Better Locally Private Sparse Estimation Given Multiple Samples Per User

Yuheng Ma [1] **Ke Jia** [1] **Hanfang Yang** [2 1]

Abstract

Previous studies yielded discouraging results for item-level locally differentially private linear regression with s^*-sparsity assumption, where the minimax rate for nm samples is $\mathcal{O}(s^*d/nm\varepsilon^2)$. This can be challenging for high-dimensional data, where the dimension d is extremely large. In this work, we investigate user-level locally differentially private sparse linear regression. We show that with n users each contributing m samples, the linear dependency of dimension d can be eliminated, yielding an error upper bound of $\mathcal{O}(s^{*2}/nm\varepsilon^2)$. We propose a framework that first selects candidate variables and then conducts estimation in the narrowed low-dimensional space, which is extendable to general sparse estimation problems with tight error bounds. Experiments on both synthetic and real datasets demonstrate the superiority of the proposed methods. Both the theoretical and empirical results suggest that, with the same number of samples, locally private sparse estimation is better conducted when multiple samples per user are available.

1. Introduction

Local differential privacy (LDP) (Kairouz et al., 2014; Duchi et al., 2018), a variant of differential privacy (DP) (Dwork et al., 2006), has gained considerable attention in recent years. LDP assumes that each sample is possessed by a data holder, who privatizes their data before it is collected by the curator. Offering a stronger sense of privacy protection compared to central DP, learning under LDP often encounters challenges such as slow convergence, high demand for local machine capacity, and limited accessibility to basic techniques (Duchi et al., 2018; Tramèr et al., 2022; Ma et al., 2024b), which obstruct the theoretical analysis and practical implementation of LDP learning.

[*]Equal contribution [1]School of Statistics, Renmin University of China [2]Center for Applied Statistics, Renmin University of China. Correspondence to: Hanfang Yang <hyang@ruc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Fortunately, in some scenarios, each user may possess multiple samples, which can serve as a way to overcome these difficulties. This is known as user-level LDP (ULDP) (Acharya et al., 2023; Bassily & Sun, 2023). Research has demonstrated performance improvement in intentionally designed models when each user has multiple samples, from both the central DP perspective (Liu et al., 2020; Ghazi et al., 2021; Levy et al., 2021; Narayanan et al., 2022; Ghazi et al., 2023) and the LDP perspective (Girgis et al., 2022; Acharya et al., 2023; Bassily & Sun, 2023). In most cases (for ULDP), the improvement lies in the effective sample size: if there are n users with m samples and privacy budget ε, the problem is as tractable as having nm users with one sample and privacy budget ε. See Table 1 for a summary.

We proceed to ask the following question: *Besides effective sample size, does having multiple samples per user offer benefits?* If the answer to this question is affirmative, it holds practical significance. For instance, when designing data collection schemes, the primary focus should be on users capable and willing to provide multiple samples. Moreover, if a significant number of users lack trust in the data collector but are willing to share information within small groups (such as family or company), then better mechanisms can be devised for conducting the learning process.

In this work, we offer an affirmative response to the question from the perspective of sparse estimation. Sparse estimation stands as a crucial task in modern machine learning, especially when dealing with high-dimensional data where structured assumptions like sparsity can significantly enhance performance. Particularly, we study sparse linear regression. We first elucidate why the minimax lower bound fails to hold when each user possesses multiple samples and provide a lower bound for ULDP (Theorem 2.4). Subsequently, we introduce an algorithm structured as follows: half of the users perform local variable selection and aggregate their findings to identify the support of non-zero variables. Under mild assumptions, we establish theoretical guarantees for both local selection (Proposition 3.2) and aggregation (Proposition 3.3). Then, to conduct estimation on the narrowed space, we propose a sub-optimal multi-round protocol (Theorem 3.4) and a two-round protocol (Theorem 3.6). The latter achieves an estimation error $\mathcal{O}(s^{*2}/nm\varepsilon^2)$. Compared to minimax error rate $\mathcal{O}(ds^*/nm\varepsilon^2)$ under LDP, our rate improves by a factor of s^*/d which can be significant for high

Table 1. Comparison of error rate between non-private, ULDP, and LDP results. Results assume the true parameter lies within ℓ_∞ unit ball. Here, we consider sparse regression with beta-min condition, which improves a $\log d$ over the usual case.

	Non-private (nm samples)	ε-ULDP (n users m samples)	ε-LDP (nm samples)
discrete distribution[1]	$\sqrt{\frac{k}{nm}}$	$\sqrt{\frac{k^2}{nm\varepsilon^2}}$	$\sqrt{\frac{k^2}{nm\varepsilon^2}}$
mean estimation[2]	$\frac{d}{nm}$	$\frac{d^2}{nm\varepsilon^2}$	$\frac{d^2}{nm\varepsilon^2}$
sparse regression[3]	$\frac{s^*}{nm}$	$\frac{\mathbf{s^{*2}}}{\mathbf{nm\varepsilon^2}}$ **(ours)**	$\frac{ds^{*2}}{nm\varepsilon^2}$

dimensional data. Furthermore, we demonstrate how the latter protocol straightforwardly extendeds to other sparse estimation problems (Theorem 3.7).

We summarize our contributions as follows.

- We formalize, for the first time, the advantage of ULDP over LDP by considering the sparse assumption. Our findings reveal that the rates of sparse problems, such as sparse linear regression and sparse mean estimation, do not scale linearly in d under ULDP, which contrasts with previous negative results for LDP.
- We provide a general framework for ULDP sparse estimation. Moreover, focusing on linear regression, we devise tailored methods that achieve tight upper bounds. The precise estimation procedures serve as solutions to low-dimensional ULDP linear regression, which are of independent interest.
- We conduct experiments on both synthetic and real datasets, with convincing results demonstrating the superiority of our methods.

The article is structured as follows: In Section 2, we discuss related literature, preliminary knowledge, and minimax results of ULDP sparse linear regression. In Section 3, we present our solutions. In Section 4, we provide experiment results. All technical proofs, detailed algorithms, and additional experiment results are included in the appendix.

2. ULDP Sparse Linear Regression

2.1. Preliminaries

We introduce necessary notations. For any vector x, let x^i denote the i-th element of x. Let $x^{\{i_1,\cdots,i_j\}}$ be a slicing vector of x, whose j-th elements is x^{i_j}. Let $\|x\|_p$ be the ℓ_p norm of x for $0 \le p \le \infty$. We will evaluate the estimation error by the squared loss, i.e. $\|\widehat{\beta} - \beta^*\|_2^2$. For matrix A, let $\lambda_i(A)$ denote the i-th largest singular value of A. Throughout this paper, we use the notation $a_n \lesssim b_n$ and $a_n \gtrsim b_n$ to denote that there exist positive constant c and c' such that $a_n \le cb_n$ and $a_n \ge c'b_n$, for all $n \in \mathbb{N}$. We use $a = \mathcal{O}(b)$ if $a \lesssim b$. We denote $a_n \asymp b_n$ if $a_n \lesssim b_n$ and $b_n \lesssim a_n$. Let $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. Besides, for any set $A \subset \mathbb{R}^d$, the diameter of A is defined by $\mathrm{diam}(A) := \sup_{x,x' \in A} \|x - x'\|_2$.

Suppose we have n users. The i-th user has m i.i.d. samples $(X_i, y_i) = \{(X_{i,j}, y_{i,j}), j = 1, \cdots, m\}$ from distribution P on domain $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$. We consider the classical sparse linear regression. Let each $X_{i,j}$ be i.i.d. sub-Gaussian. Moreover, $\Sigma = \mathbb{E}[XX^\top]$ denote the covariance matrix of the marginal distribution. Assume $C_X^{-1} \le \lambda_d(\Sigma) \le \lambda_1(\Sigma) \le C_X$ for some constant $C_X > 1$. For mean zero sub-Gaussian random variable σ, conditional distribution $\mathrm{P}_{Y|X}$ and its coefficients β^* are described by

$$y = X\beta^* + \sigma, \quad \beta^* \in \Omega^d_{s^*,a} = \Big\{ \|\beta^*\|_0 \le s^*, \quad \|\beta^*\|_\infty \le 1, \max_{\beta^{*j}>0} |\beta^{*j}| \ge a \Big\}, \tag{1}$$

which has s^*-sparsity and non-zero entries bounded away from 0. Without loss of generality, we assume the first s^* elements of β^* are non-zero.

We adopt the following setting for privacy constraints. Any estimation of β^* is considered as a random variable, while its construction process with respect to the data is *user-level locally differentially private* (ULDP). We consider the sequential interactive case where the private observation U_i is decided only by its local samples (X_i, y_i) and previous observations $U_1, \cdots, U_{i-1}$. The rigorous definition of (pure) ULDP is as follows.

Definition 2.1 (User-level local differential privacy). *Given data $\{(X_i, y_i)\}_{i=1}^n$, each (X_i, y_i) is mapped to privatized information U_i which is a random variable on $\mathcal{U}$. Let $\sigma(\mathcal{U})$ be the σ-field on $\mathcal{U}$. U_i is drawn conditional on (X_i, y_i) via the distribution $\mathrm{R}\big(U_i \mid X_i = x, Y_i = y, U_{1:(i-1)} = u_{1:(i-1)}\big)$. Then the mechanism R provides ε-user-level local differential privacy (ε-ULDP) if*

$$\frac{\mathrm{R}\left(U_i \in U \mid X_i = x, Y_i = y, U_{1:(i-1)} = u_{1:(i-1)}\right)}{\mathrm{R}\left(U_i \in U \mid X_i = x', Y_i = y', U_{1:(i-1)} = u_{1:(i-1)}\right)} \le e^\varepsilon$$

for all $1 \le i \le n$, $U \in \sigma(\mathcal{U})$, $x, x' \in \mathcal{X}^m$, and $y, y' \in \mathcal{Y}^m$.

ULDP reduces to the conventional item-level LDP for $m = 1$. Besides being more practically reasonable (Cummings et al., 2022), ULDP is also a more stringent definition than item-level LDP. To achieve ε-ULDP by trivially using group privacy, each item must use a significantly smaller

[1] Kairouz et al. (2016); Acharya et al. (2023)
[2] Duchi et al. (2018); Bassily & Sun (2023)
[3] Ndaoud (2019); Zhu et al. (2023)

budget ε/m. Conversely, on the curator side, inference of any single item is no easier than inference of the whole user, which means each item is as safe as ε-LDP against the curator. As a compromise, each item should expose information to its group mates. The requirement is typically acceptable, such as when each user has multiple records on a personal cellphone, or when data sources can be clustered into small groups where secrete information is safely shared.

2.2. Related Work

Extensive studies have been conducted focusing on the central DP setting for linear regression model in low dimensions (Wang, 2018; Avella-Medina et al., 2023; Alabi et al., 2022; Arora et al., 2022; Amin et al., 2023) and high dimensions (Kifer et al., 2012; Talwar et al., 2015; Kumar & Deisenroth, 2019; Zhang & Zhang, 2021; Cai et al., 2021; Hu et al., 2022; Khanna et al., 2023a;b; Raff et al., 2023). Despite variations in settings and assumptions, state-of-the-art results (Liu et al., 2022; Varshney et al., 2022; Cai et al., 2023) indicate a general error rate of $\mathcal{O}(s^* \log d/(n\varepsilon^2))$ for squared loss, where the dependency on the dimension of feature space is $\log d$. Thus, d can be exponentially large in $n\varepsilon^2$ to ensure consistent estimation.

This is not the case in local setting, of which there is still a lack of understanding compared to the central one. Several works addressed the problem focusing on the optimization error (Smith et al., 2017; Zheng et al., 2017). Both works assumed $\mathrm{diam}(\mathcal{X}) \leq 1$ and do not generalize to many practical settings, such as when all features are i.i.d. and therefore $\mathrm{diam}(\mathcal{X}) = \mathcal{O}(\sqrt{d})$. As for statistical estimation, Duchi et al. (2018) showed the matching upper and lower bounds for low dimensional, non-interactive linear regression are $\mathcal{O}(d/(n\varepsilon^2))$. Wang & Xu (2019) first provided the lower bound $\mathcal{O}(d/(n\varepsilon^2))$ for LDP linear regression with 1-sparsity, which is then generalized to s-sparsity by Zhu et al. (2023). In summary, these prohibitive results indicate that there exists no meaningful approach when $d \asymp n\varepsilon^2$, which is often the case in practice.

Our approach utilize a selection-estimation strategy, which is shown to be advantages under many situation. Under non-private setting, Wang et al. (2011); Liang et al. (2023) select candidate variables by aggregating Lasso fitted on random subsamples, which is also adopted with privacy (Kifer et al., 2012). More recently, the strategy has been used for communication-constrained learning (Duchi & Rogers, 2019; Barik & Honorio, 2020; Acharya et al., 2021). Note that our method is also communication efficient, as each user sends only 1 bit of information. Acharya et al. (2021) tackled LDP sparse discrete distribution estimation by selecting the support variables. However, this is only feasible for such specific problems where users can provide useful information about which variables are potentially non-zero given only one sample. Their result does not generalize to other problems.

During the preparation of the camera-ready version of this paper, Kent et al. (2024) appeared online and analyzed sparse mean estimation under ULDP. We share some results with their conclusions, including the negative results for $m \leq s^* \log d$, the established rates, and a support estimation type estimator. However, we primarily consider the case where $s^*, \log d \lesssim n\varepsilon^2, m \lesssim d$, whereas their analysis is more comprehensive, considering other regions and, more importantly, identifying the phase transition.

