples drawn from $N(0, 2(\Delta_2(f)/\epsilon)^2 \log(1.25/\delta))$, $f(\boldsymbol{X}) + \boldsymbol{w}$ is (ϵ, δ)-differentially private.*

The post-processing and composition properties are two key properties in differential privacy, which enable us to design complicated differentially private algorithms by combining simpler ones. Such properties are pivotal in the design of algorithms in later chapters.

Proposition 2.5 (Post-processing Property [13]) *Let M be an (ϵ, δ)-differentially private algorithm and g be an arbitrary function which takes $M(\boldsymbol{X})$ as input, then $g(M(\boldsymbol{X}))$ is also (ϵ, δ)-differentially private.*

Proposition 2.6 (Composition property [13]) *For $i = 1, 2$, let M_i be $(\varepsilon_i, \delta_i)$-differentially private algorithm, then (M_1, M_2) is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-differentially private algorithm.*

We also mention NoisyHT algorithm (Algorithm 1) introduced by [16], which stands for the noisy hard-thresholding algorithm. The algorithm aims to pursue both sparsity of the output and privacy at the same time.

Algorithm 1: Noisy Hard Thresholding Algorithm (NoisyHT$(\boldsymbol{v}, s, \lambda, \epsilon, \delta)$) [16]

Input: vector-valued function $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{X}) \in \mathbb{R}^d$ with data $\boldsymbol{X}$, sparsity s, privacy parameters ε, δ, sensitivity λ.

Initialization: $S = \emptyset$.

For i in 1 **to** s:

  Generate $\boldsymbol{w}_i \in \mathbb{R}^d$ with $w_{i1}, w_{i2}, \cdots, w_{id} \overset{\text{i.i.d.}}{\sim} \text{Laplace}\left(\lambda \cdot \frac{2\sqrt{3s\log(1/\delta)}}{\varepsilon}\right)$.

  Append $j^* = \text{argmax}_{j \in [d] \setminus S}(|v_j| + w_{ij})$ to S.

End For

  Generate $\tilde{\boldsymbol{w}}$ with $\tilde{w}_1, \cdots, \tilde{w}_d \overset{\text{i.i.d.}}{\sim} \text{Laplace}\left(\lambda \cdot \frac{2\sqrt{3s\log(1/\delta)}}{\varepsilon}\right)$.

Output: $P_S(\boldsymbol{v} + \tilde{w})$.

In the last step, $P_S(\boldsymbol{u})$ denotes the operator that makes $\boldsymbol{u}_{S^c} = 0$ while preserving $\boldsymbol{u}_S$. This algorithm could be seen as a private top-k selection algorithm, which helps build our proposed algorithm in later section.

Specifically, when the sparsity s is chosen to be 1, the algorithm outputs the maximum element chosen after a single iteration in the private manner. We refer this special case as the Private Max algorithm, which is implemented in Algorithm 7 used for simultaneous inference.

2.2 Federated Learning

Federated learning introduced in [32] is a technique designed to train a machine learning algorithm across multiple devices, without exchanging data samples. A central server coordinates the process, with each local machine sending model updates to be aggregated centrally. Figure 1 illustrates the basic concept of federated learning

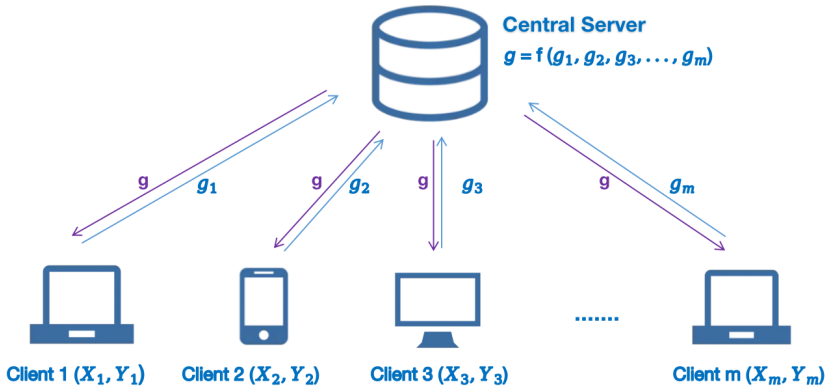

Figure 1: Federated Learning

One characteristic of federated learning is that the training of machine learning models occurs locally, and only parameters and updates are transferred to the central server and shared by each node. Specifically, communication between local machines and the server is bidirectional: machines send updates to the central server, and in return, they receive aggregated information after processing. Communication among local machines is prohibited to prevent privacy leakage. Intuitively, federated learning inherently provides a certain level of privacy.

Although without rigorous definitions, there are two main branches of central server settings in federated learning: the untrusted central server setting and the trusted central server setting [31, 39, 33, 19]. In the first setting, where the central server is untrusted, each piece of information sent from the machine to the central server should be differentially private. In the second setting, we assume a trusted central server exists. In this scenario, it is safe to send raw information from the machine to the central server without additional privacy measures. However, to prevent information leakage among local machines, the information sent back from the server should also be differentially private.

Another key aspect of federated learning is that the datasets on each local machine are commonly not independent and identically distributed (i.i.d.). This allows federated

learning to train on heterogeneous datasets, aligning with practical scenarios where the datasets on different machines are typically diverse and their sizes may also vary. We will demonstrate that federated learning can efficiently estimate the local model when models on different local machines differ but share some similarities, a concept we refer to as heterogeneous federated learning in Section 5.

2.3 Problem Formulation

In this paper, we assume that there exists a central server and m local machines. We denote the data on these machines by $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \ldots, \boldsymbol{X}_m$, respectively, with $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times d}$. On any machine $i = 1, 2, \ldots, m$, there are n_i data points $\boldsymbol{X}_i = [\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{in_i}]$. For simplicity, we assume that there are equal data points $n = n_1 = n_2 = \cdots = n_i$ for each machine. We note that the result could be easily generalized to cases where the sample sizes on each machine differ.

We consider both untrusted and trusted central server settings. For the untrusted setting, we require that the information sent from local machines to the server is private. In this scenario, we show that in the high-dimensional setting, even with sparsity assumptions, it is impossible to achieve small estimation error when the central server is untrusted. In the trusted setting, we consider the high-dimensional linear regression problem $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{W}$ with s-sparse $\boldsymbol{\beta}$. We will first study the case where all machines share the same $\boldsymbol{\beta}$, (referred to homogeneous federation learning,) and then study a more general case where models on different machines are not equal, but share certain similarities (referred to heterogeneous federation learning.) We show that our algorithm can adapt to such similarity—with larger similarity, the algorithm achieves a faster rate of convergence.

3 An Impossibility Result in the Untrusted Central Server setting

In this section, we study the untrusted server setting where the local machines need to send privatized information to the central server to ensure privacy. We show an impossibility result that in high-dimensional settings where the data dimension is comparable to or greater than the sample size, accurate estimation is not feasible even if we consider a simple sparse mean estimation problem.

As mentioned in Section 2.3, we consider a federated learning setting with m machines, where each machine $i \in [m]$ handles n data points $\boldsymbol{X}_i := [\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{in}] \in \mathbb{R}^{n \times d}$. Let $D_{\text{all}} = \{\boldsymbol{X}_i\}_{i=1}^m$. We assume that each data point $\boldsymbol{X}_{ij} \in \mathbb{R}^d$ follows a Gaussian distribution $N(\boldsymbol{\mu}, \boldsymbol{I}_d)$, where $\boldsymbol{\mu}$ is a sparse d-dimensional vector with sparsity s. The goal is to estimate $\boldsymbol{\mu}$ in the federated learning setting when the central server is untrusted. In this section, we provide an optimal rate of convergence for this problem and show that the untrusted central server setting is not suited for high-dimensional problems.

We begin by deriving the minimax lower bound, which characterizes the fundamental difficulty of this estimation problem. In untrusted server setting, we additionally assume

that each piece of information sent from the local machine to the central server follows ϵ-differential privacy. To achieve this, we introduce the privacy channel $\mathcal{W}^{\epsilon\text{-priv}} : \mathcal{X}^n \to \mathcal{Z}$, a function that is responsible for privatizing the information transmitted from the local machines. Given the input $X \in \mathcal{X}^n$ and the privacy channel $\mathcal{W}^{\epsilon\text{-priv}}$, $Z \in \mathcal{Z}$ representing all the information (from multiple rounds) transmitted to the central server. More precisely, we require privacy guarantees such that for any two adjacent datasets X and $X' \in \mathcal{X}^n$, differing by only one data point on any local machine, and for an output $Z \in \mathcal{Z}$ representing the information sent from the local machine to the central server, differential privacy guarantee $\mathbb{P}(\mathcal{W}^{\epsilon\text{-priv}}(X) = Z) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{W}^{\epsilon\text{-priv}}(X') = Z)$ holds.

We consider any mechanism M in the federated learning setting with m local machines and one central server, operated on the dataset D_{all}. M serves as a procedure to estimate $\boldsymbol{\mu}$, where each local machine collaborates exclusively with the central server without direct interaction among themselves. On each machine i, the mechanism M uses the privacy channel $\mathcal{W}_i^{\epsilon\text{-priv}}$ and data sample $\boldsymbol{X}_i$ to generate $\boldsymbol{Z}_i$, which is then transmitted to the central server. The central server receives the information from all machines. After multi-rounds of collaboration between local machines and the central server, we obtain the sparse and private estimator $\hat{\boldsymbol{\mu}} \in \mathbb{R}^d$. We denote the class of all mechanisms that satisfy the above constraints as $\mathcal{M}_{m,\epsilon}^{\text{untrust}}(D_{\text{all}})$. Under this setting we establish a lower bound for the estimation error of the mean in Theorem 1.

Theorem 1 *Suppose D_{all} is generated as above. Let μ be a s-sparse d-dimensional mean of Gaussian distribution satisfying $\|\boldsymbol{\mu}\|_\infty \leq 1$. We consider the estimation of the mean vector $\hat{\boldsymbol{\mu}}$ under the untrusted central server federated learning setting with m local machines and n data points in each machine. Then, there exists a constant $c > 0$ such that*

$$\inf_{M \in \mathcal{M}_{m,\epsilon}^{\text{untrust}}} \sup_{\boldsymbol{\mu} \in \mathbb{R}^d, \|\boldsymbol{\mu}\|_\infty \leq 1} \|\boldsymbol{\mu} - M(D_{\text{all}})\|_2^2 \geq c \cdot \min\left(\frac{s}{n}, \frac{sd}{mn\epsilon^2}\right).$$

The lower bound contains two terms. The first term, of order s/n, represents the minimax risk of mean estimation using only the samples from a local machine. The second term, of order $sd/(mn\epsilon^2)$, accounts for the error from federated learning across multiple machines under privacy constraints. Theorem 1 suggests that we cannot perform better than either choosing to estimate the mean using only the local machine or adopting the federated learning approach and combining information from different machines. However, in the latter approach, we must at least incur a rate of $O(d/(mn\epsilon^2))$, which is linearly proportional to the dimension d. This result suggests that privacy constraints significantly impact the efficiency of federated learning in high-dimensional settings. Furthermore, as the number of machines m increases, we can possibly attain better performance, highlighting the merit of federated learning.

We also show the tightness of the lower bound in Theorem 1 by providing the upper bound.

Theorem 2 *Suppose that conditions in Theorem 1 hold. Then, there exists an ϵ-differentially*

private algorithm for the estimation of $\boldsymbol{\mu}$ *as* $\hat{\boldsymbol{\mu}}$

$$\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2 \leq c \cdot \frac{s \cdot d}{mn\epsilon^2},$$

where $c > 0$ *is some constant.*

The proof follows by constructing an algorithm that transforms Gaussian mean to Bernoulli mean according to the sign of the Gaussian mean, motivated by Algorithm 2 discussed in [2], where the authors discuss l-bit protocol for estimating the product of Bernoulli family. More details of the algorithm are deferred to Section A.2. Based on the results from Theorems 1 and 2, we obtain the optimal rate of convergence for sparse mean estimation under differentially private federated learning setting. As a result, when the central server is untrusted, it is impossible to find an approach to achieve accurate estimation under the untrusted server assumption. This highlights the necessity of the trusted server setting for statistical estimation and inference in high-dimensional federated learning scenarios. In the following sections, we develop estimation and inference procedures under the trusted server settings.

4 Homogeneous Federated Learning Setting

4.1 Algorithms for Estimation Problems

In this section, we consider the setting of a trusted central server, where local machines fully trust the central server and send unprivatized information to it without implementing privacy measures. However, when the central server sends information back to the local machines, it must ensure that this information is privatized to avoid any privacy leakage across local machines.

In this subsection, we first focus on the statistical estimation problems in this setting and then develop inference results in the next subsection. More specifically, our primary focus is on the linear regression problem in a high-dimensional setting, where the ground truth, denoted as $\boldsymbol{\beta}$, is a sparse d-dimensional vector. We initially study the simpler case in this section, where the underlying generative models for each local machine are identical, which we refer to as the homogeneous federated learning setting. A more complicated heterogeneous setting will be discussed in the following section. Specifically, we consider the following high-dimensional linear regression model:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{W},$$

where we assume $\boldsymbol{W}$ is the error term whose coordinates are independent and following sub-Gaussian distribution with variance proxy σ^2, denoted by $W_i \sim \text{subG}(\sigma^2)$. $\boldsymbol{X}$ is a random matrix whose rows are following sub-Gaussian distribution with a covariance matrix $\boldsymbol{\Sigma}$.

We first introduce the parameter estimation algorithm under differentially private federated settings with a trusted central server.

Algorithm 2: Differentially Private Sparse Linear Regression under Federated Learning

Input : Dataset $D_{\text{all}} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_{i\in[m]}$, number of machines m, number of samples on each machine n, step size η^0, privacy parameters ε, δ, noise scale B_0, number of iterations T, truncation level R, feasibility parameter C_0, sparsity s, initial value $\boldsymbol{\beta}^0$.

for t **from** 0 **to** $T-1$ **do**

Step 1:

On each local machine $i = 1, 2, \ldots, m$, calculate the local gradient

$$\boldsymbol{g}_i = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{X}_{ij}^\top \boldsymbol{\beta}^t - \Pi_R(y_{ij}))\boldsymbol{X}_{ij}.$$

Send the gradient $\boldsymbol{g}_i$ to the central server.

Step 2:

Compute $\boldsymbol{\beta}^{t+0.5} = \boldsymbol{\beta}^t - (\eta^0/m)\sum_{i=1}^m \boldsymbol{g}_i$ at the central server;

Compute $\boldsymbol{\beta}^{t+1} = \Pi_{C_0}\left(\text{NoisyHT}(\boldsymbol{\beta}^{t+0.5}, s, \frac{\varepsilon}{T}, \frac{\delta}{T}, \frac{\eta^0 B_0}{mn})\right)$ at the central server.

Step 3: Send the output $\boldsymbol{\beta}^{t+1}$ back to each local machine from the server.

end

Output: Return $\boldsymbol{\beta}^T$.

In Step 1 of Algorithm 2, the information computed on each local machine is transmitted to the central server. The second step involves calculations performed at the central server. Prior to sending the information back to the local machines, it undergoes privacy preservation through the application of the NoisyHT algorithm, as introduced in Algorithm 1. Subsequently, the local machine updates its estimation based on the information received from the central server.

We compare Algorithm 2 with Algorithm 4.2 in [8], which addresses the private estimation in linear regression under non-federated learning settings. Unlike the latter, our algorithm does not transmit all data points to the central server. Instead, we calculate the gradient updates locally on each machine and send only these local gradients to the server. This design enhances privacy protection, as the original data remains visible only on the local machine and is not exposed externally. Furthermore, this approach of gradient updates also reduces communication costs by transmitting only a d-dimensional vector from each local machine for the gradient update. Previous research has also considered non-private distributed methods for linear regression problems, such as [27, 44]. Our algorithm, however, ensures differential privacy. In practice, the sparsity level s can be determined using a private version of cross-validation, while other parameters may be pre-chosen based on our theoretical analysis.

4.2 Algorithms for Inference Problems

In this subsection, we focus on statistical inference problems in the homogeneous federated learning setting, such as constructing coordinate-wise confidence intervals for parameters and performing simultaneous inference. To begin, we develop a method for constructing coordinate-wise confidence intervals, for example, for the k-th index of $\boldsymbol{\beta}$, β_k. However, it is important to note that the output of Algorithm 2 is biased due to hard thresholding. To overcome this bias, we employ a de-biasing procedure, a common technique in high-dimensional statistics, as demonstrated in previous studies [24]. This procedure involves approximating the k-th column of the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ to construct confidence intervals for each β_k. Subsequently, we focus on obtaining an estimate of the precision matrix in a private manner.

Algorithm 3: Differentially Private Precision Matrix Estimation in Federated Learning

Input : Number of machines m, number of data points in each machine n, dataset $\boldsymbol{X}_i = (\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{in})$ for $i = 1, \ldots, m$, step size η^1, privacy parameters ε, δ, noise scale B_1, number of iterations T, feasibility parameter C_1, sparsity s, initial value $\boldsymbol{\Theta}_k^0$.

for t **from** 0 **to** $T-1$ **do**

  Step 1: On each local machine $i = 1, 2, \ldots, m$, calculate local gradient $\boldsymbol{g}_i = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{X}_{ij}\boldsymbol{X}_{ij}^\top \boldsymbol{\Theta}_k^t - \boldsymbol{e}_j$. Send the gradients $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_m)$ to the central server.

  Step 2:

    Compute $\boldsymbol{\Theta}_k^{t+0.5} = \boldsymbol{\Theta}_k^t - (\eta^0/m)\sum_{i=1}^{m} \boldsymbol{g}_i$ at the central server;

    Compute $\boldsymbol{\Theta}_k^{t+1} = \Pi_{C_1}\left(\text{NoisyHT}(\boldsymbol{w}_k^{t+0.5}, s, \frac{\varepsilon}{T}, \frac{\delta}{T}, \frac{\eta^1 B_1}{mn})\right)$ at the central server.

  Step 3: Send $\boldsymbol{\Theta}_k^{t+1}$ back to each local machine from the server.

end

Output: Return $\boldsymbol{\Theta}_k^T$.

The structure of Algorithm 3 is similar to Algorithm 2, as both adopt an iterative communication between the central server and the local machines; the information is initially transmitted from the local machines to the server, then, the central server performs calculations and use the NoisyHT algorithm (Algorithm 1) to ensure the privacy of the information. Subsequently, each local machine updates the gradient and progresses to the next iteration. The primary distinction between two algorithms lies in the computation of the gradient on each machine.

Denote the output of Algorithm 2 as $\hat{\boldsymbol{\beta}}$ and the output of Algorithm 3 as $\hat{\boldsymbol{\Theta}}_k$. Then the de-biased differentially private estimator of β_k is given by

$$\hat{\beta}_k^{\mathrm{u}} = \hat{\beta}_k + \frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{g}_i + E_k, \tag{4.1}$$

where $\boldsymbol{g}_i = (1/n)\sum_{j=1}^{n}(\boldsymbol{X}_{ij}^\top \hat{\boldsymbol{\beta}} - \Pi_R(y_{ij}))\boldsymbol{X}_{ij}$, and E_k is the injected random noise to

ensure privacy, following a Gaussian distribution $N(0, 8\Delta_1^2 \log(1.25/\delta)/(n^2 m^2 \epsilon^2))$, where $\Delta_1 = \sqrt{s} c_1 c_x R + s c_0 c_1 c_x^2$ with some constants c_0, c_1, c_x defined later.

The debiased estimator in (4.1) enables us to construct a differentially private confidence intervals. Although the variance σ of the error term $\boldsymbol{W}$ in the linear regression model is usually unknown, we can estimate σ from the data in a private manner. The estimation is based on the residual term between the response $\boldsymbol{Y}$ and the fitted value $\boldsymbol{X}\hat{\boldsymbol{\beta}}$. We summarize the method to estimate σ in the private federated learning setting in Algorithm 4.

Algorithm 4: Differentially Private Variance Estimation in Federated Learning

Input : Dataset $(\boldsymbol{X}_i, \boldsymbol{Y}_i)_{[i=1,2,\ldots,m]}$, privacy parameters ε, noise scale B_2, truncation level R, estimated parameter $\hat{\boldsymbol{\beta}}$ from Algorithm 2.

Step 1: On each machine $i = 1, 2, \ldots, m$, compute $\hat{W}_i = \|\Pi_R(\boldsymbol{Y}_i) - \boldsymbol{X}_i \hat{\boldsymbol{\beta}}\|_2^2 / n$ and send $\hat{W}_i$ to the central server.

Step 2: Generate a random variable E^{var}, where $E^{\text{var}} \sim N(0, 2B_2^2 \log(1.25/\delta)/\epsilon^2)$

Step 3: Compute $\hat{\sigma}^2$ such that $\hat{\sigma}^2 = \sum_{i=1}^m \hat{W}_i/m + E^{\text{var}}$ at the central server

Output: Estimated variance $\hat{\sigma}^2$.

When examining the convergence rates of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Theta}}_k$ in Theorem 3, we observe the crucial roles of the largest and smallest restricted eigenvalues of $\boldsymbol{\Sigma}$. Since these eigenvalues directly influence the construction of confidence intervals and cannot be directly obtained from the data, their private estimation becomes essential. Below, we outline an algorithm to estimate the largest restricted eigenvalue, $\mu_s(\Sigma)$. To estimate the smallest restricted eigenvalue, $\nu_s(\Sigma)$, the same algorithm can be used by modifying Step 4 from "argmax" to "argmin".

Algorithm 5: Differentially Private Restricted Eigenvalue Estimation in Federated Learning

Input : Number of machines m, dataset $(\boldsymbol{X}_i)_{[i=1,\ldots,m]}$, number of data points in each machine n, privacy parameters ε, noise scale B_3, number of vectors n_1.

Step 1: Sample n_1 d-dimensional, s-sparse unit vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n_1}$.

Step 2: On each machine, compute $t_{i,k} = (\boldsymbol{v}_k^T \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{v}_k)/n$ where $k = 1, 2 \ldots, n_1$ and send them on to the central server.

Step 3: Sample $\xi_1, \ldots, \xi_{n_1} \sim$ Laplace $(2B_3/\epsilon)$.

Step 4: Compute $k_{\max}$ such that $k_{\max} = \arg\max_k \sum_{i=1}^m t_{i,k}/m + \xi_k$.

Output: $\mu_s(\hat{\boldsymbol{\Sigma}}) = \sum_{i=1}^m t_{i,k_{\max}}/m + \xi$, where $\xi \sim$ Laplace$(2B_3/\epsilon)$ independently.

Based on Algorithms 4 and 5, we provide a constuction for coordinate-wise confidence intervals in Algorithm 6.

Algorithm 6: Differentially Private Coordinate-wise Confidence interval for β_k in Federated Learning

Input : Number of machines m, dataset $(\boldsymbol{X}_i, \boldsymbol{Y}_i)_{[i=1,\ldots,m]}$, number of data points in each machine n, privacy parameters ε, δ, truncation level R, sparsity s, estimators of parameters $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\Theta}}_k$ from Algorithms 2 and 3, constants Δ_1, γ.

Step 1: On each local machine $i = 1, 2, \ldots, m$, calculate local gradient $\boldsymbol{g}_i = \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{X}_{ij}^\top \hat{\boldsymbol{\beta}} - \Pi_R(y_{ij}))\boldsymbol{X}_{ij}$. Send the gradient $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_m)$ to the central server.

Step 2: Generate a random variable E from the Gaussian distribution $N(0, 8\Delta_1^2 \log(1.25/\delta)/n^2m^2\epsilon^2)$.

Step 3: Calculate de-biased estimation, $\hat{\beta}_k^{\mathrm{u}} = \hat{\beta}_k + \frac{1}{m}\sum_{i=1}^{m} \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{g}_i + E$

Step 4: Estimate $\hat{\sigma}$ from Algorithm 4 and $\hat{\mu}_s, \hat{\nu}_s$ from Algorithm 5.

Step 5: Calculate the confidence interval $J_k(\alpha)$.

$$J_k(\alpha) = \Bigg[\hat{\beta}_k^{\mathrm{u}} - \frac{\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2} - \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{mn}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2\log(1/\delta)}{mn\epsilon^2}},$$

$$\hat{\beta}_k^{\mathrm{u}} + \frac{\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2} + \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{mn}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2\log(1/\delta)}{mn\epsilon^2}}\Bigg]$$

Output: Return the final result $J_k(\alpha)$.

So far we focused on constructing confidence intervals for individual coordinates of the parameter vector $\boldsymbol{\beta}$. However, in high-dimensional settings, we are often interested in group inference problem, where we test hypotheses involving multiple coordinates simultaneously. Specifically, we consider the problem of testing the null hypothesis given by

$$H_0 : \hat{\beta}_k = \beta_k, \text{ for all } k \in G$$

against the alternative hypothesis,

$$H_1 : \hat{\beta}_k \neq \beta_k,$$

for at least one $k \in G$, where G is a subset of all coordinates $\{1, 2, \ldots, d\}$ and we allow $|G|$ to be the same order as d. Additionally, we also construct simultaneous confidence intervals for all coordinates in G. Note that the problem discussed above are common in high-dimensional data analysis, with applications such as multi-factor analysis of variance [22], additive modeling [40]. Previous research works have discussed similar problems in the non-private setting, including [10, 42, 41].

To address the problem, simultaneous inference can be conducted using a test statistic

$$\max_{k\in G} |\hat{\beta}_k^{\mathrm{u}} - \beta_k|.$$

Major challenges of simultaneous inference in a private federated learning setting include: (1) minimizing the communication cost from local machines to the server while

retaining all data on the local machines, and (2) ensuring the privacy of the procedure, which necessitates a tailored privacy-preserving mechanism at each step of the algorithm.

In our framework, we propose an algorithm based on the bootstrap method. As previously mentioned, to build confidence intervals, our interest lies in the statistic computed by the maximum coordinate of $\hat{\beta}_k^{\mathrm{u}} - \beta_k$ over G. By decomposing this statistic, we obtain a term $\frac{\sigma}{\sqrt{mn}} \sum_{i=1}^{m} \sum_{j=1}^{n} \hat{\boldsymbol{\Theta}} \boldsymbol{X}_{ij} (\mathbf{y}_{ij} - \boldsymbol{X}_{ij}^T \boldsymbol{\beta})$. To determine the distribution of this term, we bootstrap the residuals $\mathbf{y}_{ij} - \boldsymbol{X}_{ij}^T \boldsymbol{\beta}$.

We outline the algorithm as follows: we first estimate $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Theta}}_k$ using Algorithm 2 and 3, respectively. Accordingly, by stacking $\hat{\boldsymbol{\Theta}}_k$ for all k, we get an estimator of the precision matrix $\hat{\boldsymbol{\Theta}}$. The details are provided in Algorithm 7.

Algorithm 7: Private Bootstrap Method for Simultaneous Inference in Federated Learning

Input : number of machines m, dataset $(\boldsymbol{X}_i, \boldsymbol{Y}_i)_{[i=1,2,\ldots m]}$, number of data on each machine n, privacy parameters ε, δ, estimators of parameters $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\Theta}}$ from Algorithms 2 and 3, number of iterations for bootstrap q, quantile α, noise level B_4, subset of coordinates G.

1 **for** t **from** 0 **to** q **do**

2 **Step 1:** For each local machine $i = 1, \ldots, m$, generate n independent standard Gaussian random variables $e_{i1}, \ldots, e_{in}$. Calculate $\boldsymbol{u}_i = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \hat{\boldsymbol{\Theta}} \boldsymbol{X}_{ij} e_{ij}$.

3 **Step 2:** Send $(\boldsymbol{u}_i)_{[i=1,2,\ldots,m]}$ from local machines to the central server.

4 **Step 3:** Calculate $U_t = \text{Privatemax}([(1/\sqrt{m}) \sum_{i=1}^{m} \boldsymbol{u}_i]_G, \epsilon, \delta, B_4)$ at the central server.

5 **end**

6 **Output:** Compute the α-quantile $C_U(\alpha)$ of $(|U_1|, |U_2|, \ldots |U_q|)$ for $\alpha \in (0, 1)$.

On line 4 of Algorithm 7, we employ the Private Max algorithm, which we mentioned earlier as a variation of NoisyHT algorithm (Algorithm 1) by directly picking $s = 1$, to obtain the maximum element in a vector in a private manner. It is also important to note that the Private Max algorithm is applied to a subset of G. After presenting the algorithm, we denote M as:

$$M(\hat{\boldsymbol{\beta}}) = \max_{k \in G} |\sqrt{mn}(\hat{\beta}_k^{\mathrm{u}} - \beta_k)|.$$

M is used as the statistic for inference problems later.

As previously mentioned, we can easily construct a simultaneous confidence interval for each $k \in G$ by:

$$\left[\hat{\beta}_k^{\mathrm{u}} - \frac{\hat{\sigma}}{\sqrt{mn}} C_U(\alpha), \quad \hat{\beta}_k^{\mathrm{u}} + \frac{\hat{\sigma}}{\sqrt{mn}} C_U(\alpha)\right],$$

where $C_U(\alpha)$ is obtained from our algorithm with prespecified α. We can similarly perform hypothesis testing; first calculate the test statistic and obtain $C_U(\alpha)$ from our algorithm with prespecified α, then reject if the statistic lies in the rejection region.

4.3 Theoretical Results

In this subsection, we provide theoretical guarantee for the algorithms and methods discussed in the previous subsections. Before proceeding, we outline key assumptions concerning the design matrix $\boldsymbol{X}$, precision matrix $\boldsymbol{\Theta}$, and the true parameter $\boldsymbol{\beta}$ of the linear regression model, which are essential for our subsequent analyses.

(P1) Parameter Sparsity: The true parameter vector $\boldsymbol{\beta}$ satisfies $\|\boldsymbol{\beta}\|_2 < c_0$ for some constant $0 < c_0 < \infty$ and $\|\boldsymbol{\beta}\|_0 \leq s_0^* = o(n)$.

(P2) Precision matrix sparsity: For each column of the precision matrix $\boldsymbol{\Theta}_k$, $k = 1, 2, \ldots, d$, it satisfies that $\|\boldsymbol{\Theta}_k\|_2 < c_1$ for some constant $0 < c_1 < \infty$ and $\|\boldsymbol{\Theta}_k\|_0 \leq s_1^* = o(n)$.

(D1) Design Matrix: for each row of the design matrix $\boldsymbol{X}$, denote by $\boldsymbol{x}$, $\boldsymbol{x}\boldsymbol{\Sigma}^{-1/2}$ is sub-Gaussian with sub-Gaussian norm $\kappa := \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_{\psi_2}$.

(D2) Bounded Eigenvalues of the covariance matrix: For the covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}\boldsymbol{x}\boldsymbol{x}^\top$, there exists a constant $0 < L < \infty$ such that $0 < 1/L < \lambda_{\min}(\boldsymbol{\Sigma}) \leq \lambda_{\max}(\boldsymbol{\Sigma}) < L$.

The above assumptions (P1) and (P2) bounds the ℓ_2 norm and ℓ_0 norm of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\Theta}_k$, and assumption (D1) guarantees that each row of X follows a sub-Gaussian distribution, and assumption (D2) requires the covariance matrix has bounded eigenvalues. These assumptions are commonly used for theoretical analysis of differentially private algorithms and debiased estimators [9, 8, 24].

With assumptions (P1)-(D2), we analyze the algorithms we presented. We begin with the estimation problem and provide a rate of convergence of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\Theta}}_k$.

Theorem 3 *Let $\{(y_{ij}, \boldsymbol{X}_{ij})\}_{i\in[m],j\in[n]}$ be an i.i.d. samples from the high-dimensional linear model. Suppose that assumptions (P1), (P2), (D1), (D2) are satisfied. Additionally,*

- *we choose parameters as follows: let $s^* = max(s_0^*, s_1^*)$, $R = \sigma\sqrt{2\log mn}$, $C_0 = c_0$, $C_1 = c_1$, $c_x = 3\sqrt{2L\kappa^2\log d}$, $B_0 = 2(R + \sqrt{s}c_0c_x)c_x$, $B_1 = 2\sqrt{s}c_1c_x^2$, $\Delta_1 = \sqrt{s}c_1c_xR + sc_0c_1c_x^2$ and $\gamma = \max(\mu_s(9\mu_s + 1/4), 17/16\mu_s + 1/96)$, where μ_s, ν_s are the largest and smallest s-restricted eigenvalues of $\hat{\Sigma}$.*
- *we set $\boldsymbol{\beta}^0 = \mathbf{0}$ and $\boldsymbol{\Theta}_k^0 = \mathbf{0}$ as the initialization used in Algorithm 2 and Algorithm 3.*

Then there exists some absolute constant $\rho > 0$ such that, if $s = \rho L^4 s^*$, $\eta^0 = \eta^1 = s/6L$, $T = \rho L^2\log\big(8c_0^2Ln\big)$ and $n \geq KR(s^*)^{3/2}\log d\sqrt{\log(1/\delta)}\log n/\varepsilon$ for a sufficiently large constant $K > 0$, then, for the output from Algorithm 2 and Algorithm 3,

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 \leq \sigma^2\left(k_0 \cdot \frac{s\log d}{mn} + \frac{6\gamma\mu_s}{\nu_s^2} \cdot \frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2}\right), \tag{4.2}$$

and

$$\|\hat{\mathbf{\Theta}}_k - \mathbf{\Theta}_k\|_2^2 \leq \sigma^2 \bigg(k_1 \cdot \frac{s \log d}{mn} + \frac{6\gamma\mu_s}{\nu_s^2} \frac{s^2 \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} \bigg), \tag{4.3}$$

hold with probability $1 - \exp(-\Omega(\log(d/s \log d) + \log n))$.

The upper bound of Algorithm 3 in (4.3) can be interpreted as follows. The first term represents the statistical error, while the second term accounts for the privacy cost. Furthermore, the result is comparable to that of Theorem 4.4 in [8], which addresses private linear regression in a non-federated setting. This comparison suggests that the federated learning approach does not affect the convergence rate adversely; instead, it allows us to leverage the benefits of federated learning. We also note that the advantages of federated learning will be further explored in the heterogeneous federated learning setting, which will be discussed in the next chapter.

The remainder of this subsection presents the theoretical results for the inference problem. We begin with the construction of coordinate-wise confidence intervals. As mentioned before, σ is usually unknown and we estimate σ in a private manner, presented in Algorithm 4. Lemma 4.1 states the statistical guarantee of our algorithm.

Lemma 4.1 *Let* $\hat{\sigma}^2$ *be the output from Algorithm 4 by choosing* $R = O(\sqrt{2 \log mn})$, $B_2 = \frac{4}{mn}(R^2 + s^2 c_0^2 c_x^2)$ *and* $\hat{\boldsymbol{\beta}}$ *as the output from Algorithm 2. Then, Algorithm 4 is* (ϵ, δ)*-differentially private, and it follows that*

$$|\sigma^2 - \hat{\sigma}^2| \leq c \cdot \bigg(\frac{1}{\sqrt{mn}} + \frac{s \log d}{mn} + \frac{s^2 \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} \bigg),$$

where $c > 0$ *is a universal constant.*

Next, we consider a simplified version of the confidence interval, where the privacy cost is dominated by the statistical error. In this scenario, we assume that the privacy level is relatively low and the privacy constraints are loose, meaning that the privacy parameters ϵ and δ are relatively large, allowing for nearly cost-free estimation. We present our result in the following theorem.

Theorem 4 *Suppose that the conditions in Theorem 3 hold. Assume that* $\frac{s^* \log d}{\sqrt{mn}} = o(1)$ *and* $\frac{s^{*2} \log^2 d \log(1.25/\delta) \log^3 mn}{mn\epsilon^2} = o(1)$. *Also assume that the privacy cost is dominated by statistical error, i.e., there exists a constant* k_0 *such that* $\frac{s^{*2} \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} \leq k_0 \cdot \frac{s^* \log d}{mn}$. *Then, given the de-biased estimator* $\hat{\beta}_k^u$ *defined in* (4.1)*, the confidence interval is asymptotically valid:*

$$\lim_{mn \to \infty} \mathbb{P}(\beta_k \in J_k(\alpha)) = 1 - \alpha,$$

where

$$J_k(\alpha) = \bigg[\hat{\beta}_k^u - \Phi^{-1}(1 - \alpha/2) \frac{\hat{\sigma}}{\sqrt{mn}} \sqrt{\hat{\mathbf{\Theta}}_k^\top \hat{\mathbf{\Sigma}} \hat{\mathbf{\Theta}}_k}, \quad \hat{\beta}_k + \Phi^{-1}(1 - \alpha/2) \frac{\hat{\sigma}}{\sqrt{mn}} \sqrt{\hat{\mathbf{\Theta}}_k^\top \hat{\mathbf{\Sigma}} \hat{\mathbf{\Theta}}_k} \bigg]$$

Also, the confidence interval $J_k(\alpha)$ *is* (ϵ, δ)*-differentially private.*

Theorem 4 assumes that the privacy cost is dominated by the statistical error. However, when the privacy constraint is more stringent with small privacy parameters ϵ and δ, the privacy cost may be larger than the statistical error. In this scenario, we generalize Theorem 4 to analyze Algorithm 6. We note that the largest and smallest restricted eigenvalues of $\hat{\mathbf{\Sigma}}$ also need to be estimated by Algorithm 5. Lemma 4.2 quantifies the estimation error of the largest restricted eigenvalue of $\hat{\mathbf{\Sigma}}$.

Lemma 4.2 *If $n_1 = cd^s$ and $B_3 = 2sc_x^2/n$ for some constant $c > 0$, then the output from Algorithm 5 is $(\epsilon, 0)$-differentially private. Moreover, $(1/9)\lambda_s \leq \hat{\lambda}_s \leq \lambda_s$ holds where λ_s is the largest restricted eigenvalue of $\hat{\mathbf{\Sigma}}$.*

We then present a theoretical result for the confidence interval in a more general case in Theorem 5.

Theorem 5 *Assume the conditions in Theorem 3 hold. Suppose that $\frac{s^* \log d}{\sqrt{mn}} = o(1)$ and $\frac{s^{*2} \log^2 d \log(1.25/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} = o(1)$, then, given the de-biased estimator $\hat{\beta}_k^u$ defined in (4.1) and the estimated restricted eigenvalues $\hat{\mu}_s$ and $\hat{\nu}_s$ from Algorithm 5, the confidence interval constructed by Algorithm 6 is asympotically valid:*

$$\lim_{mn \to \infty} \mathbb{P}(\beta_k \in J_k(\alpha)) = 1 - \alpha,$$

where

$$J_k(\alpha) = \Bigg[\hat{\beta}_k^u - \frac{\gamma \hat{\mu}_s^2}{\hat{\nu}_s^2} \frac{s^2 \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} - \Phi^{-1}(1 - \alpha/2) \frac{\hat{\sigma}}{\sqrt{mn}} \sqrt{\hat{\mathbf{\Theta}}_k^\top \hat{\mathbf{\Sigma}} \hat{\mathbf{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{mn\epsilon^2}},$$
$$\hat{\beta}_j + \frac{\gamma \hat{\mu}_s^2}{\hat{\nu}_s^2} \frac{s^2 \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} + \Phi^{-1}(1 - \alpha/2) \frac{\hat{\sigma}}{\sqrt{mn}} \sqrt{\hat{\mathbf{\Theta}}_k^\top \hat{\mathbf{\Sigma}} \hat{\mathbf{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{mn\epsilon^2}}\Bigg].$$

Also, $J_k(\alpha)$ is (ϵ, δ)-differentially private.

Compared to the non-private counterpart in [24], we claim that our confidence interval has a similar form but with additional noise injected to ensure privacy. When the noise level is low, the confidence interval closely approximates the non-private counterpart, allowing us to nearly achieve privacy without incurring additional costs. Furthermore, when the privacy level is high, the confidence interval has a larger length to attain the same confidence level.

Finally, for the simultaneous inference problems, we demonstrate that α-quantile of statistic M in (6) is close to the α-quantile of U calculated in Algorithm 7 for each $\alpha \in (0, 1)$ using the bootstrap method. The next theorem states the statistical properties of Algorithm 7.

Theorem 6 *Assume the conditions in Theorem 4 hold. Additionally, we assume that $\frac{s^* \log d}{\sqrt{mn}} = o(1)$ and the privacy cost is dominated by the statistical error, i.e., there exists a constant $c > 0$ such that $\frac{s^{*2} \log^2 d \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} \leq c \cdot \frac{s^* \log d}{mn}$. We also assume that*

there exists a constant k_0 *such that* $\log^7(dmn)/mn \leq \frac{1}{(mn)^{k_0}}$*, and that* $q = o(mn)$*, where* q *is the number of iterations for bootstrap* q*. The noise level is chosen as* $B_4 = 4L\sqrt{\log mc_x}(R + c_0c_x\sqrt{s^*})/\sqrt{mn}$*. Then,* $C_U(\alpha)$ *computed in Algorithm 7 satisfies*

$$\sup_{\alpha\in(0,1)} |\mathbb{P}(M \leq C_U(\alpha)) - \alpha| = o(1).$$

Theorem 6 has useful applications: we can obtain a good estimator of the α-quantile of U using the bootstrap method and then use it to construct confidence intervals or perform hypothesis testing. Numerical results will be presented in later chapters to further support our claims.