2.3. Minimax Lower Bound

We introduce the related minimax results of locally private sparse linear regression. For any loss function ℓ (squared loss in our case), the minimax convergence rate is

$$\inf_{\beta} \sup_{\mathrm{P} \in \mathcal{H}} \mathbb{E}_{\mathrm{P}} \left[\ell(\beta^*, \beta(X, y))\right]$$

where $\mathcal{H}$ is the hypothesis distribution class and β is any estimator of β^*. The minimax lower bound for sparse linear regression under LDP is well explored in Wang & Xu (2019) and Zhu et al. (2023).

Proposition 2.2 (**LDP lower bound**). *Let $\mathcal{H}$ be distribution class satisfying* (1) *for $0 \leq a \leq 1$. Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated from* (1) *with $n = n'm'$ and $m = 1$. For $0 < \varepsilon \leq 1$, let β_ε be any ε-LDP estimator of β^*. Then we have*

$$\inf_{\beta_\varepsilon} \sup_{\mathrm{P} \in \mathcal{H}} \mathbb{E}_{\mathrm{P}} \left[\|\beta^* - \beta_\varepsilon\|_2^2\right] \gtrsim \frac{ds^*}{n'm'\varepsilon^2}.$$

The above result yields that for $d \gtrsim nm$, any attempt to LDP sparse linear regression is effortless, as the estimation error does not even converge. In fact, similar negative result also holds when m is small yet larger than 1.

Proposition 2.3 (**Necessity of sufficiently large m**). *Suppose $s^{*2} \leq n\varepsilon^2 \lesssim \sqrt{d}$ and $m \leq s^* \log d$. Let $\mathcal{H}$ be distribution class satisfying* (1) *with some constant $a \in [0, 1]$. Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated from* (1)*. For $0 < \varepsilon \leq 1$, let β_ε be any ε-LDP estimator of β^*. Then we have*

$$\inf_{\beta_\varepsilon} \sup_{\mathrm{P} \in \mathcal{H}} \mathbb{E}_{\mathrm{P}} \left[\|\beta^* - \beta_\varepsilon\|_2^2\right] \gtrsim \frac{1}{s^*}.$$

Proposition 2.3 shows that for $m \leq s^* \log d$, the error does dot converge to zero as n grows. However, this is not the case for user-level LDP if m is sufficiently large. The rigorous counterargument is by establishing upper bound in Theorem 3.6, which is $\mathcal{O}(s^{*2}/nm\varepsilon^2)$. We explain why the bound fails to generalize. Its proof involves construction of a function class P_Z and a distribution of Z, such that the mutual information between Z and private views $U_1, \cdots, U_n$

is bounded from above and below. The former does not hold any more given m samples, since the mutual information becomes larger exponentially in m. By carefully bounding the quantity, we establish the following lower bound for ULDP.

Theorem 2.4 (ULDP lower bound). *Suppose $n\varepsilon^2 \geq s^{*2}$, $m \leq d$, and $n\varepsilon^2 \leq d$. Let $\mathcal{H}$ be distribution class satisfying (1) with $a = \sqrt{\frac{s^*}{m}}$. Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated from (1). For $0 < \varepsilon \leq 1$, let β_ε be any ε-ULDP estimator of β^*. Then we have*

$$\inf_{\beta_\varepsilon} \sup_{\mathrm{P} \in \mathcal{H}} \mathbb{E}_{\mathrm{P}} \left[\|\beta^* - \beta_\varepsilon\|_2^2 \right] \gtrsim \frac{s^{*2}}{nm\varepsilon^2}.$$

The result shows that any ULDP estimator admits an error scaling at least with $1/nm\varepsilon^2$. Thus, a possible improvement for ULDP over LDP lies in replacing d with s^*.

3. An Algorithm

We begin by outlining our approach to solving the ULDP sparse linear regression problem. A key observation is that with m samples, each user can obtain a rough estimation of parameter with its local samples. The central challenge then lies in how to aggregate these rough estimations privately. As depicted in Figure 1, our proposed solution operates in two stages. In the initial stage, users within the first group independently identify the non-zero elements of β^* from their local data and transmit privatized information accordingly. By aggregating this information, we determine the s most frequent elements, which serve as the candidate variables for our estimation process. On the narrowed parameter space, we estimate the parameter using remaining users. Subsequently, we present the candidate variable selection, final estimation, and extension to general sparse estimations in Section 3.1, 3.2, and 3.3, respectively.

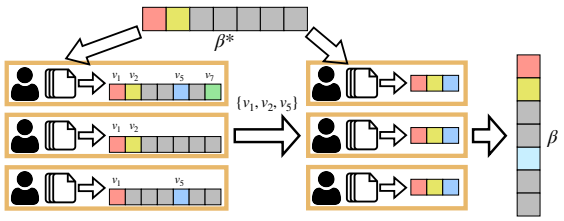


Figure 1. Illustration of the proposed sparse estimation framework.

3.1. Candidate Variable Selection

In this section, we elucidate the steps for candidate variable selection. First, each user prepares a piece of information $v_i \in [d]$, indicating the variable is selected by user i and probably belong to the true variable set. Then, a curator privately aggregates the information v_is and outputs the candidate variables.

To formalize v_i, each user i adopt a local selector $\mathcal{S}_i : (\mathcal{X} \times \mathcal{Y})^m \to [d]$. For each i, $\mathcal{S}_i$ can be any plug-in method and is chosen differently based on the constraints of sample size, computational power, and prior information, as long as it produces a good selection results described as follows.

Definition 3.1 (α-Good selector). *Consider user i and its i.i.d. samples $(X_i, y_i) \in (\mathcal{X} \times \mathcal{Y})^m$ from P. For $0 < \alpha < 1$, an α-good selector is an algorithm $\mathcal{S}$ such that for all $v \in \{1, \cdots, s^*\}$, there holds*

$$\Pr\left(v = \mathcal{S}\left(X_i, y_i\right)\right) \geq \frac{\alpha}{s^*}. \tag{2}$$

Here, the probability is taken w.r.t. randomness of both (X_i, y_i) and $\mathcal{S}$.

(2) requires a lower bound on probability for the true variables to be selected. To induce such selectors, we consider first conducting a local variable selection using (X_i, y_i) and uniformly sampling a v_i. The following proposition demonstrates that obtaining such a selector is feasible given mild assumptions on distribution P, leveraging well-developed variable selection methods.

Proposition 3.2 (Existence of good selectors). *Under model (1), if either of the following conditions holds, there exists a α-good selector with a constant α: (i) $\max_{i \neq j} |\Sigma_{ij}| \leq 3/s^*$, $a \gtrsim \sqrt{1/m}$, and $m \gtrsim s^{*2} \log d$; (ii) $1 \geq a \gtrsim \sqrt{s^*/m} \vee \sqrt{\log m \log d/m}$.*

See Appendix B.1 for examples of precise algorithms and detailed proofs. (*i*) and (*ii*) are examples of sufficient conditions that are relatively easy to satisfy. They require mild correlations among covariates, a strong signal (minimum absolute value of β^*), and an adequate number of local samples. Similar conditions are standard in high-dimensional statistics (Fan & Li, 2001; Zhao & Yu, 2006). Though the lower bound of a is considerable and will leads to a improved minimax rate in the non-private case (Ndaoud, 2019), it is not the key for ULDP sparse linear regression to be advantages over its LDP counterpart. This is because the function classes constructed in Wang & Xu (2019); Zhu et al. (2023) for lower bound proof are all covered by the assumptions. Additionally, the sample size requirement remains polynomial in s^* and $\log d$, which is theoretically reasonable.

Given the local information v_i, we conduct a private voting to identify the frequently appeared variables $\{\widehat{v}^1, \cdots, \widehat{v}^s\}$. Suppose we use the first $n/2$ users for identification, although the proportion is arbitrary and can be any constant. Considering the large size d of the variable universe compared to number of available users, this task is closely related to the problem of heavy hitter detection (Bassily et al., 2020; Acharya et al., 2021). We solve the identification problem in standard manner (Bassily et al., 2020), while any tailored approach is adoptable. Specifically, we encode

the d variables into a binary string using $\lceil \log d \rceil$ bits. Next, we traverse a binary prefix tree from level 1 to $\lceil \log d \rceil$ and eliminate nodes that cannot serve as prefixes of heavy hitters, namely those with frequencies lower than a certain threshold ρ. The key advantage of this method is its ability to identify frequent elements with frequencies above $\sqrt{n \log d \log n/\varepsilon^2}$, which overcomes the polynomial dependency on d in LDP discrete density estimation (Kairouz et al., 2016; Duchi et al., 2018). The detailed procedure (`HeavyHitter`) is provided in Appendix B.2.

In the first part of Algorithm 1, we summarize the pipline for candidate variable selection. The following proposition demonstrates its effectiveness by establishing that, provided the existence of local good selectors, the curator can select a set of variables of size $s \asymp s^*$ containing the true variables. This property, known as perfect selection or consistent selection (Zhao & Yu, 2006; Belloni & Chernozhukov, 2013), plays a crucial role in the theoretical properties of subsequent operations.

Proposition 3.3. *Let $\{\widehat{v}_1, \cdots, \widehat{v}_s\}$ be the selected variables in Algorithm 1. Suppose that all $\mathcal{S}_i$ are α-good selectors with $\alpha \gtrsim s^* \sqrt{\log n \log d/n\varepsilon^2}$. If we take $\alpha/8s^* \leq \rho \leq \alpha/4s^*$, then with probability $1 - 1/n^2$, we have (i) $\{1, \cdots, s^*\} \subseteq \{\widehat{v}_1, \cdots, \widehat{v}_s\}$, (ii) $s \leq 32s^*/\alpha$.*

Note that our scheme samples only one locally selected variable and disregards the others. This select-one-and-aggregate approach has been demonstrated to be as effective as if each user had only one variable (Zhu et al., 2020; Cohen et al., 2023). To fully utilize the information of the selected variables, we can leverage the set-value heavy hitters (Qin et al., 2016; Zhu et al., 2020; Wang et al., 2023). However, this only results in an improvement of $O(\sqrt{s^*})$ in the threshold, which is not our primary focus.

3.2. Coefficient Estimation

Given the selected variables $\{\widehat{v}_1, \cdots, \widehat{v}_s\}$, the problem is reduced to low dimensional linear regression. Efficient algorithms and fundamental limits have been established (Duchi et al., 2018; Wang & Xu, 2019) for item-level LDP. Leveraging these algorithms, one can ignore all but one sample from each user and obtain an error bound depending polynomially on s^* instead of d. However, we would like to explore the benefits brought by having multiple samples per user, as is addressed in the advanced research of ULDP.

We introduce necessary notations to define the learning problem on the selected subspace. Given Proposition 3.3, we assume the selected variables contain the true ones in the following analysis. Without loss of generality, let $(\widehat{v}_1, \cdots, \widehat{v}_s) = (1, \cdots, s)$. We put a hat over the quantities on the selected space. Let $\widehat{\mathrm{P}}$ be the marginal distribution on the selected space $\widehat{\mathcal{X}} \times \mathcal{Y} = \mathbb{R}^{s+1}$, where $\widehat{\mathrm{P}}_{\widehat{X}} = \mathrm{P}_{X^{1:s}}$. Define the data on selected space as $\widehat{X}_{i,j} = X_{i,j}^{1:s}$ and $\widehat{X}_i = \{X_{i,j}^{1:s}\}_{j=1}^m$. The underlying coefficients becomes $\widehat{\beta}^* = \beta^{*1:s}$.

3.2.1. A Multi-round Protocol via SCO

At first glance, we can directly find $\beta \in \mathbb{R}^s$ through the following ULDP stochastic convex optimization problem on the selected space

$$\underset{\|\widehat{\beta}\|_\infty \leq 1}{\arg\min} \left(F(\widehat{\beta}) = \int_{\widehat{\mathcal{X}} \times \mathcal{Y}} \left(x^\top \widehat{\beta} - y \right)^2 d\widehat{\mathrm{P}}(x, y) \right). \quad (3)$$

Recent study (Bassily & Sun, 2023) provided methodology and established theory with respect to smooth loss functions. We borrow their algorithm, presented in Appendix C.1, which is a private variant of accelerated mini-batch gradient descent. It utilize the fact that the gradient of a local batch concentrates with rate $\sqrt{1/m}$ to reduce the magnitude of noise added to the gradients. While the methodology remains the same, we improve the theoretical analysis in Bassily & Sun (2023) to accommodate squared loss, which possesses strong convexity and leads to faster convergence.

Theorem 3.4 (Informal). *Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated as in (1). Suppose $\{\mathcal{S}_i\}_{i=1}^{n/2}$ are α-good selectors with $\alpha \gtrsim s^* \sqrt{\log n \log d/n\varepsilon^2}$. Then with correct parameter choice, solving (3) leads to an estimation β such that*

$$\mathbb{E}\left[\|\beta^* - \beta\|_2^2\right] \lesssim \frac{s^{*9} \log^6 n}{nm\varepsilon^2\alpha^9} + \frac{s^{*4} \log n}{nm\alpha^4}.$$

The result stated in Theorem 3.4 holds *in expectation*, unlike the other conclusions which hold *with high probability*. This distinction arises from the formulation of the technical lemma we borrowed. Upon initial inspection, we notice that both parts of the theorem involve α, indicating a degradation associating to variable selection performance. However, according to Proposition 3.2, α is merely a constant given a sufficiently large m. The higher-order term of s^* encompasses various overheads, including the private mean estimation error and the Lipschitz constant of the squared loss over the $\|\cdot\|_\infty$ ball.