5 Heterogeneous Federated Learning Setting

5.1 Methods and Algorithms

In this section, we consider a more general setting where the parameters of interest on each machine are not identical, but they share some similarities. Specifically, we consider the scenario where, on each machine $i = 1, 2, \ldots, m$, we assume a linear regression model:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^{(i)} + \boldsymbol{W}_i,$$

where $\boldsymbol{\beta}^{(i)}$ represents the true parameter on machine i. We assume that each $\boldsymbol{W}_i$ is a vector whose coordinates follow a sub-Gaussian distribution: $\boldsymbol{W}_{ik} \sim \text{subG}(\sigma^2)$, $k = 1, 2 \ldots, d$ i.i.d. We also assume that each row of $\boldsymbol{X}$ follows a sub-Gaussian distribution i.i.d. with mean zero and covariance matrix $\boldsymbol{\Sigma}$. We further quantify the similarity of each $\boldsymbol{\beta}^{(i)}$ by assuming that that there exists a subset $\mathcal{S} \in \{1, 2, \ldots, d\}$ with $|\mathcal{S}| = s_0$ satisfying $\boldsymbol{\beta}_{\mathcal{S}}^{(i_1)} = \boldsymbol{\beta}_{\mathcal{S}}^{(i_2)}$ for any $i_1, i_2 \in \{1, 2, \ldots, m\}$.

A naive approach would be estimating each $\boldsymbol{\beta}^{(i)}$ locally, as in the non-private setting. However, in the context of federated learning, we can improve the estimation with a sharper rate of convergence by exploiting similarities of the model across machines. To achieve this, we decompose $\boldsymbol{\beta}^{(i)}$ into the sum of two vectors, $\boldsymbol{\beta}^{(i)} = \boldsymbol{u} + \boldsymbol{v}_i$, where $\boldsymbol{u}$ captures the signals common to all $\boldsymbol{\beta}^{(i)}$, and $\boldsymbol{v}_i$ captures the signals unique to each machine.

We employ a two-stage procedure to estimate each $\boldsymbol{\beta}^{(i)}$: in the first stage, we estimate $\boldsymbol{u}$ using Algorithm 2 with a sparsity level of $\|\boldsymbol{u}\|_0 = s_0$ indicating the number of shared signals. In the second stage, we estimate $\boldsymbol{v}_i$ on the individual machine. Our final estimation of $\boldsymbol{\beta}^{(i)}$ is given by $\hat{\boldsymbol{\beta}}^{(i)} = \hat{\boldsymbol{v}}_i + \hat{\boldsymbol{u}}$. The procedure is summarized in Algorithm 8.

Similar to the previous section, we next address inference problems. Our algorithms consist of two parts: the construction of coordinate-wise confidence intervals and simultaneous inference. We begin by describing the algorithm for coordinate-wise confidence intervals in Algorithm 9.

In Algorithm 9, $\hat{\boldsymbol{\Theta}}_j$ is the (ϵ, δ)-differentially private estimator of the j-th row of the precision matrix of covariance matrix $\hat{\Sigma} = 1/(mn)\sum_{i=1}^m \sum_{j=1}^n \boldsymbol{X}_{ij}\boldsymbol{X}_{ij}^\top$. We define the

Algorithm 8: Differentially Private Sparse Linear Regression in Heterogeneous Federated Learning Setting

Input : Number of machines m, dataset $(\boldsymbol{y}_i, \boldsymbol{X}_i)_{[i=1,\ldots,m]}$, number of data on each machine n, step size η^0, privacy parameters ε, δ, noise scale B_5, number of iterations T, truncation level R, feasibility parameter C_0, initial value $\boldsymbol{v}_i^0$, sparsity level of similar vector s_0, sparsity level s.

1 **Step 1:** Estimate a s_0 sparse vector $\boldsymbol{u}$ using Algorithm 2.

2 **Step 2:** Estimate a $s_1 := s - s_0$ sparse vector $\boldsymbol{v}_i$ with samples $(\boldsymbol{y}_i, \boldsymbol{X}_i)$ on machine i with the following iterations from line 3-6.

3 **for** t **from** 0 **to** $T-1$ **do**

4 Compute $\boldsymbol{v}_i^{t+0.5} = \boldsymbol{v}_i^t - (\eta^0/n)\sum_{i=1}^n (\boldsymbol{x}_i^\top \boldsymbol{v}_i^t - \Pi_R(y_i - \boldsymbol{x}_i^\top \boldsymbol{u}))\boldsymbol{x}_i$;

5 $\boldsymbol{v}_i^{t+1} = \Pi_{C_0}\big(\text{NoisyHT}(\boldsymbol{v}_i^{t+0.5}, (\boldsymbol{y}_i, \boldsymbol{X}_i), s, \varepsilon/T, \delta/T, \eta^0 B_5/n)\big)$.

6 **end**

7 **Step 3:** Estimate $\boldsymbol{\beta}^{(i)}$ by $\hat{\boldsymbol{\beta}}^{(i)} := \hat{\boldsymbol{v}}_i + \hat{\boldsymbol{u}}$.

Output: $\hat{\boldsymbol{\beta}}^{(i)}$.

Algorithm 9: Differentially Private Coordinate-wise Confidence interval for β_k in Heterogeneous Federated Learning

Input : Number of machines m, dataset $(\boldsymbol{X}_i, \boldsymbol{Y}_i)_{[i=1,\ldots,m]}$, number of data points in each machine n, privacy parameters ε, δ, truncation level R, sparsity s, estimated parameters $\hat{\boldsymbol{\beta}}^{(i)}$, $\hat{\boldsymbol{\Theta}}_k$ from Algorithms 8 and 3, and estimated eigenvalues $\hat{\mu}_s$, $\hat{\nu}_s$ from Algorithm 5, constants Δ_1, γ.

1 **Step 1:** Generate a random variable E_3 from a Gaussian distribution $N(0, 8\Delta_1^2 \log(1.25/\delta)/(n^2\epsilon^2))$.

2 **Step 2:** Calculate de-biased estimation,
$\hat{\beta}_k^{(i,\mathrm{u})} = \hat{\beta}_k^{(i)} + \frac{1}{n}\sum_{j=1}^n (\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}_{ij} \Pi_R(\boldsymbol{y}_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}_{ij} \boldsymbol{X}_{ij}^\top \hat{\beta}_k^{(i)}) + E_3$.

3 **Step 3:** Calculate the confidence interval $J_k(\alpha)$.

$$J_k(\alpha) = \Bigg[\hat{\beta}_k^{(i,\mathrm{u})} - a - \frac{\sigma\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n\epsilon^2}},$$
$$\hat{\beta}_k^{(i,\mathrm{u})} + a + \frac{\sigma\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n\epsilon^2}}\Bigg],$$

where a is defined in (5.1).

4 **Output:** Return the final result $J_k(\alpha)$.

variable a in step 3 by

$$a := \frac{2\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2 n^2 \epsilon^2} + \frac{2\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^2\epsilon^2}. \tag{5.1}$$

We then provide Algorithm 10 for the simultaneous inference problem. Similar to the

previous chapter, we can perform simultaneous inference for each $\boldsymbol{\beta}^{(i)}$ to build simultaneous confidence interval and hypothesis testing.

Algorithm 10: Private Bootstrap Method for Simultaneous Inference in Heterogeneous Federated Learning for Machine $i \in \{1, \ldots, m\}$

Input : Dataset $(\boldsymbol{y}_i, \boldsymbol{X}_i)$, number of data n, privacy parameters ε, δ, estimators of parameters $\hat{\boldsymbol{\beta}}^{(i)}$, $\hat{\boldsymbol{\Theta}}$ from Algorithms 8 and 3, number of iterations for Bootstrap q, quantile α, noise level B_6. (RN: σ?)

for t **from** 0 **to** q **do**

  Generate n independent standard Gaussian random variables $e_1, \ldots, e_n$.

  Calculate $U_t = \|\text{Privatemax}([\frac{\sigma}{\sqrt{n}} \sum_{j=1}^{n} \hat{\Theta} \boldsymbol{X}_{ij} e_j]_G, \epsilon, \delta, B_6)\|_\infty$

end

Output: Compute the α-quantile $C_U(\alpha)$ of $(|U_1|, |U_2|, \ldots, |U_q|)$ for $\alpha \in (0, 1)$.

Compared with Algorithm 7 introduced for simultaneous inference in homogeneous federated learning, bootstrap algorithm in Algorithm 10 runs within the local machine of interest. Using the output from Algorithm 10, we build a simultaneous confidence interval for each $\beta_k^{(i)}$ $(k \in G)$ using $C_U(\alpha)$ by

$$\left[\hat{\beta}_k^{(i,\mathrm{u})} - \frac{1}{\sqrt{n}} C_U(\alpha), \quad \hat{\beta}_k^{(i,\mathrm{u})} + \frac{1}{\sqrt{n}} C_U(\alpha)\right].$$

5.2 Theoretical Results

In this subsection, we provide theoretical analysis for the algorithms in heterogeneous federated learning settings. We begin our theoretical analysis with the estimation problem, which resembles Theorem 3.

Intuitively, when $\boldsymbol{\beta}^{(i)}$ are similar but not identical, federated learning can be used to estimate their common elements and the remaining parameters can be estimated individually on each machine. This results in a sharper rate of convergence as the estimation of the common component $\boldsymbol{u}$ can exploit the information from more data points. We summarize the result in Theorem 7.

Theorem 7 *Assume that the conditions in Theorem 5 hold. Further assume that for Algorithm 8,* $\|\boldsymbol{v}_i\|_0 = s_1 = s - s_0$ *for all* $i = 1, \ldots, m$, $\|\boldsymbol{u}\|_0 = s_0$, $\|\boldsymbol{u}\|_2 \leq c_0/2$, *and* $\|\boldsymbol{v}_i\|_2 \leq c_0/2$. *Let* $B_5 = c_x(2R + \sqrt{s_1} c_0 c_x)$. *Then, for the output* $\hat{\boldsymbol{\beta}}^{(i)}$ *from Algorithm 8, we have*

$$\|\hat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^{(i)}\|_2^2 \leq c_0 \frac{s_0 \log d}{mn} + c_1 \frac{{s_0}^2 \log d^2 \log(1/\delta) \log^3 mn}{m^2 n^2 \epsilon^2} + c_2 \frac{s_1 \log d}{n} + c_3 \frac{{s_1}^2 \log d^2 \log(1/\delta) \log^3 n}{n^2 \epsilon^2}, \tag{5.2}$$

where $c_0, c_1, c_2, c_3 > 0$ *are some constants.*

In the case where $s_0 \ll s_1$, i.e., the models are largely different across machines, the third and fourth term on the right hand side of (5.2) dominates the estimation error, and the estimation accuracy of $\boldsymbol{\beta}^{(i)}$ via federated learning becomes closer to that with a single

machine ($m = 1$). In high level, this is because the information from other machines is not helpful in the estimation when there exists a large dissimilarity of models across machines. However, with a large $s_0 \gg s_1$, federated learning can leverage the similarity of models to improve estimation accuracy. As a result, the rate in (5.2) becomes closer to the rate in 4.2 for homogeneous federated learning setting when $s_0/s_1 \to 0$.

We next present our results for the inference problems. To start we verify that the output from Algorithm 9 is a asymptotic $1-\alpha$ confidence interval for $\beta_k^{(i)}$.

Theorem 8 *Assume the conditions in Theorem 3 hold and assume that* $\frac{s^* \log d}{\sqrt{n}} = o(1)$ *and* $\max(\frac{2\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2}, \frac{2\gamma\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^2\epsilon^2}) = o(1)$. *Let a be the variable defined in* (5.1). *Then, for the de-biased estimator $\hat{\beta}_k^{(i,u)}$ defined in* (4.1), *the constructed confidence interval is asympotically valid:*

$$\lim_{n\to\infty} \mathbb{P}(\beta_k^{(i)} \in J_k(\alpha)) = 1-\alpha,$$

where

$$J_k(\alpha) = \Bigg[\hat{\beta}_k^{(i,u)} - a - \frac{\hat{\sigma}\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2\log(1/\delta)}{n\epsilon^2}},$$
$$\hat{\beta}_k^{(i,u)} + a + \frac{\hat{\sigma}\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2\log(1/\delta)}{n\epsilon^2}}\Bigg]$$

Also, $J_k(\alpha)$ is (ϵ, δ)-differentially private.

Finally, we provide a statistical guarantee for Algorithm 10. Similar to the previous section, we define M as:

$$M = M(\hat{\boldsymbol{\beta}}^{(i,\mathrm{u})}) = \max_{k\in G} |\sqrt{n}(\hat{\beta}_k^{(i,\mathrm{u})} - \beta_k^{(i)})|.$$

Theorem 9 *Assume that the conditions in Theorem 4 hold. We additionally assume that* $\frac{s^* \log d}{\sqrt{n}} = o(1)$ *and the privacy cost is dominated by the statistical error, i.e., there exists a constant c such that* $\frac{s^{*2}\log^2 d \log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2} \le c\cdot\frac{s^*\log d}{mn}$ *and* $\frac{s^{*2}\log^2 d \log(1/\delta)\log^3 n}{n^2\epsilon^2} \le c\cdot\frac{s^*\log d}{n}$. *We also assume that there exists a constant k_0 such that* $\log^7(dn)/n \le \frac{1}{n^{k_0}}$. *The noise level is chosen as* $B_6 = 2\sqrt{\frac{s\log n}{n}}c_x c_1$. *Then,*

$$\sup_{\alpha\in(0,1)} |\mathbb{P}(M \le C_U(\alpha)) - \alpha| = o(1).$$

Theorem 9 states that α-quantile of M is asymptotically close to $C_U(\alpha)$, which validates the $1-\alpha$ simultaneous confidence intervals based on $C_U(\alpha)$ obtained by the bootstrap method. This result allows us to perform simultaneous inference such as the confidence intervals and hypothesis testing based on $C_U(\alpha)$.

6 Simulations

In this section, we conduct simulations to investigate the performance of our proposed algorithm as discussed in the preceding sections. Specifically, we explore the more complex heterogeneous federated learning setting, where each machine operates on different models yet exhibits similarities. Our simulations are divided into three main parts.

In Section 6.1, we present the simulation results for the coordinate-wise estimation problem within a private federated setting, discussing the differences between the estimated $\hat{\boldsymbol{\beta}}$ and the true $\boldsymbol{\beta}^*$ across various scenarios. We also examine the coverage of our proposed confidence intervals. Section 6.2 extends the settings to simultaneous inference.

We generate simulateion simulation datasets as follows. First, we sample the data $\boldsymbol{X}_i$, for $i = 1, 2, \ldots, m$, where each $\boldsymbol{X}_i$ follows a Gaussian distribution with mean zero and covariance matrix $\boldsymbol{\Sigma}$. We set $\boldsymbol{\Sigma}$ such that for each $j, j' \in \{1, 2, \ldots, d\}$, $\Sigma_{j,j'} = 0.5^{|j-j'|}$. On each machine, we assume a s^*-sparse unit vector $\boldsymbol{\beta}^{(i)}$ with $s^* = s_0 + s_1$, where s_0 is the number of non-zero shared signals. For each $\boldsymbol{\beta}^{(i)}$, we set the first s_0 shared elements to $1/\sqrt{s^*}$ and additionally select machine-specific s_1 entries from the remaining $d - s_0$ indices to be $1/\sqrt{s^*}$. We then compute $\boldsymbol{Y}_i = \boldsymbol{X}_i\boldsymbol{\beta}^{(i)} + \boldsymbol{W}_i$, where each $\boldsymbol{W}_i$ follows a Gaussian distribution $N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ with $\sigma = 0.5$.

6.1 Estimation and Confidence Interval

In this subsection, we investigate the estimation accuracy and confidence interval coverage of our algorithm for coordinate-wise inference. Namely, we consider the following scenarios:

- Fix number of machines $m = 15$, $\epsilon = 0.8$, $\delta = 1/(2mn)$, $d = 800$, $s^* = 15$ and $s_0 = 6$. Set the number of samples on each machine to be $4000, 5000, 6000$, respectively.
- Fix number of samples on each machine $n = 4000$, $\epsilon = 0.8$, $\delta = 1/(2mn)$, $d = 800$, $s^* = 15$ and $s_0 = 6$. Set the number of machines m to be $5, 10, 15$,
- Fix number of machines $m = 15$, number of samples on each machine $n = 4000$, $\epsilon = 0.8$, $\delta = 1/(2mn)$, $d = 800$. Set, $s^* = 15, s_0 = 6$, $s^* = 10, s_0 = 4$, $s^* = 20, s_0 = 8$, respectively.
- Fix number of machines $m = 15$, number of samples on each machine $n = 4000$, $\epsilon = 0.8$, $\delta = 1/(2mn)$, $d = 800$, $s^* = 15$. Set $s_0 = 6, 8, 10$, respectively.
- Fix number of machines $m = 15$, number of samples on each machine $n = 4000$, $\delta = 1/(2mn)$, $d = 800$, $s^* = 15$ and $s_0 = 6$. Set $\epsilon = 0.3, 0.5, 0.8$ respectively.
- Fix number of machines $m = 15$, number of samples on each machine $n = 4000$, $\epsilon = 0.8$, $\delta = 1/(2mn)$, $s^* = 15$ and $s_0 = 6$. Set $d = 600, 800, 1000$, respectively.

For each setting, we report the average estimation error $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2$ among 50 replications. Also, in each setting, we calculate the confidence interval with $\alpha = 0.95$ for each index

of $\boldsymbol{\beta}^*$ using our proposed algorithm. To evaluate the quality of confidence interval, we define cov as the coverage of the confidence interval:

$$\text{cov} := d^{-1} \sum_{i=1}^{d} \mathbb{P}[\beta_i^* \in J_i(\alpha)].$$

We also define the coverage for non-zero and zero entries of $\boldsymbol{\beta}^*$ by $\text{cov}_{\mathcal{S}}$ and $\text{cov}_{\mathcal{S}^c}$, respectively, where $\mathcal{S}$ is the set of non-zero indices in $\boldsymbol{\beta}^*$.

$$\text{cov}_{\mathcal{S}} = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \mathbb{P}[\beta_i^* \in J_i(\alpha)] \quad , \quad \text{cov}_{\mathcal{S}^c} = |\mathcal{S}^c|^{-1} \sum_{i \in \mathcal{S}^c} \mathbb{P}[\beta_i^* \in J_i(\alpha)].$$

We report the estimation error, coverage of true parameter and length of confidence interval for each configuration listed above in Table 2:

Simulation Results					
$(n, m, d, s^*, s_0, \epsilon)$	Estimation Error (Sd)	cov	$\text{cov}_{\mathcal{S}}$	$\text{cov}_{\mathcal{S}^c}$	length
(3000,15,800,15,8,0.8)	0.0213 (0.0028)	0.940	0.929	0.940	0.0532
(4000,15,800,15,8,0.8)	0.0170 (0.0032)	0.945	0.960	0.944	0.0437
(5000,15,800,15,8,0.8)	0.0141 (0.0021)	0.940	0.945	0.940	0.0378
(4000,10,800,15,8,0.8)	0.0218 (0.0047)	0.945	0.945	0.945	0.0437
(4000,20,800,15,8,0.8)	0.0126 (0.0025)	0.944	0.941	0.944	0.0437
(4000,15,600,15,8,0.8)	0.0162 (0.0031)	0.946	0.933	0.946	0.0436
(4000,15,1000,15,8,0.8)	0.0191 (0.0027)	0.940	0.933	0.940	0.0439
(4000,15,800,15,4,0.8)	0.0188 (0.0032)	0.952	0.945	0.953	0.0420
(4000,15,800,15,12,0.8)	0.0137 (0.0016)	0.944	0.937	0.944	0.0462
(4000,15,800,10,8,0.8)	0.0105 (0.0017)	0.946	0.947	0.946	0.0389
(4000,15,800,20,8,0.8)	0.0243 (0.0036)	0.941	0.932	0.941	0.0497
(4000,15,800,15,8,0.5)	0.0240 (0.0038)	0.940	0.949	0.940	0.0550
(4000,15,800,15,8,0.3)	0.0943 (0.0281)	0.928	0.941	0.928	0.0792

Table 2: Table for Simulation Results of the private federated linear regression

From Table 2, we observe a consistent result with our theory. Namely, for the estimation error, the error becomes small as ϵ gets larger as we require less level of privacy.

Also, more data points on each machine, more number of machines, smaller sparsity level lead to better estimation accuracy. For confidence intervals, we observe that the coverage is close to 0.95 for cov, $\text{cov}_{\mathcal{S}}$, and $\text{cov}_{\mathcal{S}^c}$, is and stable in different settings. To further illustrate our claim, we pick the setting of $(n, m, d, s, s_0, \epsilon) = (4000, 15, 800, 15, 8, 0.8)$ and plot the confidence intervals versus the true value among 50 replications in Figure 3. We randomly select 60 out of 800 coordinates.

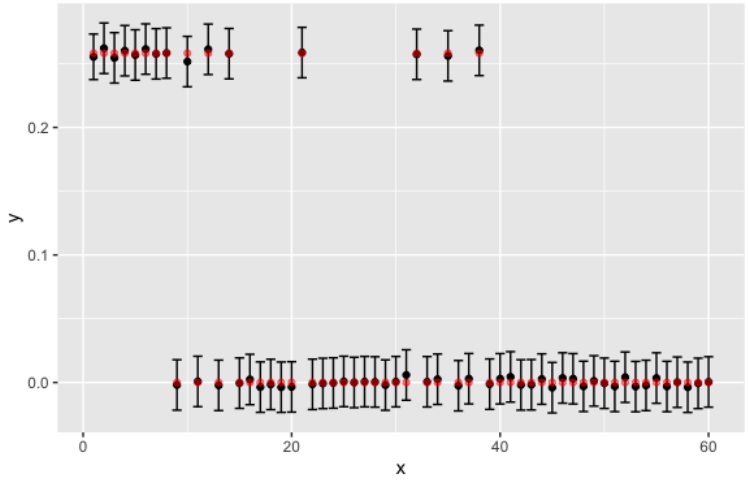

Fig 3: Confidence intervals for β_k for each coordinate k randomly selected from 800 coordinates. vertical axis stands for the value of β_k. Red points stand for the true β_k while black points stand for the estimated β_k. We mention that the result averaged over 50 iterations.

We also summarize our results in Figure 6.1, where we plot the estimation error against the change in the number of samples, sparsity, and number of machines. For the figure, we fixed $m = 15$, $d = 800$, $s^* = 16$, $s_0 = 8$, for the middle figure, we fixed $n = 4000$, $m = 15$, $d = 800$, $\epsilon = 0.5$, and for the right figure, we fixed $d = 800$, $s^* = 16$, $s_0 = 8$, $\epsilon = 0.5$. The error is averaged over 200 replications.

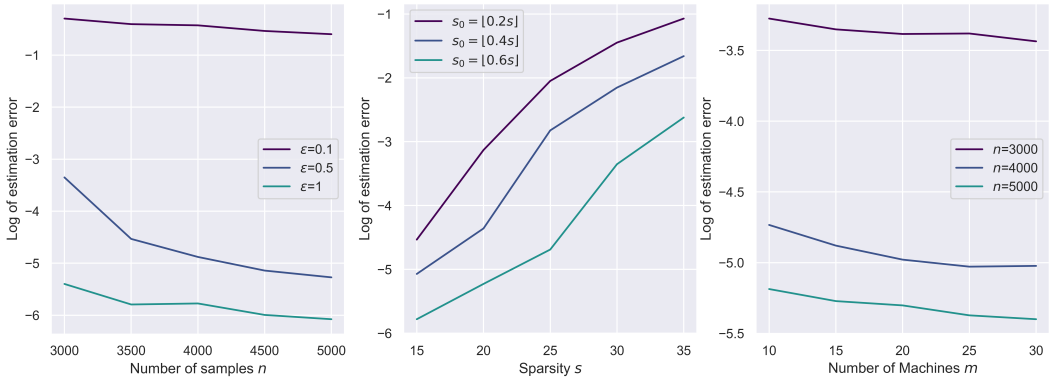

Fig 4: Plot for the estimation results. **Left:** Log estimation error with different number of samples n, **Middle:** Log estimation error with different sparsity s^*, **Right:** Log estimation error with different number of machines m.

From the left figure in Figure 6.1, we observe the decreasing error when we increase n. When the privacy parameter ϵ is large, we have better estimation error. From the middle figure, we observe that as the sparsity level s grows, the estimation error also increases. Also, when the sparsity for the shared signal s_0 becomes large, the estimation error also becomes large. In the right figure, we observe a consistent decrease of error when we increase the number of machines. All these figures support Theorem 7.

6.2 Simultaneous Inference

In this subsection, we investigate our proposed algorithms for simultaneous inference problems. We aim to build a simultaneous confidence interval when $\alpha = 0.05$ under three settings: $G = \{1, 2, \ldots, d\}$, $G = S$, and $G = S^c$. For each setting, we repeat 50 simulations and report the coverage and length of the confidence intervals. The results are shown in Table 5.

From simulation results, we can observe that our proposed simultaneous confidence interval mostly exhibit over-coverage for $G = \mathcal{S}^c$, and under-coverage for $G = \mathcal{S}$. This pattern has also been observed in previous works addressing simultaneous inference [38, 42]. Therefore, this could be attributed to the inherent nature of simultaneous inference rather than to algorithmic reasons.

7 Discussions and Future Work

In this paper, we study the high-dimensional estimation and inference problems within the context of federated learning. In scenarios involving an untrusted central server, our findings reveal that accurate estimation is infeasible, as the rate of convergence is adversely proportional to the dimension d. Conversely, in the trusted central server setting, we developed algorithms that achieve an optimal rate of convergence. We also explored

Simulation Results for Simultaneous Inference						
$(n, m, d, s^*, s_0, \epsilon)$	cov	$\text{cov}_{\mathcal{S}}$	$\text{cov}_{\mathcal{S}^c}$	len(cov)	$\text{len}(\text{cov}_{\mathcal{S}})$	$\text{len}(\text{cov}_{\mathcal{S}^c})$
(3000,15,800,15,8,0.8)	0.981	0.883	0.983	0.091	0.066	0.091
(4000,15,800,15,8,0.8)	0.985	0.910	0.987	0.079	0.057	0.079
(5000,15,800,15,8,0.8)	0.987	0.875	0.990	0.071	0.051	0.071
(4000,10,800,15,8,0.8)	0.989	0.894	0.991	0.079	0.057	0.079
(4000,20,800,15,8,0.8)	0.983	0.898	0.986	0.079	0.057	0.079
(4000,15,600,15,8,0.8)	0.993	0.878	0.995	0.077	0.057	0.077
(4000,15,1000,15,8,0.8)	0.994	0.878	0.997	0.080	0.057	0.080
(4000,15,800,15,4,0.8)	0.983	0.772	0.993	0.079	0.057	0.079
(4000,15,800,15,12,0.8)	0.975	0.957	0.974	0.079	0.058	0.079
(4000,15,800,10,8,0.8)	0.986	0.976	0.985	0.079	0.055	0.079
(4000,15,800,20,8,0.8)	0.974	0.850	0.982	0.078	0.059	0.078
(4000,15,800,15,8,0.5)	0.940	0.882	0.940	0.103	0.083	0.102
(4000,15,800,15,8,0.3)	0.953	0.789	0.975	0.127	0.097	0.126

Table 5: Simulation results of the private simultaneous inference in different settings.

inference challenges, detailing methodologies for both point-wise confidence intervals and simultaneous inference.

There are several extensions for further research. Currently, our models presume that each machine operates under a linear regression framework. We can possibly expand our algorithm to accomodate more complex models, such as generalized linear models, classification models, or broader machine learning models. Moreover, an interesting extension would be to refine our understanding of model similarity across machines. Although Section 5 currently bases model similarity on L_0 norms, reflecting non-sparse patterns, future studies could explore L_p norm-based similarities, particularly focusing on L_1 and L_2 norms, to enhance our approach to heterogeneous federated learning settings.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Jayadev Acharya, Clément L Canonne, and Himanshu Tyagi. General lower bounds for interactive high-dimensional estimation under information constraints. *arXiv preprint arXiv:2010.06562*, 2020.

[3] Naman Agarwal, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and H Brendan Mcmahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *arXiv preprint arXiv:1805.10559*, 2018.

[4] Leighton Pate Barnes, Wei-Ning Chen, and Ayfer Özgür. Fisher information under local differential privacy. *IEEE Journal on Selected Areas in Information Theory*, 1(3):645–659, 2020.

[5] Leighton Pate Barnes, Yanjun Han, and Ayfer Ozgur. Lower bounds for learning distributions under communication constraints via fisher information. *Journal of Machine Learning Research*, 21(236):1–30, 2020.

[6] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, pages 441–459, 2017.

[7] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1011–1020, 2016.

[8] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

[9] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy in generalized linear models: Algorithms and minimax lower bounds. *arXiv preprint arXiv:2011.03900*, 2020.

[10] Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. *The Annals of Statistics*, 41(6):2786–2819, 2013.

[11] John Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory*, pages 1161–1191. PMLR, 2019.

[12] Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *Conference On Learning Theory*, pages 1693–1702. PMLR, 2018.

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

[14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[15] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

[16] Cynthia Dwork, Weijie J Su, and Li Zhang. Differentially private false discovery rate control. *arXiv preprint arXiv:1807.04209*, 2018.

[17] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

[18] Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of distributed statistical estimation and dimensionality. *Advances in Neural Information Processing Systems*, 27:2726–2734, 2014.

[19] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[20] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2021.

[21] Yanjun Han, Ayfer Özgür, and Tsachy Weissman. Geometric lower bounds for distributed parameter estimation under communication constraints. In *Conference On Learning Theory*, pages 3163–3188. PMLR, 2018.

[22] Torsten Hothorn, Frank Bretz, and Peter Westfall. Simultaneous inference in general parametric models. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 50(3):346–363, 2008.

[23] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7(10):9530–9539, 2020.

[24] Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

[25] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[26] Jakub Konecy, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[27] Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1):115–144, 2017.

[28] Mengchu Li, Ye Tian, Yang Feng, and Yi Yu. Federated transfer learning with differential privacy. *arXiv preprint arXiv:2403.11343*, 2024.

[29] Sai Li, T Tony Cai, and Hongzhe Li. Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *arXiv preprint arXiv:2006.10593*, 2020.

[30] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[31] Andrew Lowy and Meisam Razaviyayn. Private federated learning without a trusted server: Optimal algorithms for convex losses. *arXiv preprint arXiv:2106.09779*, 2021.

[32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[33] Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees. *Google AI Blog*, 2022.

[34] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[35] Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 552–563. IEEE, 2017.

[36] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly-optimal private lasso. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3025–3033, 2015.

[37] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

[38] Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. 2014.

[39] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[40] Manuel Wiesenfarth, Tatyana Krivobokova, Stephan Klasen, and Stefan Sperlich. Direct simultaneous inference in additive models and its application to model undernutrition. *Journal of the American Statistical Association*, 107(500):1286–1296, 2012.

[41] Yang Yu, Shih-Kang Chao, and Guang Cheng. Distributed bootstrap for simultaneous inference under high dimensionality. *Journal of Machine Learning Research*, 23(195):1–77, 2022.

[42] Xianyang Zhang and Guang Cheng. Simultaneous inference for high-dimensional linear models. *Journal of the American Statistical Association*, 112(518):757–768, 2017.

[43] Yuchen Zhang, John C Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *NIPS*, pages 2328–2336. Citeseer, 2013.

[44] Yuchen Zhang, Martin J Wainwright, and John C Duchi. Communication-efficient algorithms for statistical optimization. *Advances in neural information processing systems*, 25, 2012.

[45] Zhe Zhang. Differential privacy in statistical learning. *ProQuest Dissertations and Theses*, page 156, 2023.

[46] Zhe Zhang and Linjun Zhang. High-dimensional differentially-private em algorithm: Methods and near-optimal statistical guarantees. *arXiv preprint arXiv:2104.00245*, 2021.

A Proof of main results

A.1 Proof of Theorem 1

We show the proof of the lower bound of the estimation. The main idea of the proof is as follows, we will first assume that in the general case where each data point on each machine follows a general distribution p_θ, then we will further assume some conditions of this distribution, and prove that the lower bound of the mean estimation could be attained under these conditions. Finally, we will show that under the assumptions that the data points follow the normal distribution, the specific conditions hold, thus we could finish the proof.

To start this proof, we first introduce the perturbation space $\mathcal{A} = \{-1, 1\}^k$, where k is a pre-chosen constant and associate each parameter θ with $a \in \mathcal{A}$ and refer the distribution p_θ as p_a. We characterize the distance between two parameters θ and θ' by the hamming distance of z and z', such approach will be compatible with the Assouad's method, as will be shown later in the proof. We note that when the hamming distance of a and a' get smaller, it indicts that the distance between θ and θ' becomes closer. Also, for each $a \in \mathcal{A}$, we further denote $a^{\oplus i} \in \mathcal{A}$ as the vector which flips the sign of the i-th coordinate of a. Then, we state below conditions:

Condition 1 *For every $a \in \mathcal{A}$ and $i \in [k]$, it holds that $\boldsymbol{p}_{a^{\oplus i}} \ll \boldsymbol{p}_a$. Further, there exist $q_{a,i}$ and measurable functions $\phi_{a,i} \colon \mathcal{X} \to \mathbb{R}$ such that $|q_{a,i}| \leq \alpha$, which q is a constant and:*

$$\frac{d\boldsymbol{p}_{a^{\oplus i}}}{d\boldsymbol{p}_a} = 1 + q_{a,i}\phi_{a,i}.$$

Condition 2 *For all $a \in \mathcal{A}$ and $i, j \in [k]$, $\mathbb{E}_{\boldsymbol{p}_a}[\phi_{a,i}\phi_{a,j}] = \mathbf{1}_{i=j}$.*

Condition 3 *There exists some $\sigma \geq 0$ such that, for all $a \in \mathcal{A}$, the random vector $\phi_a(X) = (\phi_{a,i}(X))_{i\in[k]} \in \mathbb{R}^k$ is σ^2-sub-Gaussian for $X \sim \boldsymbol{p}_a$ with independent coordinates.*

The above conditions characterize the distribution $\boldsymbol{p}_a$, we will later verify that the Gaussian distribution could satisfy the above conditions in the later proof. Then, we state our first claim.

Corollary 1 *For each coordinate of the A, for any $i = 1, 2, \ldots, k$, fix $\tau = \mathbb{P}(A_i = 1) \in (0, 1/2]$. Let $X_1, \ldots, X_m$ be the inputs on the local servers, i.i.d. with common distribution $\boldsymbol{p}_A^{\otimes n}$. Let Z^m be the information sent from all the local servers to the central machine generated through the channel $\mathcal{W}$. Then, if the condition 1 satisfies, there exists a constant c, we have:*

$$\left(\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})\right)^2 \leq \frac{c}{k}q^2mn^2 \max_{a\in\mathcal{A}} \sum_{i=1}^{k} \int_{\mathcal{Y}} \frac{\mathbb{E}_{\boldsymbol{p}_a^{\otimes n}}[\phi_{a,i}(X)\mathcal{W}(z|X)]^2}{\mathbb{E}_{\boldsymbol{p}_a^{\otimes n}}[\mathcal{W}(z|X)]} d\mu,$$

where $\boldsymbol{p}_{+i}^{Z^m} = \mathbb{E}[\boldsymbol{p}_A^{Z^m}|A_i = +1]$, $\boldsymbol{p}_{-i}^{Z^m} = \mathbb{E}[\boldsymbol{p}_A^{Z^m}|A_i = -1]$.

The proof of the above corollary is in appendix B.1. The above corollary characterizes the difference between the distribution of $\boldsymbol{p}_{+i}^{Z^m}$ and $\boldsymbol{p}_{-i}^{Z^m}$, which is the difference between the distribution of the information about the each coordinate of A, which could be seen as the information between Y and A, namely, the information between the information and the parameters.

In the precious corollary, we just assumed a general channel $\mathcal{W}$, in the following corollary, we could further specifies the above corollary when the channel $\mathcal{W}$, be a ϵ-differentially private constraint channel $\mathcal{W}^{priv}$ and we could further simplify the upper bound in Corollary 1.

Corollary 2 *If W^{priv} be a privacy constraint channel and for any family of distributions $\{\boldsymbol{p}_a, a \in \{-1,1\}^k\}$ satisfying condition 1 and condition 2. With the same notations as Corollary 1 we have:*

$$\left(\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})\right)^2 \leq \frac{7}{k} mn^2q^2(e^{n\epsilon^2}-1)$$

The proof of the above corollary could be found in appendix B.2. The above corollary focus on the upper bound of $\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})$, in the next corollary, we will focus on the lower bound, which is an Assouad-type bound. We first introduce another condition:

Condition 4 *Fix $p \in [1,\infty)$. Let ρ be the ℓ_p loss between the true parameter and the estimation. Then, for every $a, a' \in \mathcal{A} \in \{-1,+1\}^k$, the below inequalities hold:*

$$l_p(\theta_a, \theta_{a'}) \geq 4\rho\left(\frac{d_{Ham}(a,a')}{\tau k}\right)^{1/p},$$

where $d_{Ham}(a,a')$ denotes the Hamming distance with definition $d_{Ham}(a,a') = \sum_{i=1}^{k} \mathbf{1}(a_i \neq a_i')$, and $\tau = \mathbf{P}(a_i = 1) \in (0, 1/2]$ for each coordinate a_i.

The above condition characterizes the connection between θ with the perturbation space. With the above assumption, we could further obtain the lower bound of $\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})$:

Corollary 3 *Let $p \geq 1$ and assume that $\{\boldsymbol{p}_a, a \in \mathcal{A}\}$, $\tau \in [0, 1/2]$ satisfy Condition 4. Let A be a random variable on $\{-1,1\}^k$ with distribution $Rad(\tau)^{\otimes k}$. Suppose that $\hat{\theta}$ constitutes an (n,ρ)-estimator of the true parameter θ^* under l_p loss and $\mathbb{P}[\boldsymbol{p}_A \in \mathcal{P}_\Theta] \geq 1 - \tau/4$. Then the below inequality holds:*

$$\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m}) \geq \frac{n}{4},$$

where $\boldsymbol{p}_{+i}^{Z^m} = \mathbb{E}[\boldsymbol{p}_A^{Z^m} | A_i = +1]$, $\boldsymbol{p}_{-i}^{Z^m} = \mathbb{E}[\boldsymbol{p}_A^{Z^m} | A_i = -1]$.

The proof of the above corollary could be found in appendix B.3. In the following proof, we are going to verify that the Gaussian distribution satisfies all the above conditions, thus the result in Corollary 2 and Corollary 3 holds. Then, according to these two corollaries, we will present the lower bound for the mean estimation in the high-dimensional federated learning setting.