3.2.2. A Two Round Protocol

The multi-round protocol is disadvantageous from two perspectives. Firstly, as a gradient-based method, it necessitates $\mathcal{O}(\sqrt{nm\varepsilon^2})$ rounds of communication, which can be prohibitively slow in practice due to network latency (Smith et al., 2017; Zheng et al., 2017). Secondly, compared to Theorem 2.4, the upper bound provided in Theorem 3.4 is far from tight concerning s^*. We question whether these drawbacks can be mitigated for the specific problem of linear regression. In this section, we provide an affirmative answer. Our main inspiration stems from the following observation.

Proposition 3.5. *There exists estimators on selected variables $\widehat{\beta}_{n/2+1}, \cdots, \widehat{\beta}_n$, such that for all $\widehat{\beta}_i \in \mathbb{R}^s$, we have $\mathbb{E}_{\mathrm{P}}\left[\widehat{\beta}_i\right] = \widehat{\beta}^*$ and $\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \lesssim \sqrt{s \log n/m}$ with probability $1 - 1/n^2$. Moreover, if either condition in Proposition 3.2 holds, the bound improves to $\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \lesssim \sqrt{s^* \log n/m}$.*

Since the mean of $\widehat{\beta}_i$ is $\widehat{\beta}^*$, an ideal estimator would be the mean of $\widehat{\beta}_i$s. Moreover, Proposition 3.5 indicates that $\widehat{\beta}_i$ concentrates as m increases, suggesting that we can confine $\widehat{\beta}_i$ to a restricted area to enhance estimation accuracy. We propose a two-stage estimation similar to Girgis et al. (2022). First, leveraging user indices $n/2 + 1 \leq i \leq 3n/4$, we designate a histogram bin on $\mathbb{R}^s$, wherein almost all the $\widehat{\beta}_i$ values will fall. Then, the last group of users project their $\widehat{\beta}_i$ onto the bin and add a Laplace noise. Given the reduced sensitivity of the projected coefficients, the noise magnitude significantly diminishes. We provide detailed methodology (`ULDPMean`) in Appendix C and summarize the pipline in Algorithm 1.

Algorithm 1 Two-round ULDP sparse estimation.

Input: Local data sets $\{(X_i, y_i)\}_{i=1}^n$, selectors $\{\mathcal{S}_i\}_{i=1}^{n/2}$, privacy budget ε, threshold ρ, concentration radius τ.
Initialization: $\beta \in \mathbb{R}^d$ be a zero vector.
`# candidate variable selection`
`# on local machine`
for i in $1, \cdots, n/2$ **do**
  $v_i = \mathcal{S}_i(X_i, y_i)$.
end for
`# ⌈log d⌉ round communication`
$\{\widehat{v}_1, \cdots, \widehat{v}_s\}$ = `HeavyHitter`$(\{v_i\}_{i=1}^{n/2}, \varepsilon, \rho)$.
`# coefficient estimation`
`# on local machine`
for i in $n/2 + 1, \cdots, n$ **do**
  Fit $\widehat{\beta}_i$ according to $(\widehat{v}_1, \cdots, \widehat{v}_s)$.
end for
`# 2 round communication`
$\widehat{\beta}$ = `ULDPMean`$(\{\widehat{\beta}_i\}_{i=n/2+1}^{3n/4}, \{\widehat{\beta}_i\}_{i=3n/4+1}^{n}, \tau, \varepsilon)$.
$\beta^{\widehat{v}_1:\widehat{v}_s} = \widehat{\beta}$.
Output: β.

The entire protocol requires a reasonable $\log d + 2$ rounds of communication, with each user sending 1 bit of information. The $\log d$ communication rounds are necessary for `HeavyHitter`, which can be substituted by any other customized identification method for improved efficiency. In the coefficient estimation stage, our method takes two round communication, which is quite efficient. Fully utilizing multiple samples necessitates sequential interactivity (Acharya et al., 2023; Bassily & Sun, 2023).

We now present the main result, which is the error upper bound of the estimator summarized in Algorithm 1.

Theorem 3.6. *Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated as in (1). Suppose $\{\mathcal{S}_i\}_{i=1}^{n/2}$ are α-good selectors with $\alpha \gtrsim s^* \sqrt{\log n \log d/n\varepsilon^2}$. Suppose we let $\alpha/8s^* \leq \rho \leq \alpha/4s^*$ and $\tau \asymp \sqrt{\log^2 n/m}$. Let β be the output of Algorithm 1. Then we have (i) Algorithm 1 is ε-ULDP. (ii) there holds*

$$\|\beta^* - \beta\|_2^2 \lesssim \frac{s^* \log n}{nm\alpha} + \frac{s^{*2} \log^3 n}{nm\varepsilon^2\alpha^2} \tag{4}$$

with probability at least $1 - 4/n^2$. Moreover, if either condition in Proposition 3.2 holds, the bound improves to

$$\|\beta^* - \beta\|_2^2 \lesssim \frac{s^* \log n}{nm} + \frac{s^{*2} \log^3 n}{nm\varepsilon^2\alpha}. \tag{5}$$

The upper bound in (4) consists of two parts. Both terms include additional αs and $\log n$s, which are inevitable due to selection degradation and the overhead of utilizing multiple local samples. The $\log n$s are due to the union bound arguments, while α is merely constants by Proposition 3.2. Ignoring α and $\log n$, the first part recovers the rate of non-private linear regression on $\mathcal{O}(s^*)$ dimensional space. The second part corresponds to privacy. When $\varepsilon \gtrsim \sqrt{s^*}$, this part is negligible. Algorithm 1 achieves the same error as if its non-private. It worth noting that in most cases (see e.g. Table 1), locally private algorithm matches its non-private counterpart when $\varepsilon \gtrsim \sqrt{s^*}$. The improvement of (5) over (4) is based on the existence of sparse oracles that achieve error s^*/m locally, instead of s/m.

We observe that, unlike common high-dimensional results (Wang & Xu, 2019; Cai et al., 2023), our bound does not involve a $\log d$ term. This phenomenon is also noted in Ndaoud (2019), where the $\log d$ disappears if we leverage the beta-min condition in Proposition 3.2. We will observe in the experiments that if m is large enough, our method is more robust to changes in d. However, this is not to say that we can deal with arbitrarily large d. The logarithmical relationship is still contained in α, which poses a requirement of $m \gtrsim \log d$ as in Proposition 3.2. Moreover, omitting the log factors, the privacy error is decided by the total number of samples mn for sufficiently large m and n. Thus, we can achieve the same estimation error with less number of users if there are more local samples per user, while retaining the same level of privacy for each user since ε is fixed. On contrary, if there is mn users with one sample each, the error is inevitably $\mathcal{O}(ds^*/nm\varepsilon^2)$ (Proposition 2.2). This comparison illustrates the advantage of having both sufficient users and local samples compared to having abundant users and only one local sample. Note that this distinction holds only between sequential-interactive ULDP and LDP. It is unclear whether the lower bound holds under non-interactive ULDP, since most ULDP methods require sequential interactivity (Acharya et al., 2023; Bassily & Sun, 2023).

3.3. Extension to Sparse Estimation

In this section, we show our framework can be applied to various sparse problems through reduction to non-private learners. We consider estimation of β^* from data $\{X_i\}_{i=1}^n \in \mathcal{X}^{mn}$, which is generated from distribution P_{β^*} parameterized by β^*. β^* is assumed to be in $\Omega_{s,a}^d$. The assumptions include linear regression as a special case. It's important to note that Algorithm 1 depends on the particular problem form via two steps: (*i*) the selector $\mathcal{S}_i$ and (*ii*) the estimator $\widehat{\beta}_i$. Both components depend on a non-private estimator of β^*. The following theorem demonstrates that, given a qualified estimator, our framework achieves fast convergence rates for the general problem of sparse estimation.

Theorem 3.7 (Informal). *Let data $\{X_i\}_{i=1}^n$ be generated by P_{β^*} for $\beta^* \in \Omega_{s,a}^d$. Suppose we have non-private estimators: (i) estimator $\tilde{\beta}_i$ with $\|\tilde{\beta}_i - \beta^*\|_2 \leq \nu_1$ for all $1 \leq i \leq n/2$ and (ii) estimator $\widehat{\beta}_i$ on selected variables with $\mathbb{E}\left[\widehat{\beta}_i\right] = \widehat{\beta}^*$ and $\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \leq \nu_2$ for all $n/2+1 \leq i \leq n$. Then, for any $a \gtrsim \nu_1$, there exists an ε-ULDP algorithm whose output β has*

$$\|\beta^* - \beta\|_2^2 \lesssim \frac{\nu_2^2}{n} + \frac{\nu_2^2 s^* \log^2 n}{n\varepsilon^2\alpha} \tag{6}$$

with probability at least $1 - 3/n^2$. Moreover, for ℓ_1 norm, there holds

$$\|\beta^* - \beta\|_1 \lesssim \sqrt{\frac{\nu_2^2 s^*}{n\alpha}} + \sqrt{\frac{\nu_2^2 s^{*2} \log^2 n}{n\varepsilon^2\alpha^2}} \tag{7}$$

with probability at least $1 - 3/n^2$.

We also present a result for the ℓ_1 norm. Comparing (7) to (6), the difference arises from the $\sqrt{s}$ discrepancy between the ℓ_1 and ℓ_2 norms, given that we only have s non-zero elements in our sparse estimation problem. We discuss the implications of Theorem 3.7. Consider the sparse mean estimation (Duchi et al., 2018; Zhou et al., 2022), where non-private estimator achieves $\nu_2 = \mathcal{O}(\sqrt{s^* \log n/m})$ (Johnstone, 1994) under mild conditions. Then the bound (6) becomes identical to (5), which eliminates the linear dependency of d in LDP (Duchi et al., 2018). For sparse discrete distribution estimation, Acharya et al. (2021) removed the linear dependency of d. With $\nu_2 = \mathcal{O}(\sqrt{s^* \log n/m})$, our bound (7) is $\sqrt{s^*}$ larger than theirs in ℓ_1 sense.

It worth mentioning that when d is small, our upper bound matches the lower bound for $m = 1$. In this scenario, selector provides no useful information and is equivalent to a random selection, i.e. $\alpha \leq \mathrm{P}\left(v = \mathcal{S}(X_i, y_i)\right) \cdot s^* = s^*/d$. If $\alpha = s^*/d \gtrsim s^* \sqrt{\log n \log d / n\varepsilon^2}$, then (5) becomes

$$\frac{s^* \log n}{nm} + \frac{ds^* \log^3 n}{nm\varepsilon^2}.$$

Up to logarithmic factors, the second term matches the lower bound established in Zhu et al. (2023) for sparse linear regression and Duchi et al. (2018) for sparse mean estimation.

4. Experiment Results

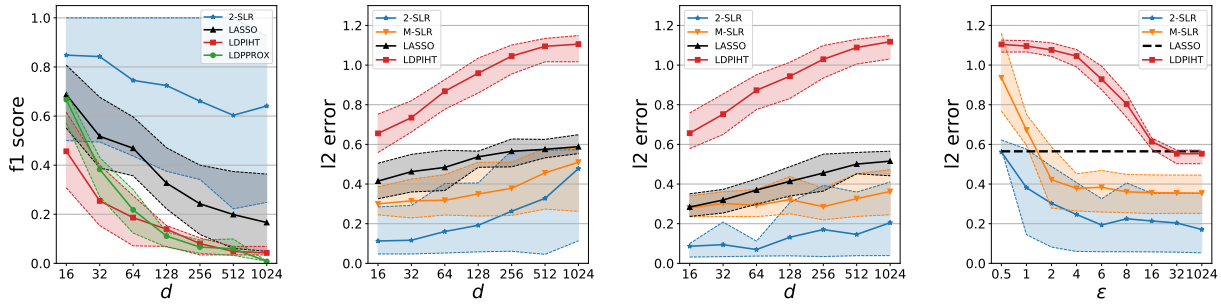
We conduct experiments on both synthetic and real datasets to show the superiority of proposed methods and to validate our theoretical findings. The tested methods include: ***(i)* 2-SLR**: The proposed two-round ULDP sparse linear regression method outlined in Algorithm 1; ***(ii)* M-SLR**: The proposed multi-round version in Algorithm 6. The competing methods are: ***(iii)* LDPPROX**: The non-interactive LDP proxy estimator in Zhu et al. (2023); ***(iv)* LDPIHT**: The LDP iterative hard thresholding in Wang & Xu (2019); Zhu et al. (2023). Both comparison methods receive nm samples with budget ε each. Additionally, we report performance of non-privately fitting ***(v)* Lasso** using m samples, representing an alternative for each user to rely solely on their local information. Implementation details are provided in Appendix E. For each model, we report the best result over its parameter grids, with the best result determined based on the average of at least 30 replications. The size of the parameter grids is selected based on running time to ensure that each method incurs an equal amount of computation. All experiments are conducted on a machine with 72-core Intel Xeon 2.60GHz and 128GB of main memory. The code is publicly available at GitHub[4].

4.1. Simulation

We conducted experiments on synthetic data to validate the theoretical findings. Two sets of parallel experiments are conducted for independent and correlated marginal distributions, respectively, while results of the latter are presented in Appendix E. We draw each $X_{i,j}^k$ and $\sigma_{i,j}$ independently from standard Gaussian distribution. For β^*, we randomly select $s^* = 8$ coordinates to be 0.2 and let others be zero. Typically, we set $n = 400$, $m = 100$, $d = 256$, and $\varepsilon = 4$, while varying one of them to observe how the evaluated metric varies. We use squared error to evaluate the estimated coefficients and F1 score to evaluate variable selection.