For the parameters, we could fix $p = 2$, $k = d$, $\mathcal{A} = \{-1, +1\}^d$. For the probability where $\tau = \mathbb{P}(a_i) = 1$, we fix $\tau = \frac{s}{2d}$. Let φ denote the probability density function of the standard Gaussian distribution $N(\mathbf{0}, \boldsymbol{I})$. We first suppose that, for some $\rho \in (0, 1/8]$, there exists an (n, ρ)-estimator for the true parameter μ under ℓ_p loss. Then, if we have $\rho^2 \geq s/n$, then we could finish the proof. Otherwise, we fix a parameter $\gamma = \frac{4\rho}{\sqrt{s/2}} \in (0, 1/2]$, this is possible with a choice of s, the sparsity level. We could design the parameter, the mean of the Gaussian distribution μ and A by the formula: $\mu_a = \gamma(a + \mathbf{1}_d)$, where $a \in \mathcal{A}$. Then, we could verify that $\mathbb{P}[\|\mu_a\|_0 \leq 2\tau d] \geq 1 - \tau/4$, where $\|\mu_a\|_0 = \sum_{i=1}^d \mathbf{1}_{a_i=1} = \|a\|_+$. From the definition of Gaussian density, for $a \in \mathcal{A}$, we have:

$$\boldsymbol{p}_a(x) = e^{-\gamma^2\|\mu_a\|_2^2/2} \cdot e^{\gamma\langle x, a+\mathbf{1}_d\rangle} \cdot \varphi(x).$$

Therefore, for $a \in \mathcal{A}$ and $i \in [d]$, we have

$$\boldsymbol{p}_{a^{\oplus i}}(x) = e^{-2\gamma x_i a_i} e^{2\gamma^2 a_i} \cdot \boldsymbol{p}_a(x) = (1 + q \cdot \phi_{a,i}(x)) \cdot \boldsymbol{p}_a(x),$$

where $q = \sqrt{e^{4\gamma^2} - 1}$ and $\phi_{a,i}(x) = \frac{1 - e^{-2\gamma x_i a_i} e^{2\gamma^2 a_i}}{\sqrt{e^{4\gamma^2}-1}}$. By using the Gaussian moment-generating function, we could verify that, for $i \neq j$,

$$\mathbb{E}_{\boldsymbol{p}_a}[\phi_{a,i}(X)] = 0, \quad \mathbb{E}_{\boldsymbol{p}_a}[\phi_{a,i}(X)^2] = 1, \text{ and } \mathbb{E}_{\boldsymbol{p}_a}[\phi_{a,i}(X)\phi_{a,j}(X)] = 0,$$

so that the condition 1 and condition 2 are satisfied. Here, notice that in the proof of Corollary 1, we require that $|q \cdot \phi_{a,i}(x)| = C/n$ where C is a constant, we could verify that since $\rho \leq c \cdot \sqrt{s/n}$, then $\gamma \leq c\sqrt{n}$ from the definition of γ, Then we have $|q \cdot \phi_{z,i}(x)| \leq c_0|\gamma^2 - \gamma x_i| \leq c_0/n$, which could verify the condition for corollary 1. Also, by the choice of γ and ρ, it is easy to verify that condition 4 also holds with:

$$\ell_2(\mu(\boldsymbol{p}_a), \mu(\boldsymbol{p}_{a'})) = 4\rho \cdot \sqrt{\frac{d_{ham}(a, a')}{\tau d}}.$$

Thus, all the conditions mentioned above have been verified. Then, we could finish the proof of our lower bound. Combining the result of corollary 2 and corollary 3, we get the result below:

$$n^2 d \leq cmn^2 q^2 (e^{n\epsilon^2} - 1),$$

where c is a constant. Also, notice that $q^2 = e^{4\gamma^2} - 1 \leq 8\gamma^2$ holds since $\gamma \leq 1/2$, we could find a constant c_0, it follows that

$$\rho^2 \geq c_0 \cdot \frac{s \cdot d}{mn\epsilon^2},$$

From the choice of ρ, we could claim that $\rho \geq \Omega(\sqrt{\frac{sd}{mn\epsilon^2}} \wedge 1)$, then we could obtain our lower bound, which finished the proof. □

A.2 Proof of Theorem 2

In the proof of Theorem 2, we will design a mechanism to get an estimation of the parameter and then we obtain the upper bound of $\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2$. The overall mechanism is designed as follows: we first calculate the mean for n data points on each machine. Then, we transform the Gaussian mean to Bernoulli mean according to the sign of the Gaussian mean motivated by the Algorithm 2 discussed in [2], l-bit protocol for estimating product of Bernoulli family.

Then, we could use the ϵ-local differentially private mechanism to achieve mean estimation for the product of Bernoulli family in the federated learning setting. After obtaining the estimation, we could convert the estimated Bernoulli mean back to Gaussian mean estimation.

First, for each data point on the machine, it follows the distribution of $N(\boldsymbol{\mu}, \boldsymbol{I}_d)$. Then for the mean on i-th machine, the mean $\bar{\boldsymbol{X}}_i$ follows a distribution of $N(\boldsymbol{\mu}, 1/n\boldsymbol{I})$. Then, we could convert it to a Bernoulli variable $\mathbf{Z}$, where $Z_i = 1$ when $\bar{X_{ij}} > 0$ and $Z_i = -1$ when $\bar{X_{ij}} \leq 0$. Then the mean of $\boldsymbol{Z}$, which denote as $\boldsymbol{v}$ is:

$$v_i = 2\mathbb{P}(X_i > 0) - 1 = \text{Erf}\left(\frac{\sqrt{n}\mu_i}{\sqrt{2}}\right),$$

for each coordinate of $\boldsymbol{v}$. Suppose the estimation of $\boldsymbol{v}$ is denoted by $\hat{\boldsymbol{v}}$, then suppose the estimation $\hat{\boldsymbol{\mu}}$ is given by $\hat{\mu_i} = \frac{\sqrt{2}}{\sqrt{n}}\text{erf}^{-1}(\hat{v_i})$, we could find such relationship:

$$\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2 = \sum_{i=1}^{d} |\mu_i - \hat{\mu}_i|^2 = \frac{2}{n} \cdot \sum_{i=1}^{d} |\text{Erf}^{-1}(v_i) - \text{Erf}^{-1}(\hat{v}_i)|^2 \leq c \cdot \frac{1}{n} \cdot \sum_{i=1}^{d} |v_i - \hat{v}_i|^2, \tag{A.1}$$

where c is a constant. The last inequality comes from the Lipschitz condition of a Erf function. Then, we could get the upper bound for the Bonoulli mean estimation directly from Theorem 3 in [2], where

$$\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|_2^2 \leq c \cdot \frac{d \cdot s}{m\epsilon^2},$$

where ϵ is the privacy parameter, m is the number of machines. Combining the last two inequalities (A.1) and (A.2), we could get the upper bound for the mean Gaussian estimation:

$$\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2 \leq c \cdot \frac{s \cdot d}{mn\epsilon^2},$$

which finished the proof. □

A.3 Proof of Theorem 3

It is not difficult to observe that the convergence rate would be the same as in the non-federated learning setting. We denote L_n as the sample loss function and L be the population level. In the estimation of $\boldsymbol{\beta}$, $L(\boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ and L_n is the sample version. In the estimation of $\boldsymbol{\Theta}_k$, $L(\boldsymbol{\Theta}_k) = \frac{1}{2}\boldsymbol{\Theta}_k^\top \Sigma \boldsymbol{\Theta}_k - \langle \boldsymbol{e}_j, \boldsymbol{\Theta}_k \rangle$. We start from the estimation of $\boldsymbol{\beta}$ and the estimation of Θ is the same. In this proof, we use n_0 to refer the total number of samples $n_0 = m \cdot n$. Then, it holds that:

Lemma A.1 *Under assumptions of Theorem 5, it holds that:*

$$8\nu_s \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}\|_2^2 \leq \langle \nabla \mathcal{L}_n(\boldsymbol{\beta}^t) - \nabla \mathcal{L}_n(\hat{\boldsymbol{\beta}}), \boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}} \rangle \leq 8\mu_s \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}\|_2^2. \tag{A.2}$$

Proof: From direct calculation, we could obtain that:

$$\langle \nabla \mathcal{L}_n(\boldsymbol{\beta}^t) - \nabla \mathcal{L}_n(\hat{\boldsymbol{\beta}}), \boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}} \rangle = 2(\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}})^T \hat{\boldsymbol{\Sigma}} (\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}) \leq 2\mu_{s+s^*} \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}\|_2^2 \leq 2\mu_{2s} \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}\|_2^2$$

The last inequality is according to the choice of s such that $s^* \leq s$. Then, we also have $\mu_{2s} \leq 4\mu_s$. Thus we have obtained the right hand side of the inequality. By a similar approach, we could also obtain the left hand side.

Lemma A.2 *Under assumptions of Theorem 5, it holds that there exists an absolute constant ρ such that*

$$\mathcal{L}_n(\boldsymbol{\beta}^{t+1}) - \mathcal{L}_n(\hat{\boldsymbol{\beta}}) \leq \left(1 - \frac{\nu_s}{24\mu_s}\right) \left(\mathcal{L}_n(\boldsymbol{\beta}^t) - \mathcal{L}_n(\hat{\boldsymbol{\beta}})\right) + c_3 \left(\sum_{i \in [s]} \|\boldsymbol{w}_i^t\|_\infty^2 + \|\tilde{\boldsymbol{w}}_{S^{t+1}}^t\|_2^2\right), \tag{A.3}$$

where c_3 is a constant number such that $c_3 = \max(\mu_s(72 \cdot 8\mu_s + 13), 68\mu_s + 2/3)$

Notice that w_i, w are injected from the NoisyHT algorithm. The proof of the above lemma follows from the result in Lemma 8.3 from [8]. Then, we could start the proof by iterating (A.3) over t. Denote $\boldsymbol{W}_t = c_3 \left(\sum_{i \in [s]} \|\boldsymbol{w}_i^t\|_\infty^2 + \|\tilde{\boldsymbol{w}}_{S^{t+1}}^t\|_2^2\right)$ to obtain

$$\begin{aligned}
\mathcal{L}_{n_0}(\boldsymbol{\beta}^T) - \mathcal{L}_{n_0}(\hat{\boldsymbol{\beta}}) &\leq \left(1 - \frac{\nu_s}{24\mu_s}\right)^T \left(\mathcal{L}_{n_0}(\boldsymbol{\beta}^0) - \mathcal{L}_{n_0}(\hat{\boldsymbol{\beta}})\right) + \sum_{k=0}^{T-1} \left(1 - \frac{1}{\rho L^2}\right)^{T-k-1} \boldsymbol{W}_k \\
&\leq \left(1 - \frac{\nu_s}{24\mu_s}\right)^T 4\mu c_0^2 + \sum_{k=0}^{T-1} \left(1 - \frac{\nu_s}{24\mu_s}\right)^{T-k-1} \boldsymbol{W}_k.
\end{aligned} \tag{A.4}$$

The second inequality is a consequence of the upper inequality in (A.2) and the ℓ_2 bounds of $\boldsymbol{\beta}^0$ and $\hat{\boldsymbol{\beta}}$. We can also bound $\mathcal{L}_{n_0}(\boldsymbol{\beta}^T) - \mathcal{L}_{n_0}(\hat{\boldsymbol{\beta}})$ from below by the lower inequality in (A.2):

$$\mathcal{L}_{n_0}(\boldsymbol{\beta}^T) - \mathcal{L}_{n_0}(\hat{\boldsymbol{\beta}}) \geq \mathcal{L}_{n_0}(\boldsymbol{\beta}^T) - \mathcal{L}_{n_0}(\boldsymbol{\beta}^*) \geq 4\nu_s \|\boldsymbol{\beta}^T - \boldsymbol{\beta}^*\|_2^2 - \langle \nabla \mathcal{L}_{n_0}(\boldsymbol{\beta}^*), \boldsymbol{\beta}^* - \boldsymbol{\beta}^T \rangle. \tag{A.5}$$

Now (A.4) and (A.5) imply that, with $T = (\rho L^2)\log\big(8c_0^2 L n_0\big)$,

$$4\nu_s\|\boldsymbol{\beta}^T - \boldsymbol{\beta}^*\|_2^2 \leq \|\nabla\mathcal{L}_{n_0}(\boldsymbol{\beta}^*)\|_\infty\sqrt{s+s^*}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^T\|_2 + \frac{1}{n_0} + \sum_{k=0}^{T-1}\left(1 - \frac{\nu_s}{24\mu_s}\right)^{T-k-1}\boldsymbol{W}_k. \tag{A.6}$$

$$\leq \|\nabla\mathcal{L}_{n_0}(\boldsymbol{\beta}^*)\|_\infty\sqrt{s+s^*}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^T\|_2 + \frac{1}{n_0} + \frac{24\mu_s}{\nu_s}\max_k \boldsymbol{W}_k. \tag{A.7}$$

Thus,

$$\|\boldsymbol{\beta}^T - \boldsymbol{\beta}^*\|_2^2 \leq k\cdot\frac{s^*\log d}{n_0} + \frac{\mu_s}{\nu_s^2}\max_k \boldsymbol{W}_k.$$

In the above inequality, k is a constant. Then, we could calculate the upper bound of $\mathbf{W}_k$. From the result of tail bound of Laplace random variables, we could find that with high probability that $\mathbf{W}_k \leq c_4 s^2\log^2 d\log(1/\delta)\log n^3/n^2\epsilon^2)$, where $c_4 = \max(\mu_s(9\mu_s + 1/4), 17/16\mu_s + 1/96)$. Then, we have with high probability:

$$\|\boldsymbol{\beta}^T - \boldsymbol{\beta}^*\|_2^2 \leq k\cdot\frac{s\log d}{n_0} + \frac{6c_4\mu_s}{\nu_s^2}s^2\log^2 d\log(1/\delta)\log n_0^3/n_0^2\epsilon^2.$$

Similarly, we could obtain the same result for the estimation of $\hat{\boldsymbol{\Theta}}_k$, which finishes the proof.

A.4 Proof of Theorem 4

The structure of the proof consist of three part, the first part is to show that our algorithm provides an (ϵ,δ)-differentially private confidence interval. In the second part, we will show that $\hat{\beta}_k$ is a consistent estimator of true β_k, which is unbiased. In the last part, we will show that the $(1-\alpha)$ confidence interval is asymptotically valid. Before we start the first part, let us first analyze c_x:
According to the assumptions of the theorem, we have learnt that for each row of $\boldsymbol{X}$, $\boldsymbol{x}\boldsymbol{\Sigma}^{-1/2}$ is sub-Gaussian with $\kappa = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_{\psi_2}$. Then according to the properties of sub-Gaussian random variables, we have: $\|\boldsymbol{x}\boldsymbol{\Sigma}^{-1/2}\|_\infty \leq 3\sqrt{2\kappa^2\log d}$ with probability $1-d^{-2}$. Then for each element of x_i, $i = 1, 2, \ldots, d$, we have:

$$x_i = \boldsymbol{e}_j^\top\boldsymbol{x} = \boldsymbol{e}_j^\top\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$$

Thus,

$$x_i \leq \|\boldsymbol{e}_j^\top\boldsymbol{\Sigma}^{1/2}\|_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_\infty \leq \|\boldsymbol{\Sigma}^{1/2}\|_2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_\infty$$

Then, with probability $1-d^{-2}$, we have $x_i \leq 3\sqrt{2L\kappa^2\log d}$. By a union bound, we could have with probability $1-d^{-1}$, $\|\boldsymbol{x}\|_\infty \leq 3\sqrt{2L\kappa^2\log d}$. By the choice of c_x in the theorem, we have $\|\boldsymbol{x}\|_\infty \leq c_x$ with a high probability.
Then, we could verify that the confidence interval is (ϵ,δ)-differentially private. From [8], we could obtain that the output $\hat{\boldsymbol{\beta}}^{\mathrm{u}}$ is (ϵ,δ)-DP. In a similar manner, we could also

verify that the output $\hat{\boldsymbol{\Theta}}_k$ is also (ϵ, δ)-DP. Thus, for two adjacent data sets $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{X}', \boldsymbol{Y}')$ which differ by one data $(\boldsymbol{x}_{ij}, y_{ij})$ and $(\boldsymbol{x}'_{ij}, y'_{ij})$, we have:

$$\begin{aligned}
|\frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\Pi_R(y_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\beta}}^{\mathrm{u}})| &\leq \frac{1}{n_0}|\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\Pi_R(y_{ij})| + \frac{1}{n_0}|\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\beta}}^{\mathrm{u}}| \\
&\leq \frac{1}{n_0}|\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}||\Pi_R(y_{ij})| + \frac{1}{n_0}|\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}||\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\beta}}^{\mathrm{u}}| \\
&\leq \frac{1}{n_0}\sqrt{s}c_1 c_x R + \frac{1}{n_0} s c_0 c_1 c_x^2
\end{aligned}$$

Thus,

$$\begin{aligned}
&|\frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\Pi_R(y_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\beta}}^{\mathrm{u}}) - \frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}'_{ij}\Pi_R(y'_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}'_{ij}\boldsymbol{x}'^\top_{ij} \hat{\boldsymbol{\beta}}^{\mathrm{u}})| \\
&= |\frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\Pi_R(y_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\beta}}^{\mathrm{u}})| + |\frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}'_{ij}\Pi_R(y'_{ij}) - \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}'_{ij}\boldsymbol{x}'^\top_{ij} \hat{\boldsymbol{\beta}}^{\mathrm{u}})| \\
&\leq \frac{2}{n_0}\sqrt{s}c_1 c_x R + \frac{2}{n} s c_0 c_1 c_x^2
\end{aligned}$$

Denote $\Delta_1 = \sqrt{s}c_1 c_x R + s c_0 c_1 c_x^2$. Thus, if E_k follows $N(0, 8\Delta_1^2/n_0^2\epsilon^2 \log(1.25/\delta))$, $\hat{\beta}_j$ is (ϵ, δ)-DP. For the term $\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k$, we could obtain that:

$$\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k = \frac{1}{n_0}\sum_{i=1}^{n} \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top \hat{\boldsymbol{\Theta}}_k = \frac{1}{n_0}\sum_{i=1}^{n} (\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij})^2$$

Thus, for two adjacent data sets $\boldsymbol{X}$ and $\boldsymbol{X}'$ differ by one data $\boldsymbol{x}_{ij}$ and $\boldsymbol{x}'_{ij}$, we have:

$$|\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k - \hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}}' \hat{\boldsymbol{\Theta}}_k| \leq \frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij})^2 + \frac{1}{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}'_{ij})^2$$

By Holder inequality and Cauchy inequality, we have $|\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{x}_{ij}| \leq \sqrt{s}c_1 c_x$, thus we have:

$$|\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k - \hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}}' \hat{\boldsymbol{\Theta}}_k| \leq \frac{2}{n_0}(\sqrt{s}c_1 c_x)^2 = \frac{2}{n_0} s c_1^2 c_x^2$$

Denote $\Delta_2 = s c_1^2 c_x^2$. Then, let E' follows a Gaussian distribution of $N(0, 8\Delta_2^2/n^2 m^2 \epsilon^2 \log(1.25/\delta))$. We could claim that $\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + E'$ is (ϵ, δ)-differentially private.
We start the second part of the proof. First, with probability $1 - k_0 \exp(-k_1 n_0)$, we have $\Pi_R(y_i) = y_i$ for each $i = 1, 2, \ldots, d$, so we could decompose $\hat{\beta}_k$ by the following approach:

$$\begin{aligned}
\hat{\beta}_k &= \hat{\beta}_k^{\mathrm{u}} + \frac{1}{n_0}\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{W} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{\mathrm{u}}) + E_k \\
&= \hat{\beta}_k^{\mathrm{u}} + \frac{1}{n_0}\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) + \frac{1}{n_0}\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top \boldsymbol{W} + E_k \\
&= \beta_k + (\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) + \frac{1}{n_0}\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top \boldsymbol{W} + E_k
\end{aligned}$$

Thus, we have:

$$\sqrt{n_0}(\hat{\beta}_j - \beta_j) = \underbrace{\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}})}_{A.8.1} + \underbrace{\frac{1}{\sqrt{n_0}} \hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top \boldsymbol{W}}_{A.8.2} + \underbrace{\sqrt{n_0} E_k}_{A.8.3} \tag{A.8}$$

We will analyze the three terms in (A.8) one by one. For the first term, we could further decompose this term as:

$$\begin{aligned}\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) &= \sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} + \boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \\ &= \sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) + \sqrt{n_0}(\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}})\end{aligned} \tag{A.9}$$

For the first term in (A.9), we could further decompose this term from $\hat{\boldsymbol{\Sigma}} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^\top$:

$$\begin{aligned}\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) &= \sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top - \boldsymbol{\Theta}_k^\top)\hat{\Sigma}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \\ &\leq \sqrt{n_0}\lambda_s(\hat{\Sigma})\|\hat{\boldsymbol{\Theta}}_k - \boldsymbol{\Theta}_k\|_2 |\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2\end{aligned} \tag{A.10}$$

In the last inequality, we use λ_s to denote the largest s-restricted eigenvalue of the covariance matrix $\hat{\Sigma}$. From Theorem 3, we could obtain that there exists a constant c such that:

$$\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2^2 \leq c \cdot \sigma^2 \left(\frac{s^* \log d}{n_0} + \frac{(s^* \log d)^2 \log(1/\delta) \log^3 n_0}{n_0^2 \varepsilon^2} \right)$$

Also, for the output $\hat{\boldsymbol{\Theta}}_k$, we could have the similar result:

$$\|\hat{\boldsymbol{\Theta}}_k - \boldsymbol{\Theta}_k\|_2^2 \leq c \cdot \sigma^2 \left(\frac{s^* \log d}{n_0} + \frac{(s^* \log d)^2 \log(1/\delta) \log^3 n_0}{n_0^2 \varepsilon^2} \right)$$

Combining (A.4) and (A.4), we could obtain that

$$\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) = o\left(\frac{s^* \log d}{\sqrt{n_0}} \right) = o(1)$$

Then, we could focus on the second term of (A.9). We first introduce the following lemma:

Lemma A.3 *(Lemma 6.2 in [24]) For the vector $\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k$. Denote $\kappa = \|\Sigma^{-1/2}\boldsymbol{X}_1\|_{\phi_2}$, then with probability $1 - 2d^{1-a^2/24e^2\kappa^4 L^2}$, we have:*

$$\|\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k\|_\infty \leq a\sqrt{\frac{\log d}{n_0}}$$

Thus, for the second term of (A.9), we have:

$$\sqrt{n_0}(\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \leq \sqrt{n_0}\|\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top\|_\infty \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_1$$

$$\leq k\sqrt{n_0}\sqrt{\frac{\log d}{n_0}}\sqrt{s^*}\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2$$
$$\leq k\cdot\sqrt{s^*\log d}\cdot\sqrt{\frac{s^*\log d}{n_0}}=o(1) \tag{A.11}$$

Combine the result from (A.4), (A.11) to (A.9), we could obtain that the first term of (A.8) is $o(1)$. We could also analyze the third term of (A.8), $\sqrt{n_0}\boldsymbol{E}_k\sim N(0,8\Delta_1^2\log(1.25/\delta)/n_0\epsilon^2)$. Then, by the definition of Δ_1, we have $8\Delta_1^2\log(1.25/\delta)/n_0\epsilon^2\sim\frac{s^{*2}\log^2 d\log(1.25/\delta)}{n_0\epsilon^2}=o(1)$ from the assumption. Also, we notice that $\boldsymbol{E}'=N(0,c\cdot\frac{s^{*2}\log^2 d\log(1.25/\delta)}{n_0^2\epsilon^2})$. By the concentration of Gaussian distribution, we also have that $\boldsymbol{E}'=o(1)$.

Finally, we analyze the term $\frac{1}{\sqrt{n_0}}\hat{\boldsymbol{\Theta}}_j^\top\boldsymbol{X}^\top\boldsymbol{W}$. From our definition, W is sub-Gaussian random noise. Then, from the central limit theorem, we could conclude that:

$$\frac{1}{\sqrt{n_0}}\hat{\boldsymbol{\Theta}}_j^\top\boldsymbol{X}^\top\boldsymbol{W}\to N(0,\sigma^2\hat{\boldsymbol{\Theta}}_j^\top\hat{\Sigma}\hat{\boldsymbol{\Theta}}_j)$$

Thus, $\sqrt{n_0}(\hat{\beta}_j-\beta_j)=\frac{1}{\sqrt{n_0}}\hat{\boldsymbol{\Theta}}_j^\top\boldsymbol{X}^\top\boldsymbol{W}+\sqrt{n_0}\boldsymbol{E}_k\sim N(0,\sigma^2\hat{\boldsymbol{\Theta}}_j^\top\hat{\Sigma}\hat{\boldsymbol{\Theta}}_j+o(1))$. Also, from lemma 4.1, we could claim that under our assumptions, $\hat{\sigma}^2=\sigma^2+o(1)$. We could get the result where with high probability, $\frac{\sqrt{n_0}(\hat{\beta}_j-\beta_j)}{\hat{\sigma}\sqrt{\hat{\boldsymbol{w}}_j{}^\top\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{w}}_j}}\to N(0,1)$.

Therefore, we could claim that $[\hat{\beta}_j-\Phi^{-1}(1-\alpha/2)\frac{\hat{\sigma}}{\sqrt{n_0}}\sqrt{\hat{\boldsymbol{\Theta}}_j^\top\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_j},\hat{\beta}_j+\Phi^{-1}(1-\alpha/2)\frac{\hat{\sigma}}{\sqrt{n_0}}\sqrt{\hat{\boldsymbol{\Theta}}_j^\top\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}}_j}]$ is asymptotically $1-\alpha$ confidence interval for β_j. Therefore, we have finished the proof of theorem. □

A.5 Proof of Theorem 5

The proof is similar to the proof of Theorem 4, the difference is that we need to consider the case where the privacy cost is not dominated by the statistical error. Then, for the proof of Theorem 5, we follow the proof of Theorem 4 until (A.8). The analysis for the second term and the third term for (A.8) stays the same. On the other hand, for the first term of (A.8), we have: We will analyze the three terms in (A.8) one by one. For the first term, in the same manner, we could decompose this term as:

$$\begin{aligned}\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{e}_k^\top)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})&=\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Theta}_k^\top\hat{\boldsymbol{\Sigma}}+\boldsymbol{\Theta}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{e}_k^\top)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})\\&=\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Theta}_k^\top\hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})+\sqrt{n}(\boldsymbol{\Theta}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{e}_k^\top)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})\end{aligned} \tag{A.12}$$

For the first term in (A.12), we could further decompose this term from $\hat{\boldsymbol{\Sigma}}=\frac{1}{n}\sum_{i=1}^n\boldsymbol{x}_i\boldsymbol{x}_i^\top$:

$$\begin{aligned}\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Theta}_k^\top\hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})&=\sqrt{n_0}(\hat{\boldsymbol{\Theta}}_k^\top-\boldsymbol{\Theta}_k^\top)\hat{\Sigma}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})\\&\leq\sqrt{n_0}\lambda_s(\hat{\Sigma})\|\hat{\boldsymbol{\Theta}}_k-\boldsymbol{\Theta}_k\|_2|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2\\&\leq\frac{k\mu_s^2}{\nu_s^2}s^2\log^2 d\log(1/\delta)\log^3 n_0/n_0^{3/2}\epsilon^2\end{aligned} \tag{A.13}$$

Thus, for the second term of (A.12), by Lemma A.3, we have:

$$\begin{aligned}\sqrt{n_0}(\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) &\leq \sqrt{n_0}\|\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top\|_\infty \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_1 \\ &\leq k\sqrt{n_0}\sqrt{\frac{\log d}{n_0}}\sqrt{s^*}\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2 \\ &\leq k \cdot \sqrt{s^* \log d} \cdot \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2 \end{aligned} \tag{A.14}$$

When the privacy cost is not dominated by the statistical error and also $s^* \log d/\sqrt{n} = o(1)$, we can observe that the equation (A.14) has smaller convergence rate that (A.13). Then, combining (A.13) and (A.14), there exists a constant k_1, such that:

$$\sqrt{n}(\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \leq \frac{\gamma \mu_s^2}{\nu_s^2} \frac{s^2 \log^2 d \log(1/\delta) \log^3 n_0}{n_0^{3/2}\epsilon^2}$$

Then, insert the result into (A.8), we have:

$$\sqrt{n_0}(\hat{\beta}_j - \beta_j) = O\left(-\frac{\mu_s^2}{\nu_s^2}\frac{s^2 \log^2 d \log(1/\delta)\log^3 n_0}{n_0^{3/2}\epsilon^2}\right) + \frac{1}{\sqrt{n_0}}\hat{\boldsymbol{\Theta}}_k^\top \boldsymbol{X}^\top \boldsymbol{W} + \sqrt{n_0}E_k \tag{A.15}$$

Notice that for the first term on the right hand, the constant could be set to 1 because it comes from the tail bound of Laplace random variable. From the result in (A.15), we could also apply the central limit theorem to show that the second term is asymptotically Gaussian, notice that in the right hand side, the second term and the third term asymptotically follows a distribution of $N(0, \sigma^2 \hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n_0 \epsilon^2})$. Also, by the concentration of Gaussian distribution, we have with high probability, $E' \leq \frac{8\Delta_1^2 \log(1/\delta)}{n_0\epsilon^2}$.Thus, the privacy conditions are satisfied. Therefore, we have:

$$\sqrt{n_0}[\hat{\beta}_j - \beta_j - \sqrt{n_0}(\boldsymbol{\Theta}_k^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_k^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}})]/\sqrt{\sigma^2 \hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n_0\epsilon^2}} \sim N(0,1)$$

Then, also by our assumptions and the result in Lemma 4.1, we could claim that $\sigma^2 = \hat{sigma}^2 + o(1)$. Thus, finally, the confidence interval is given by:

$$\begin{aligned} J_j(\alpha) = \Bigg[&\hat{\beta}_j - \frac{\gamma \hat{\mu_s}^2}{\hat{\nu_s}^2}\frac{s^2 \log^2 d \log(1/\delta)\log^3 n_0}{n_0^2\epsilon^2}) - \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{n_0}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n_0\epsilon^2}}, \\ &\hat{\beta}_j + \frac{\gamma \hat{\mu_s}^2}{\hat{\nu_s}^2}\frac{s^2 \log^2 d \log(1/\delta)\log^3 n_0}{n_0^2\epsilon^2}) + \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{n_0}}\sqrt{\hat{\boldsymbol{\Theta}}_k^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_k + \frac{8\Delta_1^2 \log(1/\delta)}{n_0\epsilon^2}}\Bigg] \end{aligned}$$

which finishes our proof.

A.6 Proof of Theorem 6

Let us first show that our algorithm is ϵ, δ private. The major proof lies in the choice of noise level B_3. To decompose, we can find:

$$\hat{\boldsymbol{\Theta}}\frac{1}{\sqrt{m}}\sum_{i=1}^{m} e_i\sqrt{n}(\boldsymbol{g}_i - \bar{\boldsymbol{g}}) = \hat{\boldsymbol{\Theta}}\frac{1}{\sqrt{m}}\sum_{i=1}^{m} e_i\sqrt{n}\boldsymbol{g}_i - \hat{\boldsymbol{\Theta}}\frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)\bar{\boldsymbol{g}}$$

Then, suppose in an adjacent data set, the different data in denoted as $(\boldsymbol{x}_{ij}, y_{ij})$ and $(\boldsymbol{x}'_{ij}, y'_{ij})$. Then, we calculate:

$$\begin{aligned}
&\left\|(\hat{\boldsymbol{\Theta}}\frac{1}{\sqrt{m}}\sum_{i=1}^{m} e_i\sqrt{n}\boldsymbol{g}_i - \hat{\boldsymbol{\Theta}}\frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)\bar{\boldsymbol{g}}) - (\hat{\boldsymbol{\Theta}}\frac{1}{\sqrt{m}}\sum_{i=1}^{m} e_i\sqrt{n}\boldsymbol{g}'_i - \hat{\boldsymbol{\Theta}}\frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)\bar{\boldsymbol{g}'})\right\|_\infty \\
&\leq \left\|\hat{\boldsymbol{\Theta}}\frac{\sqrt{n}}{\sqrt{m}}e_i(\boldsymbol{g}_i - \boldsymbol{g}'_i) - \hat{\boldsymbol{\Theta}}\frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)(\bar{\boldsymbol{g}} - \bar{\boldsymbol{g}'})\right\|_\infty \\
&\leq \left\|\hat{\boldsymbol{\Theta}}\right\|_{\max}\left\|\frac{\sqrt{n}}{\sqrt{m}}e_i(\boldsymbol{g}_i - \boldsymbol{g}'_i) - \frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)(\bar{\boldsymbol{g}} - \bar{\boldsymbol{g}'})\right\|_\infty \\
&\leq (\left\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\right\|_{\max} + \|\boldsymbol{\Theta}\|_{\max})\left\|\frac{\sqrt{n}}{\sqrt{m}}e_i(\boldsymbol{g}_i - \boldsymbol{g}'_i) - \frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)(\bar{\boldsymbol{g}} - \bar{\boldsymbol{g}'})\right\|_\infty \\
&\leq (\left\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\right\|_{1} + \|\boldsymbol{\Theta}\|_{2})\left\|\frac{\sqrt{n}}{\sqrt{m}}e_i(\boldsymbol{g}_i - \boldsymbol{g}'_i)\right\|_\infty + \left\|\frac{\sqrt{n}}{\sqrt{m}}\left(\sum_{i=1}^{m} e_i\right)(\bar{\boldsymbol{g}} - \bar{\boldsymbol{g}'})\right\|_\infty) \\
&\leq (o(1) + L)\frac{\sqrt{n}}{\sqrt{m}}\sqrt{\log m}\|(\boldsymbol{g}_i - \boldsymbol{g}'_i)\|_\infty + \frac{\sqrt{n}}{\sqrt{m}}m\sqrt{\log m}\|\bar{\boldsymbol{g}} - \bar{\boldsymbol{g}'}\|_\infty) \\
&\leq L\frac{4\sqrt{\log m}}{\sqrt{mn}}\left\|\boldsymbol{x}_{ij}(\pi_R(y_{ij}) - \boldsymbol{x}_{ij}\hat{\boldsymbol{\beta}})\right\|_\infty \\
&\leq L\frac{4\sqrt{\log m}}{\sqrt{mn}}c_x(R + c_0c_x\sqrt{s^*})
\end{aligned}$$

Thus, the privacy could be guaranteed. Then, let us start the proof of consistency. Throughout the proof, we define $n_0 = m \cdot n$, $(\boldsymbol{X}, \boldsymbol{Y})$ be the whole data set where $\boldsymbol{X} \in \mathbb{R}^{n_0 * d}$ and $\boldsymbol{Y} \in \mathbb{R}^d$. $U' = \max_{k\in G}\hat{\boldsymbol{\Theta}}\frac{1}{\sqrt{m}}\sum_{i=1}^{m} e_i\sqrt{n}(\boldsymbol{g}_i - \bar{\boldsymbol{g}})$. Let us define another multiplier bootstrap statistic:

$$U^* = \max_{k\in G}\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\Sigma^{-1}\boldsymbol{x}_{ij}(y_{ij} - \boldsymbol{x}_{ij}\boldsymbol{\beta})e_{ij},$$

where e_{ij} are all standard Gaussian variables. At the same time, we also define:

$$M_0 = \max_{k\in G}\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\Sigma^{-1}\boldsymbol{x}_{ij}(y_{ij} - \boldsymbol{x}_{ij}\boldsymbol{\beta})$$

The proof consists three major steps, we start from the first step and measure $\sup_{\alpha\in(0,1)}|\mathbb{P}(M_0 \leq C_{U^*}(\alpha)) - \alpha|$. This measurement is quite straightforward, we could apply Theorem 3.1 from [10]. However, we need to verify Corollary 2.1 from [10]. Notice that for any k, $\mathbb{E}[\Theta_k^T \boldsymbol{x}_{ij}(y_{ij} - \boldsymbol{x}_{ij}\boldsymbol{\beta})]^2 = \sigma^2\Theta_k^T\Sigma\Theta_k \geq \sigma^2/L$. Also, it is not difficult to verify that $\Theta_k^T\boldsymbol{x}_{ij}(y_{ij} - \boldsymbol{x}_{ij}\boldsymbol{\beta})$ is sub-exponential. Since from assumption D1, we have $\boldsymbol{x}_{ij}$ is sub-Gaussian and from the linear model, we know that $(y_{ij} - \boldsymbol{x}_{ij}\boldsymbol{\beta})$ is also, sub-Gaussian. Then, the condition could be verified. Thus, by applying Theorem 3.1 and also under the condition where there exists a constant k, k_0, k_1 such that $\log^7(dmn)/mn \leq \frac{1}{(mn)^k}$we could have:

$$\begin{aligned}\sup_{\alpha\in(0,1)}|\mathbb{P}(T_0 \leq C_{U^*}(\alpha)) - \alpha| &\leq k_0 \cdot \frac{1}{(mn)^{k_1}} + k_2 v^{1/3}(\max(1, \log(d/v)))^{2/3} + P(\Delta > v) \\ &\leq k_2 v^{1/3}(\max(1, \log(d/v)))^{2/3} + P(\square > v) + o(1), \quad \text{(A.16)}\end{aligned}$$

where $\square$ represents the maximum element between the two matrix Ω_1 and Ω_2, denote as $\|\Omega_1 - \Omega_2\|_{\max}$, where Ω_1 and Ω_2 are defined as:

$$[\Omega_1]_{k,l} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\Theta_k^\top \boldsymbol{x}_{ij}(y_{ij} - \boldsymbol{x}_{ij}^T\boldsymbol{\beta})\Theta_l$$

and

$$\Omega_2 = \sigma^2\Theta$$

Then, from Corollary 3.1 in [10] and Lemma E.2 in [41], we could verify that $\|\Omega_1 - \Omega_2\|_{\max} = O(\sqrt{\frac{\log d}{n_0}} + \frac{\log^2(dn_0)\log d}{n_0})$. With a proper choice of v, e.g, there exists a constant κ and let $v = (\sqrt{\frac{\log d}{n_0}} + \frac{\log^2(dn_0)\log d}{n_0})^{1-\kappa}$, we have $k_2 v^{1/3}(\max(1, \log(d/v)))^{2/3} + P(\square > v) = o(1)$. Next, we would like to associate M with M_0. Similarly, from Theorem 3.2 in [10] and (A.16), we could have:

$$\sup_{\alpha\in(0,1)}|\mathbb{P}(M \leq C_{U^*}(\alpha)) - \alpha| \leq o(1) + v_1\sqrt{max(1, \log(d/v_1))} + \mathbb{P}(\|M - M_0\| > v_1),$$

From the definition of M and M_0, we have:

$$\sqrt{n_0}(M - M_0) = \max_{1\leq k\leq d}\frac{1}{\sqrt{n_0}}|(\hat{\boldsymbol{\Theta}}_k{}^\top - \boldsymbol{\Theta}_k^\top)\boldsymbol{X}^\top\boldsymbol{W}|$$

Then, for any k in $1, \ldots, d$, by Holder inequality and Cauchy–Schwarz inequality, we have:

$$\frac{1}{\sqrt{n_0}}|(\hat{\boldsymbol{w}}_k{}^\top - \boldsymbol{w}_k^\top)\boldsymbol{X}^\top\boldsymbol{W}| \leq \|\hat{\boldsymbol{w}}_k{}^\top - \boldsymbol{w}_k^\top\|_1\|\frac{1}{\sqrt{n_0}}\boldsymbol{X}^\top\boldsymbol{W}\|_\infty \leq \sqrt{s^*}\|\hat{\boldsymbol{w}}_k{}^\top - \boldsymbol{w}_k^\top\|_2\|\frac{1}{\sqrt{n_0}}\boldsymbol{X}^\top\boldsymbol{W}\|_\infty$$

On one hand, from previous proof, we obtain that $\|\hat{\boldsymbol{w}}_k{}^\top - \boldsymbol{w}_k^\top\|_2 = c\cdot\sqrt{\frac{s^*\log d}{mn}}$ when the privacy cost is dominated by statistical error uniformly for k. On the other hand, by the

fact that $\mathbf{\Sigma}$ have bounded maximum eigenvalue and traditional linear regression model, we could apply Bernstein inequality and also obtain that $\|\frac{1}{\sqrt{n_0}}\boldsymbol{X}^\top \boldsymbol{W}\|_\infty$ is $O(\sqrt{\frac{\log d}{n_0}})$. Combine these two results, we could claim that there exist constants k_0 such that:

$$\frac{1}{\sqrt{n_0}}|(\hat{\mathbf{\Theta}}_k{}^\top - \mathbf{\Theta}_k^\top)\boldsymbol{X}^\top \boldsymbol{W}| = k_0 \cdot (s^* \log d/\sqrt{n_0})$$

uniformly for all k, then we can choose v_1 properly such that $\sup_{\alpha\in(0,1)} |\mathbb{P}(M \le C_{U^*}(\alpha)) - \alpha| = o(1)$. At last, we need to relate U^* with U. Our major goal is to prove that $C_U(\alpha)$ and C_{U^*} are close to each other for any $\alpha \in (0,1)$. We first associate U with U'. From the design of private max algorithm, from Lemma 3.4 in [8], suppose l_1 is the element chosen from U' and l_2 is from U without noise injection, we use w to represent the noise injected when we pick the largest value privately, we find that, for any $c > 0$:

$$l_2^2 \le l_1^2 \le (1+c)l_2^2 + 4(1+1/c)\|w\|_\infty^2$$

From Lemma A.1 in [8], we can verify that there exists constant k_0, k_1 such that $\|w\|_\infty^2 \le k_0 \cdot \frac{s^* \log^4 d \log m}{n_0}$. When we choose $c = o(1)$, e.g, $c = k_1 \frac{s^* \log d}{n_0}$, then from the conditions, we could claim that $l_1 = l_2 + o(1)$, also notice that the scale of noise we injected is small, it is easy to verify that $U = U' + o(1)$. The following discussions will be between U' and U^*. Denote $\ominus$ as the symmetric difference, then we have:

$$\begin{aligned}&\mathbb{P}(T \le C_U(\alpha) \ominus T \le C_{U^*}(\alpha))\\ &\le 2\mathbb{P}(C_{U^*}(\alpha - \pi(u)) < T \le C_{U^*}(\alpha + \pi(u))) + \mathbb{P}(C_{U^*}(\alpha - \pi(u)) > C_U(\alpha)) + \mathbb{P}(C_{U^*}(\alpha + \pi(u)) < C_U(\alpha))\end{aligned} \tag{A.17}$$