We conduct experiments w.r.t. d. We first analyze the variable selection performance. For $d \in \{16, 32, \cdots, 1024\}$, we compute the averaged F1 scores of the proposed candidate variable selection (represented by 2-SLR) and other methods. As shown in Figure 2(a), the selection performance of 2-SLR is superior to variables induced by other methods. Particularly noteworthy is that 2-SLR achieved higher F1 scores than Lasso. This observation aligns with Wang et al. (2011); Liang et al. (2023), where aggregating

[4] https://github.com/Karlmyh/ULDP-SL



(a) d - F1 score. (b) d - ℓ_2 error with $m = 100$. (c) d - ℓ_2 error with $m = 200$. (d) ε - ℓ_2 error.

Figure 2. Experiments w.r.t. d and ε. We plot the quantiles over 30 repetitions with 95% coverage. We exclude LDPPROX in the last three figures since it is highly unstable and do not fit into our plot scale.

Lasso fitted on random subsamples leads to performance gains in both selection and prediction.

Next, we analyze the estimation performance with respect to d. In Figures 2(b) and 2(c), we plot the curve of ℓ_2 error w.r.t. d. Given a large $m = 200$, the proposed methods are less sensitive to d compared to LDPIHT and Lasso. This observation is compatible with the rate in (5), which is independent of d. Conversely, for smaller $m = 100$, the local selectors can not provide a constant α for exponentially larger d. As a result, the trend of our methods is steeper.

We examine the privacy-utility trade-offs by investigating performances under different ε. In Figure 2(d), the error decreases as ε increases for all private methods. Moreover, the error of 2-SLR is consistently better than Lasso, while error of M-SLR quickly drops below Lasso at medium privacy levels ($\varepsilon \geq 2$). This shows the superiority of our methods compared to fitting Lasso using only local information.

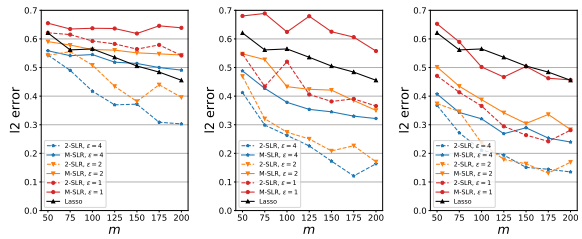
Finally, we analyze the impact of sample sizes. We conducted experiments with varying m (ranging from 50 to 200) under different n, comparing the performance of our methods with Lasso on local samples. The results for varying m are presented in Figure 3. We observe that given a sufficiently large $n = 800$, 2-SLR always outperforms Lasso, and M-SLR performs comparably even for $\varepsilon = 1$. If $n = 400$, only M-SLR with $\varepsilon = 1$ performs worse than Lasso. However, given an insufficient $n = 100$, Lasso performs comparably to 2-SLR with $\varepsilon = 2$. Similarly, in Figures 4(a) and 4(b), the ℓ_2 error decreases as n increases for all ε. The results indicate that our methods outperform Lasso under various (n, m, ε) settings, except for M-SLR with $\varepsilon = 1$. This observation is reasonable and aligns with phenomena commonly observed in ULDP learning, federated learning, or transfer learning, where incorporating information from other data sources may not necessarily improve estimation if the quality of that additional information is low due to factors such as privacy constraints, data heterogeneity, or data compression.

Moreover, we set $nm = 400 \times 100$ and varied the ratio n/m. In Figure 4(c), we observe that, for each ε, the error of 2-SLR remains stable when $n \approx m$, while it slightly increases when either n or m is too small, which is consistent with Theorem 3.6. Furthermore, the performance of M-SLR is more sensitive to n becoming small. This is attributed to its gradient nature, which requires a large number of users.

4.2. Real Data

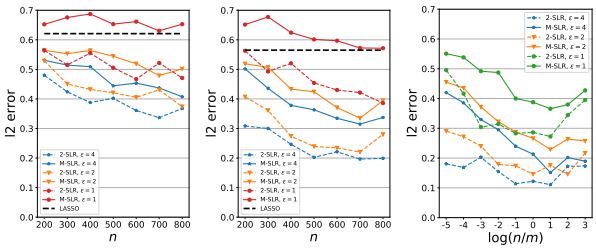
We conduct experiments on six real datasets with various sample sizes and dimensionalities. Among the datasets, `Airline` and `Taxi` are the most suitable for our setting, where each user possesses small local samples with large dimensions. The datasets contain sensitive information and have been used in privacy research (Ma et al., 2024b). The other datasets are manually grouped to fit our framework. See Appendix E.3 for description of datasets.

We first compute the mean squared error over 30 random train-test splits for $\varepsilon = 1$ and $\varepsilon = 4$. To standardize the



(a) $n = 100$ (b) $n = 400$ (c) $n = 800$

Figure 3. Experiments w.r.t. m and ℓ_2 error.



(a) n - ℓ_2 error. (b) m - ℓ_2 error. (c) n/m - ℓ_2 error.

Figure 4. Experiments w.r.t. n and n/m.

Table 2. Real data performances. To ensure significance, we employ the Wilcoxon signed-rank test (Wilcoxon, 1992) with a significance level of 0.05 to determine if a result is significantly better. The best results are **bolded** and those holding significance towards the rest results are marked with $*$.

Budget	Datasets	NP-2-SLR	NP-M-SLR	Lasso	2-SLR	M-SLR	LDPPROX	LDPIHT
$\varepsilon = 1$	Airline	1.01	0.82	1.02	1.02	**0.98***	1.38	1.85
	Loan	0.97	0.88	0.99	0.98	**0.97**	5.27	2.00
	MIP	1.00	0.96	1.65	1.00	**0.98***	2.54	1.87
	Taxi	0.95	0.01	1.04	0.96	**0.01***	1.20	1.02
	Wine	1.19	1.17	**1.14***	1.34	1.37	7.71	2.30
	Yolanda	1.10	1.14	**1.19**	**1.19**	1.22	1.90	2.36
$\varepsilon = 4$	Airline	1.01	0.82	1.02	1.02	**0.88***	1.15	1.02
	Loan	0.97	0.88	0.99	0.98	**0.90***	2.05	1.65
	MIP	1.00	0.96	1.65	1.01	**0.96***	3.30	1.82
	Taxi	0.95	0.01	1.04	0.95	**0.01***	1.16	1.88
	Wine	1.19	1.17	**1.14***	1.19	1.27	5.39	1.74
	Yolanda	1.10	1.14	1.19	**1.11***	1.18	1.79	2.03
Rank sum		-		31	24	**19**	56	50

scale across datasets, we report the MSE ratio relative to non-private fitting with Lasso over all samples. The results are displayed in Table 2. For both high privacy ($\varepsilon = 1$) and medium privacy ($\varepsilon = 4$), the proposed methods significantly outperform competitors in terms of both average performance (rank sum) and the number of best results achieved. It is worth noting that in most cases, Lasso fitted on local datasets outperforms LDP competitors, yielding the effortlessness of LDP sparse regression. Moreover, the running time of the methods is displayed in Appendix E.3. The results show that, if properly paralleled, our methods are quite efficient.

We observe that our methods (2-SLR and M-SLR) can sometimes outperform non-private Lasso on the whole data. This is somewhat expected. As explained in previous literature (Ndaoud, 2019), given strong signal strength ($\min_{\beta^j>0} |\beta^j|$ is large), the optimal error can actually be improved and simply performing Lasso does not achieve this optimality. Moreover, methodological works (Wang et al., 2011; Liang et al., 2023) showed the effectiveness of selecting candidate variables by aggregating Lasso fitted on random subsamples. Intuitively, even with strong signal strength, fitting Lasso does not guarantee the selection of all true variables due to randomness, while aggregating variables selected on random subsamples is more likely to identify true variables. We validated our conjecture by running non-private SLRs ($\varepsilon = 1024$). The results are presented in Table 2. We observe that 2-SLR and M-SLR never outperform their non-private counterparts, while non-private SLRs occasionally outperform Lasso on some datasets.

We also observe that 2-SLR outperforms M-SLR in simulation, while the opposite is true in real data. The phenomenon is attributed to the implicit regularization. In synthetic data, where the data is neatly generated, estimations tend to converge well. However, real data often contains more noise, leading to potentially unstable estimations. In such cases, using zero coefficients as the initial point yields a regularized estimator (Ali et al., 2019), which are biased yet stable.

5. Discussion

In this work, we investigate the ULDP sparse linear regression. By proposing a two-phase solution, we show the theoretical advantage of having multiple samples per user, which is then validated by exhaustive experiments.

It is worth mentioning that we do not explore scenarios where m is small, such as $m \leq s^* \log d$. Our experiments, particularly with the MIP dataset, demonstrate that even with few local samples, satisfactory results can be achieved. However, dealing with small m may require a comprehensive distributional analysis of variable selection, which could be a promising avenue for future research. We also hope to establish a tight minimax lower bound of sparse estimation under ULDP.

Currently, we consider a support estimation based algorithm. As suggested by the reviewers, an interesting topic would be an algorithm that simultaneously learns the sparse coefficients and optimizes the model, potentially with a Lasso-type optimization objective. Directly solving such a problem is ineffective (Bassily & Sun, 2023). Each update step will involve updating $d - s$ redundant parameters, whose information needs to be protected under differential privacy. Thus, excessive random noise is injected. By utilizing support estimation, we circumvent this issue in the second phase, leading to improved final rates. A private analog for algorithms with limited message passing each round is promising, such as least-angle regression (LARS) or coordinate gradient descent.

Impact Statement

We believe that it is difficult to clearly foresee societal consequence of the present work, which has a primary focus on machine learning theory and methodology. We believe this work can serve as a forward step to enclosing the gap between the theoretical study of LDP and practical situations.

Acknowledgement

We would like to thank the reviewers for their help and advice, which led to a significant improvement of the article. We also thank Yifan Gu for providing discussion on variable selection issues. The research is supported by the Special Funds of the National Natural Science Foundation of China (Grant No. 72342010). Yuheng Ma is supported by the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China. This research is also supported by Public Computing Cloud, Renmin University of China.

References

Acharya, J., Canonne, C. L., Sun, Z., and Tyagi, H. Unified lower bounds for interactive high-dimensional estimation under information constraints. *arXiv preprint arXiv:2010.06562*, 2020.

Acharya, J., Kairouz, P., Liu, Y., and Sun, Z. Estimating sparse discrete distributions under privacy and communication constraints. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*. PMLR, 16–19 Mar 2021.

Acharya, J., Liu, Y., and Sun, Z. Discrete distribution estimation under user-level local differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 8561–8585. PMLR, 2023.

Alabi, D., McMillan, A., Sarathy, J., Smith, A., and Vadhan, S. Differentially private simple linear regression. *Proceedings on Privacy Enhancing Technologies*, 2022.

Ali, A., Kolter, J. Z., and Tibshirani, R. J. A continuous-time view of early stopping for least squares regression. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1370–1378. PMLR, 2019.

Amin, K., Joseph, M., Ribero, M., and Vassilvitskii, S. Easy differentially private linear regression. In *The Eleventh International Conference on Learning Representations*, 2023.

Arora, R., Bassily, R., Guzmán, C., Menart, M., and Ullah, E. Differentially private generalized linear models revisited. *Advances in Neural Information Processing Systems*, 35:22505–22517, 2022.

Avella-Medina, M., Bradshaw, C., and Loh, P.-L. Differentially private inference via noisy optimization. *The Annals of Statistics*, 51(5):2067–2092, 2023.

Barik, A. and Honorio, J. Exact support recovery in federated regression with one-shot communication. *arXiv preprint arXiv:2006.12583*, 2020.

Bassily, R. and Sun, Z. User-level private stochastic convex optimization with optimal rates. In *International Conference on Machine Learning*, pp. 1838–1851. PMLR, 2023.

Bassily, R., Nissim, K., Stemmer, U., and Thakurta, A. Practical locally private heavy hitters. *The Journal of Machine Learning Research*, 21(1):535–576, 2020.

Belloni, A. and Chernozhukov, V. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521, 2013.

Bergdoll, R.-D. Mip-2016-regression, 2019. URL https://www.openml.org/search?type=data&status=active&id=41702.

Cai, T. T., Wang, Y., and Zhang, L. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5): 2825–2850, 2021.

Cai, Z., Li, S., Xia, X., and Zhang, L. Private estimation and inference in high-dimensional regression with fdr control. *arXiv preprint arXiv:2310.16260*, 2023.

Cohen, E., Lyu, X., Nelson, J., Sarlos, T., and Stemmer, U. Hot pate: Private aggregation of distributions for diverse task. *arXiv preprint arXiv:2312.02132*, 2023.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4): 547–553, 2009.

Cummings, R., Feldman, V., McMillan, A., and Talwar, K. Mean estimation with user-level privacy under data heterogeneity. *Advances in Neural Information Processing Systems*, 35:29139–29151, 2022.

Dieuleveut, A., Flammarion, N., and Bach, F. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18(1): 3520–3570, 2017.

DrivenData. Loan default prediction - imperial college london, 2021a. URL https://www.kaggle.com/c/loan-default-prediction/data.

DrivenData. Differential privacy temporal map challenge: Sprint 3 (prescreened arena), 2021b. URL https://www.drivendata.org/competitions/77/deid2-sprint-3-prescreened/page/332/.

Duchi, J. and Rogers, R. Lower bounds for locally private estimation via communication complexity. In *Conference on Learning Theory*, pp. 1161–1191. PMLR, 2019.

Duchi, J., Jordan, M., and Wainwright, M. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

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

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Fan, J. and Lv, J. Sure independence screening for ultra-high dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70 (5):849–911, 2008.