For the first term in (A.17), define $\pi(u) = u^{1/3}\max(1, \log(d/u))^{2/3}$, then exist a constant k_0, such that:

$$\mathbb{P}(C_{U^*}(\alpha - \pi(u)) < M \le C_{U^*}(\alpha + \pi(u))) \le \mathbb{P}(M \le C_{U^*}(\alpha + \pi(u))) - \mathbb{P}(M \le C_{U^*}(\alpha - \pi(u))) \le k \cdot \pi(u) + o(1)$$

Then, for the second term and third term in (A.17), from Lemma 3.2 in [10], we have:

$$\mathbb{P}(C_{U^*}(\alpha - \pi(u)) > C_U(\alpha)) + \mathbb{P}(C_{U^*}(\alpha + \pi(u)) < C_U(\alpha)) \le 2\mathbb{P}(\|\Omega_1 - \Omega_3\|_{\max} > u),$$

where Ω_3 is defined as:

$$[\Omega_3]_{k,l} = \frac{1}{m}\sum_{i=1}^m n\hat{\mathbf{\Theta}}_k(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top \hat{\mathbf{\Theta}}_l,$$

and Ω_1 is defined the same as we defined before. Then, our major focus is to analyze $\|\Omega_1 - \Omega_3\|_{\max}$, by triangle inequality, we have $\|\Omega_1 - \Omega_3\|_{\max} \le \|\Omega_1 - \Omega_2\|_{\max} + \|\Omega_3 - \Omega_2\|_{\max}$. Since we have analyzed $\|\Omega_1 - \Omega_2\|_{\max}$ before, we will focus on $\|\Omega_3 - \Omega_2\|_{\max}$.

$$\|\Omega_3 - \Omega_2\|_{\max} \le \left\|\frac{1}{m}\sum_{i=1}^m n\hat{\mathbf{\Theta}}(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top \hat{\mathbf{\Theta}} - \sigma^2\hat{\mathbf{\Theta}}\Sigma\hat{\mathbf{\Theta}}\right\|_{\max} + \|\sigma^2\hat{\mathbf{\Theta}}\Sigma\hat{\mathbf{\Theta}} - \sigma^2\mathbf{\Theta}\|_{\max} \tag{A.18}$$

We will analyze the two terms separately. We start from the second term in (A.18), we have:

$$\begin{aligned}
&\|\hat{\boldsymbol{\Theta}}\Sigma\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_{\max} \\
&\leq \|(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta} + \boldsymbol{\Theta})\Sigma(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta} + \boldsymbol{\Theta}) - \boldsymbol{\Theta}\|_{\max} \\
&\leq \|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1^2\|\boldsymbol{\Sigma}\|_{\max} + 2\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1 \\
&\leq k_0\frac{s^{*2}\log d}{n_0} + k_1 s^*\sqrt{\frac{\log d}{n_0}}
\end{aligned}$$

On the other hand, for the first term in (A.18),notice that:

$$\frac{1}{m}\sum_{i=1}^{m} n\hat{\boldsymbol{\Theta}}(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top\hat{\boldsymbol{\Theta}} = \frac{1}{m}\sum_{i=1}^{m} n\hat{\boldsymbol{\Theta}}\boldsymbol{g}_i\boldsymbol{g}_i^\top\hat{\boldsymbol{\Theta}} - n\hat{\boldsymbol{\Theta}}\bar{\boldsymbol{g}}\bar{\boldsymbol{g}}^\top\hat{\boldsymbol{\Theta}}^\top \tag{A.19}$$

Denote the data set on the i-th local machine as $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ and in the linear model, the random noise as $\boldsymbol{W}_i$. Also, we can further decompose the first term by:

$$\begin{aligned}
&\frac{1}{m}\sum_{i=1}^{m} n\boldsymbol{g}_i\boldsymbol{g}_i^\top \\
&= \frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top W_i + \boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top W_i + \boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}]^\top \\
&= \frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top W_i}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top + \frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}]^\top + \frac{2}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top
\end{aligned} \tag{A.20}$$

Then, for the equation (A.19), we have:

$$\begin{aligned}
&\|\frac{1}{m}\sum_{i=1}^{m} n\hat{\boldsymbol{\Theta}}(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top\hat{\boldsymbol{\Theta}} - \sigma^2\hat{\boldsymbol{\Theta}}\Sigma\hat{\boldsymbol{\Theta}}\|_{\max} \\
&\leq \|\hat{\boldsymbol{\Theta}}\|_{\max}\|\frac{1}{m}\sum_{i=1}^{m} n(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top\hat{\boldsymbol{\Theta}} - \sigma^2\Sigma\hat{\boldsymbol{\Theta}}\|_{\max} \\
&\leq \|\hat{\boldsymbol{\Theta}}\|_{\max}^2\|\frac{1}{m}\sum_{i=1}^{m} n(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top - \sigma^2\Sigma\|_{\max}
\end{aligned} \tag{A.21}$$

And, we could insert (A.20) into (A.21),

$$\begin{aligned}
&\|\frac{1}{m}\sum_{i=1}^{m} n(\boldsymbol{g}_i - \bar{\boldsymbol{g}})(\boldsymbol{g}_i - \bar{\boldsymbol{g}})^\top - \sigma^2\Sigma\|_{\max} \\
&\leq \|\frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top W_i}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top - \sigma^2\Sigma\|_{\max} + n\|\bar{\boldsymbol{g}}\bar{\boldsymbol{g}}^\top\|_{\max} + \|\frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})}{n}]^\top\|_{\max}
\end{aligned}$$

$$+\|\frac{2}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top\|_{\max} \tag{A.22}$$

We will analyze the four terms in (A.22) one by one. For the first term, it is quite simple, from the proof of Lemma F.2 in [41], we have the first term is $O_p(\sqrt{\frac{\log d}{m}}+\frac{\log^2(dm)\log d}{m})$. For the second term, we have:

$$n\|\bar{\boldsymbol{g}}\bar{\boldsymbol{g}}^\top\|_{\max} \le n\|\bar{\boldsymbol{g}}\|_\infty^2 = n\|\frac{1}{n_0}\boldsymbol{X}^\top(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\|_\infty^2$$

Also, we have:

$$\begin{aligned}
&\|\frac{1}{n_0}\boldsymbol{X}^\top(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})\|_\infty \\
&\le \|\frac{1}{n_0}\boldsymbol{X}^\top(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta})\|_\infty + \|\frac{1}{n_0}\boldsymbol{X}^\top\boldsymbol{X}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_\infty \\
&\le \|\frac{1}{n_0}\boldsymbol{X}^\top\boldsymbol{W}\|_\infty + \|(\hat{\Sigma}-\Sigma)(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\|_\infty + \|\Sigma\|_{\max}\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_1 \\
&\le k_0(\sqrt{\frac{\log d}{n_0}}) + k_1(\sqrt{\frac{\log d}{n_0}})\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_1 + \|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_1 \\
&\le k_0\sqrt{\frac{\log d}{n_0}} + k_1\frac{s^*\log d}{n_0} + k_2 s^*\sqrt{\frac{\log d}{n_0}}
\end{aligned} \tag{A.23}$$

Thus, for the second term, we can obtain that $n\|\bar{\boldsymbol{g}}\bar{\boldsymbol{g}}^\top\|_{\max} \le k_0 s^{*2}\log d/m + k_1 s^{*2}\log^2 d/m^2 n$. For the third term, we have:

$$\begin{aligned}
&\|\frac{1}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}]^\top\|_{\max} \\
&\le \frac{1}{m}\sum_{i=1}^{m} n\|[\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}]^\top\|_{\max} \\
&\le \frac{1}{m}\sum_{i=1}^{m} n\|[\frac{\boldsymbol{X}_i^\top\boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}]\|_\infty^2 \\
&\le \frac{1}{m}\sum_{i=1}^{m} n(\|\hat{\Sigma}_i-\Sigma\|_{\max}+\|\Sigma\|_{\max})^2\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_1^2 \\
&\le \frac{1}{m}\sum_{i=1}^{m} 2n(\|\hat{\Sigma}_i-\Sigma\|_{\max}^2+\|\Sigma\|_{\max}^2)\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_1^2 \\
&\le \frac{1}{m}\sum_{i=1}^{m} 2n(O(\sqrt{\frac{\log d}{n}})+O(1))\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_1^2 \\
&\le k_0 s^{*2}\frac{\log d}{m}
\end{aligned} \tag{A.24}$$

For the fourth term, we could apply Cauchy-Schwarz inequality, which give us the result:

$$
\begin{aligned}
&\|\frac{2}{m}\sum_{i=1}^{m} n[\frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top\|_{\max} \\
&\leq \frac{2}{m}\sum_{i=1}^{m} n\|[\frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}][\frac{\boldsymbol{X}_i^\top W_i}{n}]^\top\|_{\max} \\
&\leq \frac{2}{m}\sum_{i=1}^{m} n\|\frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{n}\|_\infty\|\frac{\boldsymbol{X}_i^\top W_i}{n}\|_{\max} \\
&\leq \frac{2}{m}\sum_{i=1}^{m} n\|\frac{\boldsymbol{X}_i^\top \boldsymbol{X}_i}{n}\|_{\max}\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_1\|\frac{\boldsymbol{X}_i^\top W_i}{n}\|_\infty \\
&\leq k_0 n\cdot\frac{s^*\log d}{n_0} \\
&\leq k_0\frac{s^*\log d}{m}
\end{aligned}
\tag{A.25}
$$

We could combine the result in (A.23), (A.24), (A.25) and insert into (A.22) and into (A.21). We could finally get the first term of (A.18) has an order of $O(\sqrt{\frac{\log d}{n_0}}+\frac{s^*\log d}{n_0}+\frac{{s^*}^2\log d}{m})$. Insert this result into (A.17), when u is chosen properly, we could verify that $\sup_{\alpha\in(0,1)}|\mathbb{P}(T\leq C_U(\alpha))-\alpha| = o(1)$, which finishes the proof.

A.7 Proof of Theorem 7

The proof of theorem 7 is quite straight forward. We could decompose true $\boldsymbol{\beta}_i = \boldsymbol{u}+\boldsymbol{v}_i$. Then, $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{u}}+\hat{\boldsymbol{v}}_i$. Thus, from the result of estimation, we could get the result that:

$$
\|\boldsymbol{u}-\hat{\boldsymbol{u}}\|_2^2 \leq c_0\frac{s_0\log d}{mn}+c_2\frac{{s_0}^2\log d^2\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2},
$$

and

$$
\|\hat{\boldsymbol{v}}_i-\boldsymbol{v}_i\|_2^2 \leq c_1\frac{s_1\log d}{n}+c_3\frac{{s_1}^2\log d^2\log(1/\delta)\log^3 n}{n^2\epsilon^2}
$$

Also, combing the above two results with the inequality that $\|\hat{\boldsymbol{\beta}}_i-\boldsymbol{\beta}_i\|_2 \leq \|\hat{\boldsymbol{v}}_i-\boldsymbol{v}_i\|_2+\|\boldsymbol{u}-\hat{\boldsymbol{u}}\|_2$ gives the proof of theorem 7. □

A.8 Proof of Theorem 8

The proof of Theorem 8 follows the proof of and Theorem 4 and Theorem 5. We follow the proof of Theorem 4 until (A.8). The analysis for the second term and the third term for (A.8) stays the same. We will analyze the three terms in (A.8) one by one. For the first term, in the same manner, we could decompose this term as:

$$
\sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{e}_j^\top)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}}) = \sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{\Theta}_j^\top\hat{\boldsymbol{\Sigma}}+\boldsymbol{\Theta}_j^\top\hat{\boldsymbol{\Sigma}}-\boldsymbol{e}_j^\top)(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}^{\mathrm{u}})
$$

$$= \sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_j^\top \hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) + \sqrt{n}(\boldsymbol{\Theta}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \tag{A.26}$$

For the first term in (A.26), we could further decompose this term from $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$:

$$\begin{aligned}
\sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_j^\top \hat{\boldsymbol{\Sigma}})(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) &= \sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top - \boldsymbol{\Theta}_j^\top)\hat{\Sigma}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \\
&\leq \sqrt{n}\lambda_s(\hat{\Sigma})\|\hat{\boldsymbol{\Theta}}_j - \boldsymbol{\Theta}_j\|_2 |\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2 \\
&\leq o(1) + \frac{\gamma\mu_s^2}{\nu_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2 n^{3/2}\epsilon^2} + \frac{\gamma\mu_s^2}{\nu_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^{3/2}\epsilon^2}
\end{aligned} \tag{A.27}$$

Thus, for the second term of (A.26), by Lemma A.3, we have:

$$\begin{aligned}
\sqrt{n}(\boldsymbol{\Theta}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) &\leq \sqrt{n}\|\boldsymbol{\Theta}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top\|_\infty \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_1 \\
&\leq k\sqrt{n}\sqrt{\frac{\log d}{mn}}\sqrt{s}\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2 \\
&\leq k \cdot \sqrt{n}\sqrt{\frac{s\log d}{mn}} \cdot \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}\|_2 \\
&\leq o(1) + \frac{\gamma\mu_s^2}{\nu_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2 n^{3/2}\epsilon^2} + \frac{\gamma\mu_s^2}{\nu_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^{3/2}\epsilon^2}
\end{aligned} \tag{A.28}$$

Then, combining (A.27) and (A.28), we have that:

$$\sqrt{n}(\hat{\boldsymbol{\Theta}}_j^\top \hat{\boldsymbol{\Sigma}} - \boldsymbol{e}_j^\top)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{\mathrm{u}}) \leq \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2 n^{3/2}\epsilon^2} + \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^{3/2}\epsilon^2}$$

Then, insert the result into (A.8), we have:

$$\sqrt{n}\left(\hat{\beta}_j - \beta_j - \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)\log^3 mn}{m^2 n^2 \epsilon^2} - \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)\log^3 n}{n^2\epsilon^2}\right) = \frac{1}{\sqrt{n}}\hat{\boldsymbol{\Theta}}_j^\top \boldsymbol{X}^\top \boldsymbol{W} + \sqrt{n}E_3 \tag{A.29}$$

From the result in (A.29), notice that the right hand side asymptotically follows a distribution of $N(0, \sigma^2 \hat{\boldsymbol{\Theta}}_j^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_j + \frac{8\Delta_1^2 \log(1/\delta)}{n\epsilon^2})$. Also, by the concentration of Gaussian distribution, we have with high probability, $E_2 \leq \frac{8\Delta_1^2 \log(1/\delta)}{n\epsilon^2})$. Thus, we have:

$$\sqrt{n}\left(\hat{\beta}_j - \beta_j - \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_1^2 \log^2 d \log(1/\delta)}{m^2 n^2 \epsilon^2} - \frac{2\gamma\mu_s^2}{\nu_s^2}\frac{s_0^2 \log^2 d \log(1/\delta)}{n^2\epsilon^2}\right) \Big/ \sqrt{\sigma^2 \hat{\boldsymbol{\Theta}}_j^\top \hat{\boldsymbol{\Sigma}} \hat{\boldsymbol{\Theta}}_j + \frac{8\Delta_1^2 \log(1/\delta)}{n\epsilon^2}} \sim N(0,1)$$

Then, we could replace μ_s, ν_s with the estimation $\hat{\mu}_s, \hat{\nu}_s$ introduced in Algorithm 5, the constant could be scaled to one given the tail bound of Laplace random variable. Also, for the estimation of σ, according to the assumption, we have $\hat{\sigma} = \sigma + o(1)$. For simplicity,

we denote $a = \frac{2k\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_1^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2} + \frac{2k\hat{\mu}_s^2}{\hat{\nu}_s^2}\frac{s_0^2\log^2 d\log(1/\delta)\log^3 n}{n^2\epsilon^2}$, the confidence is given by:

$$J_j(\alpha) =$$
$$[\hat{\beta}_j - a - \frac{\sigma\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\mathbf{\Theta}}_j^\top\hat{\mathbf{\Sigma}}\hat{\mathbf{\Theta}}_j + \frac{8\Delta_1^2\log(1/\delta)}{n\epsilon^2}}, \hat{\beta}_j + a + \frac{\sigma\Phi^{-1}(1-\alpha/2)}{\sqrt{n}}\sqrt{\hat{\mathbf{\Theta}}_j^\top\hat{\mathbf{\Sigma}}\hat{\mathbf{\Theta}}_j + \frac{8\Delta_1^2\log(1/\delta)}{n\epsilon^2}}]$$

A.9 Proof of Theorem 9

In this proof, we first need to show that our algorithm is (ϵ, δ) private. We assume in two data sets, the adjacent data set is different in $(\boldsymbol{x}_{ij}, y_{ij})$ and $(\boldsymbol{x}'_{ij}, y'_{ij})$. Then, we have:

$$\begin{aligned}
\|\frac{1}{\sqrt{n}}\hat{\mathbf{\Theta}}\boldsymbol{x}_{ij}e_j - \frac{1}{\sqrt{n}}\hat{\mathbf{\Theta}}\boldsymbol{x}'_{ij}e_j\|_\infty &\leq \frac{2}{\sqrt{n}}\|\hat{\mathbf{\Theta}}\boldsymbol{x}_{ij}e_j\|_\infty \\
&\leq \frac{2}{\sqrt{n}}\|\hat{\mathbf{\Theta}}\boldsymbol{x}_{ij}\|_\infty\|e_j\|_\infty \\
&\leq \frac{2\sqrt{\log n}}{\sqrt{n}}\|\hat{\mathbf{\Theta}}\|_1\|\boldsymbol{x}_{ij}\|_\infty \\
&\leq 2\sqrt{\frac{\log n}{n}}c_x\sqrt{s}c_1
\end{aligned}$$

According to the choice of B_5, the privacy could be guaranteed. Then, let us start the proof of consistency. In this proof specifically, we define $U' = \max_{k\in G}\hat{\mathbf{\Theta}}_k^T\frac{1}{\sqrt{n}}\sum_{i=1}^n\boldsymbol{x}_{ij}e_j$ and also, we define:

$$M_0 = \max_{k\in G}\frac{1}{\sqrt{n}}\sum_{j=1}^n\xi_{jk},$$

where ξ_j follows a Gaussian Distribution $N(0, \Theta_k^\top\Sigma\Theta_k\sigma^2)$. Also, we define the α-quantile of M_0 as $U_{M_0}(\alpha)$. Then, we could start the proof.

We are aiming at proving $\sup_{\alpha\in(0,1)}|\mathbb{P}(M \leq C_U(\alpha)) - \alpha| = o(1)$. First, we could prove that $C_U(\alpha)$ and $C_{U'}$ are close to each other for any $\alpha \in (0,1)$. From the design of private max algorithm, from Lemma 3.4 in [8], suppose l_1 is the element chosen from U' and l_2 is from U without noise injection, we use w to represent the noise injected when we pick the largest value privately, we find that, for any $c > 0$:

$$l_2^2 \leq l_1^2 \leq (1+c)l_2^2 + 4(1+1/c)\|w\|_\infty^2$$

From Lemma A.1 in [8], we can verify that there exists constant k_0, k_1 such that $\|w\|_\infty^2 \leq k_0\cdot\frac{s^*\log^4 d\log n}{n}$. When we choose $c = o(1)$, e.g, $c = k_1\frac{s^*\log d}{n}$, then from the conditions, we could claim that $l_1 = l_2 + o(1)$, also notice that the scale of noise we injected is small, it is easy to verify that $U = U' + o(1)$. The following discussions will be between U' and U_{T_0}.

Motivated by the proof of Theorem 3.1 and Theorem 3.2 in [10], our proof will be divided into two major parts, to measure the closeness between U' and U_{M_0} and measure

the closeness between T and T_0. We start from the measurement between M and M_0. From the definition that $M(\hat{\boldsymbol{\beta}}^{(i)}) = \max_{k\in G}\sqrt{n}(\hat{\boldsymbol{\beta}}_k^{(i)} - \boldsymbol{\beta}_k^{(i)})$, we notice that for each k in $1, 2, \ldots, d$, we have:

$$\sqrt{n}(\hat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}^{(i)}) = \frac{1}{\sqrt{n}}\hat{\Theta}X_i^\top \boldsymbol{W}_i + (\hat{\Theta}\Sigma - I)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + \sqrt{n}E$$

Then,

$$|M - M_0| \leq (\|\frac{1}{\sqrt{n}}\hat{\Theta}X_i^\top \boldsymbol{W}_i\|_\infty - \|\frac{1}{\sqrt{n}}\sum_{j=1}^{n}\xi_j\|_\infty) + \|(\hat{\Theta}\Sigma - I)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + \sqrt{n}E\|_\infty \tag{A.30}$$

We analyze the two parts in (A.30) separately. For the first term in (A.30). First, from Lemma 1.1 in [42], we could obtain the result that for any z, $\sup_z |P(\|\frac{1}{\sqrt{n}}\boldsymbol{\Theta}\boldsymbol{X}_i^\top \boldsymbol{W}_i\|_\infty \leq z) - P(M_0 \leq z)| \leq c_0 \cdot \frac{1}{n^{c_1}}$, where c_0 and c_1 are constants. Also,

$$\begin{aligned}
\|\frac{1}{\sqrt{n}}\hat{\boldsymbol{\Theta}}\boldsymbol{X}_i^\top \boldsymbol{W}_i\|_\infty - \|\frac{1}{\sqrt{n}}\boldsymbol{\Theta}\boldsymbol{X}_i^\top \boldsymbol{W}_i\|_\infty &\leq \frac{1}{\sqrt{n}}\|\hat{\boldsymbol{\Theta}}\boldsymbol{X}_i^\top \boldsymbol{W}_i - \boldsymbol{\Theta}\boldsymbol{X}_i^\top \boldsymbol{W}_i\|_\infty \\
&\leq \|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1 \cdot \|\frac{1}{\sqrt{n}}\boldsymbol{X}_i^\top \boldsymbol{W}_i\|_\infty \\
&\leq c \cdot s^* \sqrt{\frac{\log d}{n}}\sqrt{\log d} \leq c \cdot s^* \log d \sqrt{\frac{1}{n}} = o(1)
\end{aligned}$$

On the other hand, we also notice that for the second term in (A.30), following the proof in Theorem 4, we also know that the second part is $o(1)$, hence finishes the first part of the proof. In the second part of the proof. By the arguments in the proof of Theorem 3.2 in [10], we have for any v:

$$\sup_{\alpha\in(0,1)} |\mathbb{P}(T \leq C_{U'}(\alpha)) - \alpha| \leq c_0\frac{1}{n^{c_1}} + c_2 v^{1/3}(1 \vee \log(d/v))^{2/3} + P(\square > v),$$

where $\square = \max_{k,l} \hat{\Theta}_k^\top \hat{\Sigma}\hat{\boldsymbol{\Theta}}_l - \boldsymbol{\Theta}_k^\top \boldsymbol{\Sigma}\boldsymbol{\Theta}_l$.Then, we have:

$$\begin{aligned}
&\|\hat{\boldsymbol{\Theta}}^\top \hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^\top \boldsymbol{\Sigma}\boldsymbol{\Theta}\|_{\max} \\
&\leq \|\hat{\boldsymbol{\Theta}}^\top \hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Theta}} - \hat{\boldsymbol{\Theta}}^\top \boldsymbol{\Sigma}\hat{\boldsymbol{\Theta}}\|_{\max} + \|\hat{\boldsymbol{\Theta}}\Sigma\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_{\max} \\
&\leq \|\hat{\boldsymbol{\Theta}}\|_\infty \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|_{\max}\|\hat{\boldsymbol{\Theta}}\|_1 + \|(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta} + \boldsymbol{\Theta})\Sigma(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta} + \boldsymbol{\Theta}) - \boldsymbol{\Theta}\|_{\max} \\
&\leq L^2\sqrt{\frac{\log d}{n_0}} + \|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1^2\|\boldsymbol{\Sigma}\|_{\max} + 2\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1 \\
&\leq k_0\frac{s^{*2}\log d}{n_0} + k_1 s^*\sqrt{\frac{\log d}{n_0}},
\end{aligned}$$

where k_0, k_1 are constants. Then, with a proper choice of v, we could claim that $\sup_{\alpha\in(0,1)} |\mathbb{P}(T \leq C_{U'}(\alpha)) - \alpha| = o(1)$, which finishes the proof.

B Appendix

In this section, we will give proofs of the corollary in the main proof. We will introduce them one by one:

B.1 Proof of corollary 1

Following the proof of Theorem 1 in [2], we can gain the upper bound of $\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})$, when we consider the central DP:

$$\frac{1}{k}\left(\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})\right)^2 \leq 7\sum_{t=1}^{m}\mathbb{E}_A\left[\sum_{i=1}^{k}\int_{\mathcal{Z}}\frac{(\mathbb{E}_{\boldsymbol{p}_a^{\otimes n}}[\mathcal{W}(z\mid X)] - \mathbb{E}_{\boldsymbol{p}_{a^{\oplus i}}^{\otimes n}}[\mathcal{W}(z\mid X))]^2}{\mathbb{E}_{\boldsymbol{p}_a}[\mathcal{W}(z\mid X)]}d\mu\right]$$

Also notice that:

$$\mathbb{E}_{\boldsymbol{p}_{a^{\oplus i}}^{\otimes n}}[\mathcal{W}(z\mid X))] = \mathbb{E}_{\boldsymbol{p}_a^{\otimes n}}\left[\frac{d\boldsymbol{p}_{a^{\oplus i}}^{\otimes n}}{d\boldsymbol{p}_a^{\otimes n}}\mathcal{W}(z\mid X))\right] = \mathbb{E}_{\boldsymbol{p}_a}(1+q_{a,i}\phi_{a,i}(X))^n\cdot\mathcal{W}(z\mid X)$$

The last equation is from the definition of condition 1. Also, by the inequality that we can find constants c_0, c_1 that when $x>0$ and $x \asymp 1/n$, $(1+x)^n \leq 1+c_0\cdot nx$ and $(1-x)^n \leq 1-c_0\cdot nx$. If we have $|q_{a,i}\phi_{a,i}(X)| \asymp 1/n$. we could find a constant c_2, such that:

$$\frac{1}{k}(\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m}))^2 \leq c_2 q^2 n^2 \sum_{t=1}^{m}\mathbb{E}_A\left[\sum_{i=1}^{k}\int_{\mathcal{Z}}\frac{\mathbb{E}_{\boldsymbol{p}_A^{\otimes n}}[\phi_{a,i}(X)\mathcal{W}(z\mid X)]^2}{\mathbb{E}_{\boldsymbol{p}_A^{\otimes n}}[\mathcal{W}(z\mid X)]}d\mu\right]$$

which finishes the proof of corollary 1.

B.2 Proof of Corollary 2

We continue to show the proof of Corollary 2. First, from Theorem 2 in [2], we get a direct result when condition 2 is satisfied, given all the conditions in corollary 2 hold, we have:

$$\left(\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})\right)^2 \leq \frac{7}{k}q^2n^2\sum_{t=1}^{m}\max_{a\in\mathcal{A}}\int_{\mathcal{Z}}\frac{\mathrm{Var}_{\boldsymbol{p}_a}[\mathcal{W}(z\mid X)]}{\mathbb{E}_{\boldsymbol{p}_a}[\mathcal{W}(z\mid X)]}d\mu$$

Then, the focus of the proof of this corollary is on the calculation of $\int_{\mathcal{Z}}\frac{\mathrm{Var}_{\boldsymbol{p}_a}[\mathcal{W}(z|X)]}{\mathbb{E}_{\boldsymbol{p}_a}[\mathcal{W}(z|X)]}d\mu$ when the channel $\mathcal{W}$ is a privacy constraint channel $\mathcal{W}^{priv}$. For simplicity, we denote $L(\boldsymbol{z},\boldsymbol{X}) = log\mathcal{W}^{priv}(\boldsymbol{z}|\boldsymbol{X})$, where $\boldsymbol{z}\in\mathbb{R}^d$ and $\boldsymbol{X}\in\mathbb{R}^{n\times d}$. Then, notice that $\mathcal{W}^{priv}$ is ϵ-differentially private constraint, for two adjacent dataset $\boldsymbol{X}$ and $\boldsymbol{X'}$, we have:

$$|L(\boldsymbol{z},\boldsymbol{X}) - L(\boldsymbol{z},\boldsymbol{X'})| \leq \epsilon$$

By McDiarmid's inequality, we could claim that L is $\sqrt{n}\epsilon$- subGaussian. So we could find a constant c, which satisfies that:

$$\mathbb{E}[e^{2L}] \leq c \cdot e^{2\mathbb{E}[L]} \cdot e^{2n\epsilon^2}$$

Then, by Jensen inequality, we have:

$$\mathbb{E}[e^{2L}] \leq c \cdot (e^{E[L]})^2 \cdot e^{2n\epsilon^2}$$

Thus, we have:

$$\frac{\mathrm{Var}[\mathcal{W}^{priv}(\boldsymbol{z}|\boldsymbol{X})]}{\mathbb{E}[\mathcal{W}^{priv}(\boldsymbol{z}|\boldsymbol{X})]^2} = \frac{\mathbb{E}[\mathcal{W}^{priv}(\boldsymbol{z}|\boldsymbol{X})^2]}{(\mathbb{E}[\mathcal{W}^{priv}(\boldsymbol{z}|\boldsymbol{X})])^2} - 1 = \frac{\mathbb{E}[e^{2L}]}{(\mathbb{E}[e^L])^2} - 1 \leq e^{2n\epsilon^2} - 1$$

Thus, we have:

$$\begin{aligned}
\left(\frac{1}{k}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m})\right)^2 \leq & \frac{7}{k}\alpha^2 n^2 \sum_{t=1}^{m} \max_{a\in\mathcal{A}} \int_{\mathcal{Z}} \frac{\mathrm{Var}_{\boldsymbol{p}_a}[\mathcal{W}(z \mid X)]}{\mathbb{E}_{\boldsymbol{p}_a}[\mathcal{W}(z \mid X)]^2} \cdot \mathbb{E}_{\boldsymbol{p}_a}[W(z \mid X)] d\mu \\
\leq & \frac{7}{k}\alpha^2 n^2 (e^{2n\epsilon^2} - 1) \sum_{t=1}^{m} \max_{a\in\mathcal{A}} \int_{\mathcal{Z}} \mathbb{E}_{\boldsymbol{p}_a}[\mathcal{W}(z \mid X)] d\mu \\
\leq & \frac{7}{k} q^2 m n^2 (e^{2n\epsilon^2} - 1)
\end{aligned}$$

which finishes the proof of corollary 2.

B.3 Proof of corollary 3

For an (n, ρ)-estimator $\hat{\theta}$ of the true parameter under ℓ_p loss, we define $\hat{A}$ for A as

$$\hat{A} = \underset{a\in\mathcal{A}}{\mathrm{argmin}} \|\theta_a - \hat{\theta}\|_p.$$

Then, by the triangle inequality, we have

$$\left\|\theta_A - \theta_{\hat{A}}\right\|_p \leq \left\|\theta_A - \hat{\theta}\right\|_p + \left\|\theta_{\hat{A}} - \hat{\theta}\right\|_p \leq 2\left\|\hat{\theta} - \theta_A\right\|_p.$$

Because $\hat{\theta}$ is an (n, ρ)-estimator under ℓ_p loss, we have,

$$\mathbb{E}_Z[\mathbb{E}_{\boldsymbol{p}_Z}[\left\|\theta_Z - \theta_{\hat{Z}}\right\|_p^p]] \leq 2^p\rho^p\mathbb{P}[\boldsymbol{p}_Z \in \mathcal{P}_\Theta] + \max_{z\neq z'} \|\theta_z - \theta_{z'}\|_p^p \mathbb{P}[\boldsymbol{p}_Z \notin \mathcal{P}_\Theta] \tag{B.1}$$

$$\leq 2^p\rho^p + 4^p\rho^p \frac{1}{\tau} \cdot \frac{\tau}{4} \tag{B.2}$$

$$\leq \frac{3}{4} 4^p \epsilon^p, \tag{B.3}$$

using the fact that $\mathbb{P}[\boldsymbol{p}_A \in \mathcal{P}_\Theta] \geq 1 - \tau/4$ and condition 4. Also, from condition 4, Next, combining condition 4 and B.3, we could have: $\frac{1}{\tau k}\sum_{i=1}^{k} \mathbb{P}[A_i \neq \hat{A}_i] \leq \frac{3}{4}$. Also, since the

Markov relation $A_i - X^m - Z^m - \hat{A}_i$ holds for all i, by the standard relation between total variation distance and hypothesis testing, and also the definition of τ to be less than $1/2$, we have:

$$\begin{aligned}\mathbb{P}[A_i \neq \hat{A}_i] &\geq \tau\mathbb{P}[\hat{A}_i = -1|A_i = 1] + (1-\tau)\mathbb{P}[\hat{A}_i = 1|A_i = -1] \\ &\geq \tau(\mathbb{P}[\hat{A}_i = -1|A_i = 1] + \mathbb{P}[\hat{A}_i = 1|A_i = -1]) \\ &\geq \tau(1 - d_{TV}(\boldsymbol{p}_{+i}^{X^m}, \boldsymbol{p}_{-i}^{X^m})) \\ &\geq \tau(1 - 1/n \cdot d_{TV}(\boldsymbol{p}_{+i}^{Z^m}, \boldsymbol{p}_{-i}^{Z^m}))\end{aligned}$$

The last inequality uses the definition of total variation, because Z^m is generated by X^m from the privacy constraint channel W^{priv}, so for each dataset X_i, $i = 1, 2, \ldots, m$ on the i-th machine, let X_{ijk} be the dataset which changes the order of X_{ij} and X_{ik}, then for any $z \in \mathcal{Z}$,$W^{priv}(z|X_i) = W^{priv}(z|X_{ijk})$. Thus, by the definition of total variation, we could verify that $d_{TV}(\boldsymbol{p}_{+i}^{X_i}, \boldsymbol{p}_{-i}^{X_i}) = 1/n \cdot d_{TV}(\boldsymbol{p}_{+i}^{Z_i}, \boldsymbol{p}_{-i}^{Z_i})$. Summing over $1 \leq i \leq k$ and combining it with the previous bound, we obtain

$$\frac{3}{4} \geq \frac{1}{\tau k}\sum_{i=1}^{k}\mathbb{P}[A_i \neq \hat{A}_i] \geq 1 - \frac{1}{nk}\sum_{i=1}^{k} d_{TV}(\boldsymbol{p}_{+i}^{Z^n}, \boldsymbol{p}_{-i}^{Z^n})$$

which finishes the proof of corollary 3.

B.4 Proof of Lemma 4.1

Proof of Lemma 4.1: First, we would like to show that our algorithm is (ϵ, δ)-differentially private. For two adjacent data sets, we have:

$$\begin{aligned}&\frac{1}{mn}|(\pi_R(y_i) - x_i^T\hat{\boldsymbol{\beta}})^2 - (\pi_R(y_i') - x_i'^T\hat{\boldsymbol{\beta}})^2| \\ &\leq \frac{2}{mn}(\pi_R(y_i) - x_i^T\hat{\boldsymbol{\beta}})^2 \\ &\leq \frac{4}{mn}(\pi_R(y_i)^2 + (x_i^T\hat{\boldsymbol{\beta}})^2) \leq \frac{4}{mn}(R^2 + sc_0^2c_x^2)\end{aligned}$$

From the definition of Gaussian Mechanism, we could claim that our algorithm is (ϵ, δ)-differential private. Then, for the convergence rate of our estimated σ, from our algorithm, first, we observe with the choice of R, we claim that with high prob, we have $\pi_R(Y) = Y$. Therefore, we have:

$$\begin{aligned}|\sigma^2 - \hat{\sigma}^2| &\leq |\frac{1}{mn}\|\boldsymbol{X\beta} + \boldsymbol{W} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\|_2^2 - \sigma^2| + |E| \\ &\leq |\frac{1}{mn}\boldsymbol{W}^T\boldsymbol{W} - \sigma^2| + (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\hat{\Sigma}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) + \frac{1}{mn}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\boldsymbol{X}^T\boldsymbol{W} + |E|\end{aligned}$$

For the first term, we could obtain that $|\frac{1}{mn}\boldsymbol{W}^T\boldsymbol{W} - \sigma^2| = O(\frac{1}{\sqrt{mn}})$ Also, we have:

$$(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\hat{\Sigma}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \leq \lambda_s(\Sigma)\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_2^2$$

$$\leq cL\left(\frac{s\log d}{mn}+\frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2}\right)$$

Then, from Bernstein inequality, we could obtain that:

$$\|\frac{1}{mn}\boldsymbol{X}^T\boldsymbol{W}\|_\infty = c_1\cdot\sqrt{\frac{\log d}{mn}}$$

Therefore, we claim that:

$$\begin{aligned}
&\frac{1}{mn}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\boldsymbol{X}^T\boldsymbol{W}\\
&\leq \frac{1}{mn}\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_1\|\boldsymbol{X}^T\boldsymbol{W}\|_\infty\\
&\leq c_2\frac{\sqrt{s}}{mn}\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_2\|\boldsymbol{X}^T\boldsymbol{W}\|_\infty\\
&\leq c_3\sqrt{s}\left(\sqrt{\frac{s\log d}{mn}}+\frac{s\log d\sqrt{\log(1/\delta)}\log^{3/2} mn}{mn\epsilon}\right)\sqrt{\frac{\log d}{mn}}\\
&= O_p\left(\frac{s\log d}{mn}+\frac{s\log d\sqrt{\log(1/\delta)}\log^{3/2} mn}{mn\epsilon}\cdot\sqrt{\frac{s\log d}{mn}}\right)\\
&= O_p\left(\frac{s\log d}{mn}+\frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2}\right)
\end{aligned}$$

Also, from our algorithm, we have $E\sim N(0,2B_2^2\log(1.25/\delta)/\epsilon^2)$. Then,

$$|E| = \frac{2B_2^2\log(1/\delta)}{\epsilon^2}|N(0,1)| = c_4\frac{R^4+s^2c_0^4c_x^4\log(1/\delta)}{m^2n^2\epsilon^2} = c_5\cdot\frac{s^2\log^2 d\log(1/\delta)\log^3 mn}{m^2n^2\epsilon^2},$$

by observing that $c_x = O(\sqrt{\log d})$. Combining above inequalities, we have reached our conclusion. Therefore, we finish our proof.

B.5 Proof of Lemma 4.2

Proof of Lemma 4.2: It is not difficult to verify the privacy conditions. Then, from the theory of covering number, we could find n_1 vectors $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_{n_1}$, such that for each s-sparse unit vector $\boldsymbol{v}$, we have $\|\boldsymbol{v}-\boldsymbol{v}_i\|\leq 1/9$. Thus, we have:

$$\begin{aligned}
\lambda_s-\hat{\lambda}_s &= \boldsymbol{v}^*\hat{\boldsymbol{\Sigma}}\boldsymbol{v}^*-\boldsymbol{v}_i\hat{\boldsymbol{\Sigma}}\boldsymbol{v}_i\\
&\leq \boldsymbol{v}^*\hat{\boldsymbol{\Sigma}}(\boldsymbol{v}^*-\boldsymbol{v}_i)+(\boldsymbol{v}^*-\boldsymbol{v}_i)\hat{\boldsymbol{\Sigma}}\boldsymbol{v}_i\\
&\leq \frac{2}{9}\lambda_{2s}\\
&\leq 8/9\lambda_s
\end{aligned}$$

Note that the second last inequality is a direct result of Cauchy inequality and the last inequality holds because let $\boldsymbol{v}^{*'}$ be the corresponding eigenvector of λ_{2s}, then we could

break this eigenvector to two s-sparse vectors $\boldsymbol{v}_1'$ and $\boldsymbol{v}_2'$ such that $\boldsymbol{v}^{*'} = \boldsymbol{v}_1' + \boldsymbol{v}_2'$, then $\lambda_{2s} = (\boldsymbol{v}_1' + \boldsymbol{v}_2')^T \hat{\boldsymbol{\Sigma}} (\boldsymbol{v}_1' + \boldsymbol{v}_2') \leq 4\lambda_s$.
Also, notice that for the noise ξ, by the concentration of Laplace distribution, we could find a constant c such that $\xi \leq cs \log d/\sqrt{n} = o(1)$ with high probability. By the definition of λ_s, we conclude the proof. $\square$