Fan, J. and Lv, J. Nonconcave penalized likelihood with np-dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484, 2011.

Fan, J., Xue, L., and Zou, H. Strong oracle optimality of folded concave penalized estimation. *Annals of statistics*, 42(3):819, 2014.

Ghazi, B., Kumar, R., and Manurangsi, P. User-level differentially private learning via correlated sampling. *Advances in Neural Information Processing Systems*, 34: 20172–20184, 2021.

Ghazi, B., Kamath, P., Kumar, R., Manurangsi, P., Meka, R., and Zhang, C. On user-level private convex optimization. In *International Conference on Machine Learning*, pp. 11283–11299. PMLR, 2023.

Girgis, A. M., Data, D., and Diggavi, S. Distributed user-level private mean estimation. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2196–2201. IEEE, 2022.

Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., et al. Analysis of the automl challenge series. *Automated Machine Learning*, 177, 2019.

Hsu, D., Kakade, S. M., and Zhang, T. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17:1, 2012.

Hu, L., Ni, S., Xiao, H., and Wang, D. High dimensional differentially private stochastic optimization with heavy-tailed data. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 227–236, 2022.

Johnstone, I. M. On minimax estimation of a sparse normal mean vector. *The Annals of Statistics*, pp. 271–289, 1994.

Kairouz, P., Oh, S., and Viswanath, P. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems*, 27, 2014.

Kairouz, P., Bonawitz, K., and Ramage, D. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pp. 2436–2444. PMLR, 2016.

Kent, A., Berrett, T. B., and Yu, Y. Rate optimality and phase transition for user-level local differential privacy. *arXiv preprint arXiv:2405.11923*, 2024.

Khanna, A., Lu, F., and Raff, E. The challenge of differentially private screening rules. *arXiv preprint arXiv:2303.10303*, 2023a.

Khanna, A., Lu, F., and Raff, E. Sparse private lasso logistic regression. *arXiv preprint arXiv:2304.12429*, 2023b.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1. JMLR Workshop and Conference Proceedings, 2012.

Kumar, K. and Deisenroth, M. P. Differentially private empirical risk minimization with sparsity-inducing norms. *arXiv preprint arXiv:1905.04873*, 2019.

LeDell, E. Airlines depdelay 10m, 2020. URL https://www.openml.org/search?type=data&status=active&id=42728.

Levy, D., Sun, Z., Amin, K., Kale, S., Kulesza, A., Mohri, M., and Suresh, A. T. Learning with user-level privacy. *Advances in Neural Information Processing Systems*, 34: 12466–12479, 2021.

Liang, J., Wang, C., Zhang, D., Xie, Y., Zeng, Y., Li, T., Zuo, Z., Ren, J., and Zhao, Q. Vsolassobag: a variable-selection oriented lasso bagging algorithm for biomarker discovery in omic-based translational research. *Journal of Genetics and Genomics*, 50(3):151–162, 2023.

Liu, X., Kong, W., and Oh, S. Differential privacy and robust statistics in high dimensions. In *Conference on Learning Theory*, pp. 1167–1246. PMLR, 2022.

Liu, Y., Suresh, A. T., Yu, F. X. X., Kumar, S., and Riley, M. Learning discrete distributions: user vs item-level privacy. *Advances in Neural Information Processing Systems*, 33: 20965–20976, 2020.

Ma, Y. and Yang, H. Optimal locally private nonparametric classification with public data. *Journal of Machine Learning Research*, 2024.

Ma, Y., Jia, K., and Yang, H. Locally private estimation with public features. *arXiv preprint arXiv:2405.13481*, 2024a.

Ma, Y., Zhang, H., Cai, Y., and Yang, H. Decision tree for locally private estimation with public data. *Advances in Neural Information Processing Systems*, 36, 2024b.

Narayanan, S., Mirrokni, V., and Esfandiari, H. Tight and robust private mean estimation with few users. In *International Conference on Machine Learning*, pp. 16383–16412. PMLR, 2022.

Ndaoud, M. Interplay of minimax estimation and minimax support recovery under sparsity. In *Algorithmic Learning Theory*, pp. 647–668. PMLR, 2019.

Papernot, N. and Steinke, T. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2021.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Qin, Z., Yang, Y., Yu, T., Khalil, I., Xiao, X., and Ren, K. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 192–203, 2016.

Raff, E., Khanna, A. A., and Lu, F. Scaling up differentially private lasso regularized logistic regression via faster frank-wolfe iterations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Smith, A., Thakurta, A., and Upadhyay, J. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 58–77. IEEE, 2017.

Talwar, K., Guha Thakurta, A., and Zhang, L. Nearly optimal private lasso. *Advances in Neural Information Processing Systems*, 28, 2015.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Tramèr, F., Kamath, G., and Carlini, N. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022.

Varshney, P., Thakurta, A., and Jain, P. (nearly) optimal private linear regression for sub-gaussian data via adaptive clipping. volume 178 of *Proceedings of Machine Learning Research*, pp. 1126–1166. PMLR, 2022.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, D. and Xu, J. On sparse linear regression in the local differential privacy model. In *International Conference on Machine Learning*, pp. 6628–6637. PMLR, 2019.

Wang, S., Nan, B., Rosset, S., and Zhu, J. Random lasso. *The annals of applied statistics*, 5(1):468, 2011.

Wang, S., Li, Y., Zhong, Y., Chen, K., Wang, X., Zhou, Z., Peng, F., Qian, Y., Du, J., and Yang, W. Locally private set-valued data analyses: Distribution and heavy hitters estimation. *IEEE Transactions on Mobile Computing*, 2023.

Wang, Y. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, pp. 93–103, 2018.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pp. 196–202. Springer, 1992.

Zhang, Z. and Zhang, L. High-dimensional differentially-private em algorithm: Methods and near-optimal statistical guarantees. *arXiv preprint arXiv:2104.00245*, 2021.

Zhao, P. and Yu, B. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7: 2541–2563, 2006.

Zheng, K., Mou, W., and Wang, L. Collect at once, use effectively: Making non-interactive locally private learning possible. In *International Conference on Machine Learning*, pp. 4130–4139. PMLR, 2017.

Zhou, M., Wang, T., Chan, T. H., Fanti, G., and Shi, E. Locally differentially private sparse vector aggregation. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 422–439. IEEE, 2022.

Zhu, L., Ding, M., Aggarwal, V., Xu, J., and Wang, D. Improved analysis of sparse linear regression in local differential privacy model. *arXiv preprint arXiv:2310.07367*, 2023.

Zhu, W., Kairouz, P., McMahan, B., Sun, H., and Li, W. Federated heavy hitters discovery with differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 3837–3847. PMLR, 2020.

In this appendix, we provide the omitted content for minimax lower bound (Appendix A), the algorithm and theoretical results of candidate variable selection (Appendix B), the algorithm and theoretical results of coefficient estimation (Appendix C), an extension from our framework to general problems (Appendix D), and details as well as additional results of experiments (Appendix E).

A. Minimax Lower Bound

We first borrow assumptions and definitions from Acharya et al. (2020). Let $Z = (Z_1, \ldots, Z_d)$ be a random variable over $\mathcal{Z} = \{-1, +1\}^d$ such that $\mathbb{P}[Z_i = 1] = \tau$ for all $i \in [d]$ and the Z_i s are all independent; we denote this distribution by $\mathrm{Rad}(\tau)^{\otimes d}$. For $z \in \mathcal{Z}$, we denote $z^{\oplus i} \in \mathcal{Z}$ as the vector obtained by flipping the sign of the i-th coordinate of z.

Condition A.1. *For every $z \in \mathcal{Z}$ and $i \in [d]$ it holds that $\mathrm{P}_{z^{\oplus i}} \ll \mathrm{P}_z$ (we refer to P_{β_z} simply as P_z), and there exist measurable functions $\phi_{z,i} : \mathbb{R}^d \to \mathbb{R}$ such that*

$$\frac{\mathrm{dP}_{z^{\oplus i}}}{\mathrm{dP}_z} = 1 + \phi_{z,i}.$$

Condition A.2. *There exists some $\alpha^2 \geq 0$ such that, for all $z \in \mathcal{Z}$ and distinct $i, j \in [d], \mathbb{E}_{\mathrm{P}_z}[\phi_{z,i} \cdot \phi_{z,j}] = 0$ and $\mathbb{E}_{\mathrm{P}_z}[\phi_{z,i}^2] \leq \alpha^2$.*

Condition A.3. *For every $z, z' \in \mathcal{Z} = \{-1, +1\}^d$,*

$$\ell_2(\theta_z, \theta_{z'}) = 4\nu \left(\frac{\mathrm{d}_{\mathrm{Ham}}(z, z')}{\tau d} \right)^{1/2}$$

where $\mathrm{d}_{\mathrm{Ham}}(z, z') := \sum_{i=1}^d \mathbf{1}\{z_i \neq z_i'\}$ denotes the Hamming distance, where $\tau = s^*/2d, s^*$ and ν denotes sparsity and error rate respectively.

Proof of Theorem 2.4. First, suppose X^j is uniformly distributed on $\{-1, 1\}$ for $1 \leq j \leq d$. Let

$$\beta_{Z,j}^* = \frac{4\sqrt{2}\nu}{\sqrt{s^*}} \frac{Z_j + 1}{2}$$

for $1 \leq j \leq d$ where Z_js are i.i.d. random variables with

$$\Pr[Z_i = +1] = \frac{s^*}{2d}, \quad \Pr[Z_i = -1] = 1 - \frac{s^*}{2d}.$$

There holds β_Z^* satisfies the conditions that $\|\beta_Z^*\|_\infty \leq 1$ and $\|\beta_Z^*\|_0 \leq s^*$ with probability $1 - s^*/2d$ using Fact 1 in Acharya et al. (2020). Next, for each Z we let:

$$\sigma_Z = \begin{cases} 1 - \langle X, \beta_Z^* \rangle & \text{w.p.} \quad \frac{1 + \langle X, \beta_Z^* \rangle}{2} \\ -1 - \langle X, \beta_Z^* \rangle & \text{w.p.} \quad \frac{1 - \langle X, \beta_Z^* \rangle}{2} \end{cases}$$

Thus, $Y \in \{-1, 1\}$. The above distribution satisfies (1) with probability $1 - s^*/2d$. The distribution P_Z has density function $(1 + Y\langle X, \beta_Z^* \rangle)/2^{d+1}$ for $(X, Y) \in \{+1, -1\}^{d+1}$. Then, for the i-th user who has the data sample (X_i, y_i) from the distribution P_Z^m, it sends its information through a private algorithm $\mathcal{S}$ after getting messages $S_1, \cdots, S_{i-1}$. By definition, for $1 \leq j \leq m$, we have

$$\frac{d\mathrm{P}_{z^{\oplus k}}}{d\mathrm{P}_z} = \prod_{j=1}^m \frac{1 + y_{i,j}\langle X_{i,j}, \beta_{z^{\oplus k}} \rangle}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} = \prod_{j=1}^m 1 + \frac{y_{i,j}\langle X_{i,j}, \beta_{z^{\oplus k}} - \beta_z \rangle}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} = \prod_{j=1}^m 1 - \frac{y_{i,j} X_{i,j}^k z_k}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} \cdot \frac{4\sqrt{2}\nu}{\sqrt{s^*}} \tag{8}$$

where the last step follows from Zhu et al. (2023). If we let ν to be small enough, we can guarantee that $|y_{i,j}\langle X_{i,j}, \beta_z \rangle| \leq 1/2$ for each z and $|\frac{y_{i,j} X_{i,j}^k z_k}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} \cdot \frac{4\sqrt{2}\nu}{\sqrt{s^*}}| \leq 1/2$. We compute the $\log$ transformation of the above quantity which is $\sum_{j=1}^m \log\left(1 - \frac{y_{i,j} X_{i,j}^k z_k}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} \cdot \frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)$. For each j, we bound the expectation by Jensen's inequality

$$\mathbb{E}\left[\log\left(1 - \frac{y_{i,j} X_{i,j}^k z_k}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle} \cdot \frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)\right] \leq \log\left(1 - \mathbb{E}\left[\frac{y_{i,j} X_{i,j}^k z_k}{1 + y_{i,j}\langle X_{i,j}, \beta_z \rangle}\right] \cdot \frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right). \tag{9}$$

For each k, we have

$$\left|\mathbb{E}\left[\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\right]\right| \le \left|\frac{1}{2+8\sqrt{2s^*}\nu}-\frac{1}{2-8\sqrt{2s^*}\nu}\right| \le 8\sqrt{2s^*}\nu. \tag{10}$$

Bringing (10) into (9) leads to

$$\mathbb{E}\left[\log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)\right] \le \log\left(1+64\nu^2\right)$$

As a result, the expectation of the log transformation has

$$\mathbb{E}\left[\sum_{j=1}^m \log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{s^*}\right)\right] \le m\cdot\log\left(1+64\nu^2\right) \tag{11}$$

Moreover, since $1-5|x|\le\log(1+x)\le 1+5|x|$ for $|x|\le 1/2$, we have

$$1-\frac{10\sqrt{2}\nu}{\sqrt{s^*}} \le \log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right) \le 1+\frac{10\sqrt{2}\nu}{\sqrt{s^*}}.$$

Recall that $|\{z\in\{-1,1\}^d|\sum_j \mathbf{1}\{z^j=1\}\le s^*\}|\le d^{s^*}$. Thus, applying Hoeffding's inequality with union bound yields

$$\begin{aligned}&\left|\sum_{j=1}^m \log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)-\mathbb{E}\left[\sum_{j=1}^m \log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)\right]\right|\\ \le&\frac{20\nu\sqrt{m(\log n+s^*\log d)}}{\sqrt{s^*}} \le \frac{20\nu\sqrt{md}}{s^*}\end{aligned} \tag{12}$$

for all $1\le i\le n$ and $z\in\{-1,1\}^d$ with $\sum_j \mathbf{1}\{z^j=1\}\le s^*$ with probability at least $1-2/n^2$. As a result, plugging (11) and (12) into (8) yields

$$\begin{aligned}\frac{d\mathrm{P}_{z\oplus k}}{d\mathrm{P}_z} =& \exp\left(\sum_{j=1}^m \log\left(1-\frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}}\right)\right)\\ \le& \exp\left(m\cdot\log\left(1+64\nu^2\right)+\frac{20\nu\sqrt{md}}{s^*}\right).\end{aligned}$$

Since $n\varepsilon^2\ge s^{*2}$, for sufficiently small ν, one can justify condition A.1 and A.2 for the defined P_z, with $\alpha^2\asymp\frac{\nu^2 md}{s^{*2}}$. Applying Corollary 1 in Acharya et al. (2020) leads to

$$\left(\frac{1}{d}\sum_{i=1}^d \mathrm{d}_{\mathrm{TV}}\left(\mathrm{P}_{+i}^{S^n},\mathrm{P}_{-i}^{S^n}\right)\right)^2 \lesssim \frac{nm\nu^2\varepsilon^2}{s^{*2}}. \tag{13}$$

Note that this result, as well as Lemma 3 of Acharya et al. (2020) in the following, are developed for X_i being a single sample. They are extendable to X_i being multiple samples since we can apply the original conclusion to the $m(d+1)$ dimensional vector, formulated by stacking the $X_{i,j}$s. Next we focus on lower bound of the total variation distance. Since

$$\|\beta_z-\beta_{z'}\|_2=\sqrt{\frac{32\nu^2}{s^*}\sum_{i=1}^d \mathbf{1}\left\{Z_i\ne\hat{Z}_i\right\}}=4\nu\left(\frac{d_{\mathrm{Ham}(z,\hat{z})}}{\tau d}\right)^{1/2},$$

i.e. Condition A.3 holds, applying Lemma 3 of Acharya et al. (2020) leads to

$$\frac{1}{d}\sum_{i=1}^d \mathrm{d}_{\mathrm{TV}}\left(\mathrm{P}_{+i}^{S^n},\mathrm{P}_{-i}^{S^n}\right)\ge\frac{1}{4}. \tag{14}$$

Combining (13) and (14) leads to the desired conclusion.

□

Proof of Proposition 2.3. We follow the same construction as in the proof of Theorem 2.4 while adopting a different strategy to bound $\frac{d\mathrm{P}_{z\oplus k}}{d\mathrm{P}_z}$. Namely, we let

$$\frac{d\mathrm{P}_{z\oplus k}}{d\mathrm{P}_z} = \prod_{j=1}^{m} 1 - \frac{y_{i,j}X_{i,j}^k z_k}{1+y_{i,j}\langle X_{i,j},\beta_z\rangle}\cdot\frac{4\sqrt{2}\nu}{\sqrt{s^*}} \le \left(1+\frac{8\sqrt{2}\nu}{\sqrt{s^*}}\right)^m. \tag{15}$$

Then one can justify condition A.1 and A.2 for the defined P_z, with

$$\alpha^2 \asymp \left(\left(1+\frac{8\sqrt{2}\nu}{\sqrt{s^*}}\right)^m - 1\right)^2.$$

Applying Corollary 1 in Acharya et al. (2020) leads to

$$\left(\frac{1}{d}\sum_{i=1}^{d} \mathrm{d}_{\mathrm{TV}}\left(\mathrm{P}_{+i}^{S^n}, \mathrm{P}_{-i}^{S^n}\right)\right)^2 \lesssim \frac{n\varepsilon^2}{d}\cdot\left(\left(1+\frac{8\sqrt{2}\nu}{\sqrt{s^*}}\right)^m - 1\right)^2. \tag{16}$$

There holds similarly

$$\frac{1}{d}\sum_{i=1}^{d} \mathrm{d}_{\mathrm{TV}}\left(\mathrm{P}_{+i}^{S^n}, \mathrm{P}_{-i}^{S^n}\right) \ge \frac{1}{4}. \tag{17}$$

Combining (16) and (17) leads to

$$\exp\left(\frac{\nu m}{\sqrt{s^*}}\right) \asymp \left(1+\frac{8\sqrt{2}\nu}{\sqrt{s^*}}\right)^m \gtrsim 1+\sqrt{\frac{d}{n\varepsilon^2}}.$$

which yields

$$\nu^2 \gtrsim \frac{s^*}{m^2}\log^2\left(1+\sqrt{\frac{d}{n\varepsilon^2}}\right).$$

Note that if $n\varepsilon^2 \lesssim \sqrt{d}$ and $m \le \log d$, there holds

$$\nu^2 \gtrsim \frac{s^*}{m^2}\log^2\left(1+\sqrt{\frac{d}{n\varepsilon^2}}\right) \gtrsim \frac{s^*}{m^2}\log^2\left(1+d^{1/4}\right) \gtrsim \frac{\log^2 d}{s^*\log^2 d} = \frac{1}{s^*}$$

which yields the desired result. Note that in this case, the constructed function class has beta-min condition with $a = \nu/\sqrt{s^*} \gtrsim 1$ which is a constant in a $[0,1]$.

□

B. Candidate Variable Selection

B.1. Good Selectors

B.1.1. Plug-in High Dimensional Variable Selection

In the following, we provide some example selectors and demonstrate that, under mild assumptions, they serve as components of a good selector. We introduce commonly used variable selection approaches along with their associated theoretical results. Our goal is twofold. Firstly, we want the true variables to be selected. Conversely, the redundant variables that are selected should be as few as possible. We derive this from the perfect selection property (also known as strong oracle or consistent selection), which asserts that our goal is achieved with a high probability. The primary conditions we impose on the potential distributions fall into two categories:

- Beta-min conditions, which necessitate that $\min_{\beta^{*j}>0}|\beta^{*j}|$ is greater than a specified threshold. With this condition, the signal strength from the regression functions is robust enough for the selector to identify the variables..
- Mild correlation conditions, which require that the correlation between the true and redundant variables is weak enough for the selectors to distinguish.

In this section, we omit the user index i and write (X, y) representing the data of some user (X_i, y_i), since the results in the section consider one local dataset at a time.

Example B.1 (**Lasso (Tibshirani, 1996)**). *Lasso, or Least Absolute Shrinkage and Selection Operator, is a regularization technique in statistical learning that adds a penalty term to the linear regression objective function, effectively promoting sparsity by encouraging some of the model coefficients to be exactly zero. Specifically, Lasso solves the regularized optimization object*

$$\min_{\beta\in\mathbb{R}^d}\left\{\frac{1}{n}\|y-X\beta\|_2^2+\lambda\|\beta\|_1\right\}. \tag{18}$$

Used for variable selection, Lasso identifies the non-zero elements of the optimization solution as the selected variable.

To study the selection consistency of Lasso, Zhao & Yu (2006) proposed a general condition called the Irrepresentable condition. Specifically, for $\widehat{\Sigma}=X^\top X/n$, let the block matrix

$$\widehat{\Sigma}=\begin{pmatrix}\widehat{\Sigma}_{11} & \widehat{\Sigma}_{12}\\ \widehat{\Sigma}_{21} & \widehat{\Sigma}_{22}\end{pmatrix}.$$

Here $\widehat{\Sigma}_{11}$ is a $s^*\times s^*$ matrix, corresponding to the covariance matrix of the true variables. Irrepresentable Condition states that there exists a positive constant vector η

$$\left|\widehat{\Sigma}_{21}\left(\widehat{\Sigma}_{11}\right)^{-1}\operatorname{sign}\left(\beta^{*1:s^*}\right)\right|\le \mathbf{1}-\eta, \tag{19}$$

where $\mathbf{1}-\eta$ is a $d-s^*$ vector with $1-\eta$ elementwisely. The following result holds for irrepresentable condition.

Lemma B.2. *Under our assumptions, when using (18) as selector, let β_{LASSO} be the solution. Suppose (19) holds. Suppose the following conditions hold: (i) $m\gtrsim s^{*2}\log d$. (ii) $\min_{\beta^{*j}>0}|\beta^{*j}|\gtrsim\sqrt{1/m}$. Then there exists a constant $C_p<1$ such that, for sufficiently large m, with probability C_p, there holds*

$$\beta^j_{LASSO}\neq 0 \quad \textit{for} \quad j=1,\cdots,s^* \quad \textit{and} \quad \beta^j_{LASSO}=0 \textit{ for } j=s^*+1,\cdots,d.$$

Moreover, for $\Sigma=\mathbb{E}XX^\top$, if $|\Sigma_{ij}|\le 3/s^*$ for $i\neq j$, then we have the Irrepresentable Condition.

Proof of Lemma B.2. Since we assume sub-Gaussian noises, any k-th moment of the random noise exists, i.e. k can be arbitrarily large. As a result, any $\lambda\gtrsim\sqrt{m}$ implies $(\lambda/\sqrt{m})^2k/d\to\infty$ for some k. By Theorem 3 in Zhao & Yu (2006), for sufficiently large m, the probability of

$$\operatorname{sign}(\beta^j_{LASSO})=\operatorname{sign}(\beta^{*j}) \quad \text{for } j=1,\cdots,d$$

is larger than some constant C_p, given that the conditions (5,6,7,8) are satisfied. Thus it suffices to verify the conditions. Condition (5) and (6) holds naturally due to our assumption of i.i.d. designs and boundedness of covariance matrix norm. (7) and (8) are in our assumptions. As for the last statement, Zhao & Yu (2006) provides several commonly seen sufficient conditions for the irrepresentable condition to hold, such as when $|\widehat{\Sigma}_{ij}|\le 1/(2s^*-1)$. If $|\Sigma_{ij}|\le 1/3s^*$, then $|\widehat{\Sigma}_{ij}|\le|\Sigma_{ij}|+|\Sigma_{ij}-\widehat{\Sigma}_{ij}|\le 1/3s^*+c/\sqrt{m}\le 1/(2s^*-1)$ for some constant c and sufficiently large $m\gtrsim s^*$. This bound holds for all users and all position i, j if we apply union bound, where we need $\log d/m\lesssim 1/s^{*2}$, i.e. $m\gtrsim s^{*2}\log d$. Note here we assumed $d\gtrsim n$. Thus the lemma is proved. □

Example B.3 (**SCAD (Fan & Li, 2001)**). *SCAD, or smoothly clipped absolute deviation, is a non-convex penalty function used in statistical learning and regression analysis. It is designed to address limitations of traditional L1 regularization methods like Lasso by providing a smooth and more robust penalty on regression coefficients, promoting sparsity while*

mitigating some of the biases associated with sharp discontinuities in penalty functions. Specifically, SCAD solves the regularized optimization object

$$\min_{\beta\in\mathbb{R}^d}\left\{\frac{1}{n}\|y-X\beta\|_2^2+\lambda\sum_{j=1}^{d}\psi_\lambda\left(\beta_j\right)\right\} \text{ where } \psi'_\lambda(t)=\lambda I_{\{t\le\lambda\}}+\frac{(a\lambda-t)_+}{a-1}I_{\{t>\lambda\}} \text{ for some } a>2. \tag{20}$$

Used for variable selection, SCAD identifies the non-zero elements of the optimization solution as the selected variable.

The following lemma, which is a straightforward implication of Fan & Lv (2011), states that the essential condition for SCAD estimator to consistently select the variables is the Beta-min condition, given that the sample size is relatively large.

Lemma B.4. *Under our assumptions, when using (20) as selector, let β_{SCAD} be the solution. Suppose the following conditions hold: (i) The sparsity s^* is $\mathcal{O}(1)$. (ii) $m\gtrsim s^*\vee\log m\log d$. (iii) $\min_{\beta^{*j}>0}|\beta^{*j}|\gtrsim\sqrt{s^*/m}\vee\sqrt{\log d\log m/m}$. Then there exists a constant $C_p<1$ and a suitable choice of λ_m such that, for sufficiently large m, with probability C_p, there holds*

$$\beta^j_{SCAD}\neq 0 \quad \text{for} \quad j=1,\cdots,s^* \quad \text{and} \quad \beta^j_{SCAD}=0 \text{ for } j=s^*+1,\cdots,d.$$

Proof of Lemma B.4. By Theorem 3 in Fan & Lv (2011), for sufficiently large m, the probability of

$$\|\beta_{SCAD}-\beta^*\|_2\lesssim\sqrt{\frac{s^*}{m}} \quad \text{for} \quad j=1,\cdots,s^* \quad \text{and} \quad \beta^j_{SCAD}=0 \text{ for } j=s^*+1,\cdots,d$$

is larger than some constant C_p, given that the regularity conditions in the theorem are satisfied. Note that we have $|\beta^j_{SCAD}|\ge|\beta^{*j}|-\sqrt{s^*/m}\ge|\beta^{*j}|/2>0$. Thus it suffices to verify the conditions. The condition 1 is satisfied by SCAD penalty. Condition 5 is satisfied by our setting of sample size (note that $\log d\lesssim n^{\alpha'}$ for α' defined in their context). (26) and (28) of Condition 2 follows from our assumptions on the upper and lower bound of $\|\mathbb{E}XX^\top\|_2$ and the estimation error of covariance matrix which is $\mathcal{O}(\sqrt{s^*/m})$ (Wainwright, 2019). (27) comes from $s^*=\mathcal{O}(1)$. □

B.1.2. Proof of Proposition 3.2

Proof of Proposition 3.2. Under the two conditions, using Lemma B.2 and B.4, we can show that there exists a variable selection method that perfectly select the true variables with a positive probability C_p. Then by sampling among the selected variables, the probability can be computed as

$$\Pr\left(\mathcal{S}(X_i,y_i)=v\right)\ge C_p\Pr\left(v=j \text{ for } v\sim\text{Unif}\left(1,\cdots,s^*\right)\right)\ge\frac{C_p}{s^*}$$

for $1\le v\le s^*$. This yields the desired conclusion. □

B.1.3. Computational Issue

For SCAD, the incorporation of a non-convex penalty proves effective in attaining coefficient sparsity while maintaining oracle properties. Nonetheless, the non-convex nature introduces a challenge—the guarantee of solution uniqueness becomes elusive, leading to the presence of multiple local optima. Consequently, the stability of results may be compromised. Fan et al. (2014) introduce additional concave parameter to ensure consistency, which contributes to increased computational complexity, further posing challenges in the computational efficiency of SCAD. As a result, Lasso is more preferable In practice. We introduce another technique which can be useful to enhance the computation efficiency.

Example B.5 (Screening (Fan & Lv, 2008)). *Sure Independence Screening (SIS) is a feature selection method in statistical learning that aims to identify relevant variables in high-dimensional datasets. It does so by assessing the correlation between each predictor and the response variable, and selecting a subset with the highest scores. Specifically, For $(X,y)\in(\mathcal{X}\times\mathcal{Y})^m$, let*

$$w=X^\top y.$$

Then the s most largest position of w are identified as the selected variables. Screening can be a valuable pre-procedure for other selection methods. Screening is employed to quickly identify and retain a subset of potentially important features, reducing the dimensionality of the data before applying more computationally intensive or elaborate feature selection techniques.

B.2. Aggregation of Local Selected Variables

In this section, we present the omitted algorithm and technical proofs for the aggregation step after local variable selection. In B.2.1, we introduce the detailed variable selection algorithm. In B.2.2, we present proofs omitted in Section 3.1.

B.2.1. Heavy Hitter Algorithm

First, we introduce necessary definitions. Let $\mathcal{V}$ be a collection of binary prefixes. The define $\texttt{ChildSet} = \{v+0, v+1 \text{ for } v \in \mathcal{V}\}$. We define several public randomness that will be shared among users. See Bassily et al. (2020, Section 3.1) for details. Let $\overline{\mathcal{V}} = \left\{v \in \{0,1\}^{\ell} \text{ for some } \ell \in [\log d]\right\}$. Define integer $t = 3\log(n)$ and $k = O(\sqrt{n/3\log(n)})$. We will consider a set of t pairs of hash functions $\{(h_1, g_1), \ldots, (h_t, g_t)\}$, where for each $i \in [t], h_i : \overline{\mathcal{V}} \to [k]$ and $g_i : \overline{\mathcal{V}} \to \{-1, +1\}$ are independently and uniformly chosen pairwise independent hash functions. We assume that the server creates a random partition $\Pi : [n] \to [\log d] \times [k]$ that assigns to each user $i \in [n]$ a random pair $(\ell_i, j_i) \leftarrow [\log(d)] \times [k]$, as in the initialization of Algorithm 4. We also have another random function $\mathcal{Q} : [n] \leftarrow [k]$ that assigns to each user i a uniformly random index $r_i \leftarrow [k]$. We assume that such random indices ℓ_i, j_i, r_i are shared between the server and each user. Finally, we adopt shared encoding and decoding schemes for bijection between $[d]$ and $\lceil \log d \rceil$ binary strings, denoted as `Encoding` and `Decoding`, respectively.

Before presenting the `HeavyHitter`, we first introduce the functions it uses. The following algorithm generate a private report for a single user. We seal the information of each v_i into a binary value that is the Hardamard transform of hashes of its prefix. The information is privatized using the random response mechanism (Warner, 1965) and sent to the curator.

Algorithm 2 `LocalRnd` (Bassily et al., 2020)

Input: Privacy budget ε, input v_i.
Compute $\tilde{v}_i = \texttt{Encoding}(v_i)$ the binary string encoding.
Using pubic randomness to get (ℓ_i, j_i) and r_i.
Let $s_i := g_{j_i}(\tilde{v}_i[1:\ell_i])$ and $c_i := h_{j_i}(\tilde{v}_i[1:\ell_i])$. Here $v[1:\ell]$ denote the ℓ -bit prefix of v.
Compute $x_i = s_i \cdot W_{r_i, c_i}$. Here $W_{r,c}$ denotes the sign of (r, c) entry of Hadamard matrix with size k.
Random permute x_i with

$$y_i = \begin{cases} x_i & \text{w.p. } \frac{e^{\epsilon}}{e^{\epsilon}+1} \\ -x_i & \text{w.p. } \frac{1}{e^{\epsilon}+1} \end{cases}$$

Output: y_i.

The following algorithm shows how `LocalRnd` is invoked multiple times to scan the prefix tree.

Algorithm 3 `FreqOracle` (Bassily et al., 2020)

Input: Prefixes length ℓ, a subset of ℓ-bit prefixes $\widehat{\mathcal{V}} \subseteq \{0,1\}^{\ell}$, collection of t disjoint subsets of users: $\left\{\tilde{\mathcal{I}}_j : j \in [t]\right\}$, privacy budget ε.
for $\widehat{v} \in \widehat{\mathcal{V}}$ **do**
  for Hash index $j = 1$ to t **do**
    Let $s := g_j(\widehat{v})$ and $c := h_j(\widehat{v})$.
    for $i \in \tilde{\mathcal{I}}_j$ **do**
      y_i = `LocalRnd`(ε, v_i).
    end for
    Compute the j-th estimate of the frequency of $\widehat{v}$: $\widehat{f}_j(\widehat{v}) = t\log d \cdot \frac{e^{\varepsilon}+1}{e^{\varepsilon}-1} \sum_{i \in \tilde{\mathcal{I}}_j} y_i \cdot s \cdot W_{r_i, c}$.
  end for
  The final estimation of $\widehat{v}$: $\widehat{f}(\widehat{v}) := \text{Median}\left(\left\{\widehat{f}_j(\widehat{v}) : j \in [t]\right\}\right)$.
end for
FreqList $= \{(\widehat{v}, \widehat{f}(\widehat{v})) : \widehat{v} \in \widehat{\mathcal{V}}\}$.
Output: FreqList

The final algorithm is presented in Algorithm 4. We modify the algorithm in Bassily et al. (2020) by removing the second

phase of frequency estimation, since we only want to identify the heavy hitters and do not care about their frequencies. This allows a saving of $\varepsilon/2$ budget.

Algorithm 4 `HeavyHitter`

Input: User values $\mathcal{V} = \{v_i \in [d]\}$, privacy budget ε, threshold ρ.
Initialization: Prefixes $= \{\}$, public randomness pairs $\Gamma = \{(\ell_i, j_i) \in [\log d] \times [3\log n]$ for $1 \le i \le n\}$, partition $I_{\ell,j} = \{i$ if $(\ell_i, j_i) = (\ell, j)\}$.
for ℓ in $1, \cdots, \lceil \log d \rceil$ **do**
  $\{(\widehat{v}, \widehat{f}(\widehat{v})) : \widehat{v} \in$ `ChildSet`(Prefixes)$\} =$ `FreqOracle`$(\ell,$ ChildSet (Prefixes) $, \{\mathcal{I}_{\ell,j} : j \in [3\log n]\}, \varepsilon)$.
  Let NewPrefixes $= \{\}$.
  for $v \in$ `ChildSet`(Prefixes) **do**
    if $\widehat{f}(\widehat{v}) \ge \rho n$ **then**
      Add $\widehat{v}$ to NewPrefixes.
    end if
  end for
  if $|$NewPrefixes$| = 0$ **then**
    Add $\arg\max_{\widehat{v}} \widehat{f}(\widehat{v})$ to NewPrefixes. `# Ensure NewPrefixes is non-empty.`
  end if
  Prefixes $\leftarrow$ NewPrefixes.
end for
Output: $\{$`Decoding`(v) for $(v, \widehat{f}(v)) \in$ Prefixes$\}$.

B.2.2. Proof Related to Section 3.1

To give the proof of Proposition 3.3, we need the following necessary technical result.

Lemma B.6. *Algorithm 4 is ε- ULDP. Moreover, if $\alpha \gtrsim s^* \sqrt{\log n \log d/n}/\varepsilon$, then with probability at least $1 - 1/n^2$, the output list of the* `HeavyHitter` *protocol satisfies the following properties given sufficiently large n: (i) it contains all items $v \in \mathcal{V}$ whose true frequencies above $2\rho n$. (ii) it does not contain any item $v \in \mathcal{V}$ whose true frequency below $\rho n/2$.*

Proof of Lemma B.6. Lemma 5.3 in Bassily et al. (2020) yields that the variables v retained in Prefixes in Algorithm 4 has $|\widehat{f}(v) - f(v)| \lesssim \sqrt{n \log n \log d}/\varepsilon$. Since $\alpha \gtrsim s^* \sqrt{\log n \log d/n}/\varepsilon$, we have $\sqrt{n \log n \log d}/\varepsilon \le \rho n/2$ for sufficiently large n. Then for any v in Prefixes, we have $f(v) \gtrsim \rho n - \sqrt{n \log n \log d}/\varepsilon \ge \rho n/2$. On the contrary, if $f(v) \ge 2\rho n$, then $\widehat{f}(v) \gtrsim 2\rho n - \sqrt{n \log n \log d}/\varepsilon \ge \rho n$, which will be included in Prefixes.

□

Proof of Proposition 3.3. For notation simplicity, we denote the number of users and selectors used in the selection as n instead of $n/2$ throughout this proof. We compute the frequency of variable j, namely $\sum_{i=1}^n \mathbf{1}(v_i = j)$. By Hoeffding's inequality, we have

$$\Pr\left(\left|\sum_{i=1}^n \mathbf{1}(v_i = j) - \sum_{i=1}^n \Pr(v_i = j)\right| \ge \sqrt{n(\log nd)}\right) \le 2\exp\left(-2(\log n + \log d)\right).$$

Applying union bound, we get

$$\begin{aligned}\Pr\left(\left|\sum_{i=1}^n \mathbf{1}(v_i = j) - \sum_{i=1}^n \Pr(v_i = j)\right| \ge \sqrt{n \log nd} \quad \text{for } 1 \le j \le d\right) \le & 2d\exp\left(-2(\log n + \log d)\right) \\ < & \exp\left(-2\log n\right) = 1/n^2. \end{aligned} \tag{21}$$

For conclusion (*i*), since v_i is generated by a good selector, Definition 3.1 yields that

$$\sum_{i=1}^n \Pr(v_i = j) \ge \frac{n\alpha}{s^*}$$

for $j = 1, \cdots, s^*$. This together with (21) leads to

$$\sum_{i=1}^{n} \mathbf{1}\left(v_i = j\right) \geq \frac{n\alpha}{s^*} - \sqrt{n \log nd} \geq \frac{n\alpha}{2s^*}$$

for any $1 \leq j \leq s^*$ and sufficiently large n. Then for any $\rho \leq \alpha/4s^*$, by Lemma B.6, we have $\widehat{f}(j) \geq \rho n$. This means the frequency of any true variable must be large enough to be detected as a heavy hitter. Next, we show (*ii*). Suppose that there are s variables $j_1, \cdots, j_s$ satisfying $\sum_{i=1}^{n} \mathbf{1}\left(v_i = j\right) \geq \rho n/2$, i.e. potentially identified by the heavy hitters by Lemma B.6. Then by applying (21), there holds

$$\sum_{k=1}^{s} \sum_{i=1}^{n} \Pr\left(v_i = j_k\right) \geq s \cdot \frac{\rho n}{2} - s\sqrt{n \log nd} \geq s \cdot \frac{\rho n}{4}$$

for sufficiently large n, with probability at least $1 - 1/n^2$. However, there holds

$$s \cdot \frac{\rho n}{4} \leq \sum_{k=1}^{s} \sum_{i=1}^{n} \Pr\left(v_i = j_k\right) \leq \sum_{j=1}^{d} \sum_{i=1}^{n} \Pr\left(v_i = j\right) = n,$$

which indicates that $s \leq 4/\rho \leq 32s^*/\alpha$.

□

C. Coefficient Estimation

C.1. The Multiple Round Protocol

C.1.1. SCO Algorithm

We use the same algorithm as in the Bassily & Sun (2023) while adopting a different set of default values of its parameters. Such changes are due to the differential technical requirements for the theoretical analysis with strong convexity. Also, the algorithm requires a solution to the user-level locally differentially private mean estimation (`ULDPMean`), which is presented later in Section C.2.2. For notation simplicity, we denote the number of users and selectors used in the selection as n instead of $n/2$ in this section.

Algorithm 5 `ULDPSCO`

Input: Local data sets $\{(X_i, y_i)\}_{i=1}^{n}$, number of iterations T, concentration radius τ, privacy budget ε.
Initialization : $\beta_0 = \overrightarrow{0}$, $\beta^{ag} = \beta_0$, and $\{\eta_t, \gamma_t\}_{t \in [T]}$ as in Lemma C.2.
for $t = 0, 1, \cdots, T-1$ **do**
  Compute $\beta_t^{md} = \gamma_t^{-1}\beta_t + \left(1 - \gamma_t^{-1}\right)\beta_t^{ag}$.
  Choose two fresh batches $S_{t,1}$ and $S_{t,2}$ of $n_0 = \lfloor n/2T \rfloor$ users, respectively.
  Compute the average gradient at each user at β_t, $g_i\left(\beta_t^{md}\right) = \frac{1}{mL}\sum_{j=1}^{m}\left(X_{i,j}^{\top}\beta_t^{md} - y_{i,j}\right)X_{i,j}$ for $i \in S_{t,1} \cup S_{t,2}$.
  Compute the average gradients $\tilde{\nabla}F\left(\beta_t^{md}\right) = \texttt{ULDPMean}(\{g_i\left(\beta_t^{md}\right)_i\}_{i \in S_{t,1}}, \{g_i\left(\beta_t^{md}\right)_i\}_{i \in S_{t,2}}, \tau, \varepsilon)$.
  Update $\beta_{t+1} = \beta_t^{md} - \eta_t \cdot L \cdot \tilde{\nabla}F\left(\beta_t^{md}\right)$.
  Compute $\beta_{t+1}^{ag} = \gamma_t^{-1}\beta_{t+1} + \left(1 - \gamma_t^{-1}\right)\beta_t^{ag}$.
end for
Output: β_T^{ag}.

For the algorithm, we have the following result. Note that Algorithm 5 adopts disjoint mini-batch when computing the gradients while Lemma C.1 was established completely based on stochastic gradient descent. Yet, the theoretical analysis generalize straightforwardly.

Lemma C.1 (**Theorem 3 of Dieuleveut et al. (2017)**). *Consider the stochastic convex optimization problem* (3). *Suppose each $\tilde{\nabla}F(\beta)$ is an unbiased stochastic oracle to $\nabla F(\beta)$ with variance ν^2. Let $\beta_T'^{ag}$ be the associated non-private output of*

Algorithm 5 ($\varepsilon = \infty$). There exists settings of $\{\eta_t, \gamma_t\}_{t\in[T]}$ such that

$$\mathbb{E}\left[F\left(\beta_T'^{ag}\right) - \min_{\beta} F(\beta)\right] \lesssim \frac{s\nu^2}{T} + \frac{\|\beta^*\|_2^2 \lambda_n \left(\mathbb{E}\left[XX^\top\right]\right)^{-1}}{T^2}.$$

For clearness, we additionally include the full multi-round protocol.

Algorithm 6 Multi-round ULDP sparse linear regression.

Input: Local data sets $\{(X_i, y_i)\}_{i=1}^n$, selectors $\{\mathcal{S}_i\}_{i=1}^{n/2}$, privacy budget ε, threshold ρ.
Initialization: $\beta \in \mathbb{R}^d$ be a zero vector.
# candidate variable selection
# on local machine
for i in $1, \cdots, n/2$ **do**
  $v_i = \mathcal{S}_i(X_i, y_i)$.
end for
# $\lceil \log d \rceil$ round communication
$\{\widehat{v}_1, \cdots, \widehat{v}_s\}$ = HeavyHitter($\{v_i\}_{i=1}^{n/2}, \varepsilon, \rho$).
coefficient estimation
$n \wedge \sqrt{nm\varepsilon^2}$ round communication
$\widehat{\beta}$ = ULDPSCO($\{(X_i, y_i)\}_{i=n/2+1}^n, T, \tau, \varepsilon$).
$\beta^{\widehat{v}_1:\widehat{v}_s} = \widehat{\beta}$.
Output: β.

C.1.2. Proof of Theorem 3.4

We need the following technical result which states the effectiveness of optimization procedures in Algorithm 5.

Lemma C.2. *Consider the stochastic convex optimization problem (3). Let $T = n \wedge \sqrt{nm\varepsilon^2}$, and $\{\eta_t, \gamma_t\}_{t\in[T]}$ as in Lemma C.2, $L = 6s^3 \log n$, and $\tau \asymp L\sqrt{\log n \log(n \vee m) \log T/m}$. Then Algorithm 5 is ε-ULDP and has*

$$\mathbb{E}\left[\left\|\beta_T^{ag} - \widehat{\beta}^*\right\|_2^2\right] \lesssim \mathbb{E}\left[F\left(\beta_T^{ag}\right) - F(\widehat{\beta}^*)\right] \lesssim \frac{s^9 \log^6 n}{nm\varepsilon^2} + \frac{s^4 \log n}{nm}.$$

Proof of Lemma C.2. The privacy guarantee comes from the privacy of Algorithm 9 and the fact that each batch of samples are disjoint. For the privacy guarantee, consider the mean squared error

$$F(\beta) = \int_{\widehat{\mathcal{X}}\times\mathcal{Y}} \left(x^\top\beta - y\right)^2 d\mathrm{P}(x, y) = \mathbb{E}\left[\sigma^2\right] + (\beta - \beta^*)^\top \Sigma (\beta - \beta^*).$$

By assumption on $\Sigma = \mathbb{E}\left(XX^\top\right)$, we have

$$F(\beta) - \inf_{\beta} F(\beta) = F(\beta) - F(\widehat{\beta}^*) = (\beta - \beta^*)^\top \Sigma (\beta - \beta^*) \geq C_X^{-1}\|\beta - \beta^*\|_2^2.$$

Thus, it suffices to bound $\mathbb{E}\left[F(\beta) - F(\widehat{\beta}^*)\right]$ and the estimation error is of the same order. Note that under the assumption $\|\beta^*\|_2 \leq 1$, the squared loss function $\ell(\beta)$ constraint on the unit ball has

$$\|\nabla\ell(\beta)\|_2 \leq \|(x^\top\beta - y)x\|_2 \lesssim s^3 \log n$$

i.e. $s^3 \log n$-Lipschitzness. Let $(\beta_1^{ag}, \ldots, \beta_T^{ag})$ be the parameter trajector of Algorithm 5. Let $\left(\beta_1'^{ag}, \ldots, \beta_T'^{ag}\right)$ be the parameter trajectory of another algorithm which replaces the gradient estimate $\tilde{\nabla}F\left(\theta_t^{\mathrm{md}}\right)$ by

$$\tilde{\nabla}F'\left(\beta_t^{md}\right) \sim \frac{1}{n_0} \sum_{i\in S_{t,1}\cup S_{t,2}} g_i\left(\beta_t^{md}\right) + \mathrm{Lap}\left(0, \frac{6\tau}{\varepsilon} I_d\right).$$

By analysis analogous to the proof of Lemma C.5, if we take $\tau \asymp L\sqrt{\log n \log(n \vee m) \log T/m}$, there holds

$$\beta_t^{ag} \overset{\mathcal{D}}{=} \beta_t^{'ag}$$

with probability $1 - 1/nm$ for all $1 \le t \le T$, where $\overset{\mathcal{D}}{=}$ stands for equal in distribution. Hence we have

$$\mathbb{E}\left[F\left(\beta_T^{ag}\right)\right] \le \mathbb{E}\left[F\left(\beta_T^{'ag}\right)\right] + \frac{s^4 \log n}{nm}. \tag{22}$$

For $\mathbb{E}\left[F\left(\beta_T^{'ag}\right)\right]$, we use the fact that $\mathbb{E}\left[\tilde{\nabla} F'\left(\beta_t^{md}\right)\right] = \nabla F\left(\beta_t^{md}\right)$ and, by Lemma C.5,

$$\mathbb{E}\left[\left\|\tilde{\nabla} F'\left(\beta_t^{md}\right) - \nabla F\left(\beta_t^{md}\right)\right\|_2^2\right] \lesssim \frac{L^2 s^2 \log^3 n \log T}{n_0 m \varepsilon^2} + \frac{s \log n}{n_0 m}.$$

Applying Lemma C.1, we have

$$\mathbb{E}\left[F\left(\beta_T^{'ag}\right) - \min_{\beta} F(\beta)\right] \lesssim \frac{s^9 \log^5 n \log T}{nm\varepsilon^2} + \frac{s^2 \log n}{nm} + \frac{s}{T^2}.$$

Taking $T \asymp n \wedge \sqrt{nm\varepsilon^2}$, this together with (22) lead to

$$\mathbb{E}\left[F\left(\beta_T^{ag}\right) - F(\widehat{\beta}^*)\right] \lesssim \frac{s^9 \log^6 n}{nm\varepsilon^2} + \frac{s^4 \log n}{nm}.$$

□

Theorem C.3 (**Formal version of Theorem 3.4**). *Let data $\{(X_i, y_i)\}_{i=1}^n$ be generated as in (1). Suppose $\{\mathcal{S}_i\}_{i=1}^n$ are α-good selectors with $\alpha \gtrsim s^* \sqrt{\log n \log d/n\varepsilon^2}$. Suppose we let $\alpha/8s^* \le \rho \le \alpha/4s^*$, $T = n \wedge \sqrt{nm\varepsilon^2}$, and $\{\eta_t, \gamma_t\}_{t\in[T]}$ as in Lemma C.2, $L = 6s^3 \log n$, $\tau \asymp L\sqrt{\log n \log(n \vee m) \log T/m}$. Let β be the output of Algorithm 6. Then we have (i) Algorithm 6 is ε-ULDP. (ii) there holds*

$$\mathbb{E}\left[\|\beta^* - \beta\|_2^2\right] \lesssim \frac{s^9 \log^6 n}{nm\varepsilon^2} + \frac{s^4 \log n}{nm}.$$

Proof of Theorem C.3. By Lemma B.6 and C.2, both `HeavyHitter` and `ULDPSCO` are ε-ULDP. Since their associated users do not cross, we have Algorithm 6 is also ε-ULDP. As for (*ii*), by Proposition 3.3, we know that all the non-zero variables of β^* is included in $\{\widehat{v}_1, \cdots, \widehat{v}_s\}$ with probability $1 - 1/n^2$. Thus, we have

$$\|\beta^* - \beta\|_2^2 = \left\|\widehat{\beta}^* - \widehat{\beta}\right\|_2^2.$$

Applying Lemma C.2, this leads to

$$\mathbb{E}\left[\|\beta^* - \beta\|_2^2\right] \lesssim \mathbb{E}\left[\left\|\beta_T^{ag} - \widehat{\beta}^*\right\|_2^2\right] \lesssim \frac{s^9 \log^6 n}{nm\varepsilon^2} + \frac{s^4 \log n}{nm} + \frac{1}{n^2} \lesssim \frac{s^{*9} \log^6 n}{nm\varepsilon^2\alpha^9} + \frac{s^{*4} \log n}{nm\alpha^4},$$

where in the last step we used $s \lesssim s^*/\alpha$ as in Proposition 3.3. The additional term $1/n^2$ is due to the failure probability of Proposition 3.3 and is omitted since it is adjustable to any level with a constant multiplicative cost on the other terms. □

C.2. The Two Round Protocol

C.2.1. Proof of Proposition 3.5

Proof of Proposition 3.5. For the first conclusion, consider th local OLS estimator on selected variables of user i, which is $\widehat{\beta} = (\widehat{X}_i^\top \widehat{X}_i)^{-1} \widehat{X}_i^\top y_i$. Given the fact that $m \ge s$, $\widehat{X}_i^\top \widehat{X}_i$ is invertible and we have

$$\widehat{\beta}_i = (\widehat{X}_i^\top \widehat{X}_i)^{-1} \widehat{X}_i^\top \widehat{y}_i = (\widehat{X}_i^\top \widehat{X}_i)^{-1} \widehat{X}_i^\top (\widehat{X}_i \widehat{\beta}^* + \sigma_i) = \widehat{\beta}^* + (\widehat{X}_i^\top \widehat{X}_i)^{-1} \widehat{X}_i^\top \sigma_i,$$

where $\sigma_{i,j}$ are i.i.d. sub-Gaussian random variables for $1 \le j \le m$. Therefore, the first argument follows from

$$\mathbb{E}[\widehat{\beta}_i] = \widehat{\beta}^* + (\widehat{X}_i^\top \widehat{X}_i)^{-1} \widehat{X}_i^\top \mathbb{E}[\sigma_i] = \widehat{\beta}^*.$$

By implication of Hsu et al. (2012, Theorem 2.1), we have

$$\Pr\left(\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \ge \sqrt{3 \log n \cdot \operatorname{tr}\left[\left(\widehat{X}_i^\top \widehat{X}_i \right)^{-1} \right] \cdot \mathbb{E}[\sigma_{i,j}^2]} \right) \le 1 - \frac{1}{n^3}.$$

This together with covariance matrix estimation bounds (e.g. Wainwright (2019, Theorem 6.5)) lead to

$$\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \lesssim \sqrt{\frac{\operatorname{tr}[\widehat{\Sigma}^{-1}] \log n}{m}} \lesssim \sqrt{\frac{s \log n}{m}} \tag{23}$$

with probability $1 - 1/n^3$. Applying union bound, (23) holds for all $i = n/2+1, \cdots, n$ with probability at least $1 - 1/n^2$. For the second statement, if either conditions in Proposition 3.2 holds, we can adopt Lasso (or SCAD) on the selected variables. See Example B.1 and B.3. The oracle results in Belloni & Chernozhukov (2013) (or Fan & Lv (2011)) yield the concentration bound with the true sparsity parameter

$$\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \lesssim \sqrt{\frac{s^* \log n}{m}}$$

for all $i = n/2+1, \cdots, n$ with probability at least $1 - 1/n^2$. □

C.2.2. ULDP Mean Estimation

We borrow the idea from Girgis et al. (2022) while slight modifications are made. The estimation is conducted in two stages. In the first stage, a histogram partition of $\widehat{\mathcal{X}}$ with bin width $\sqrt{\log^2 n/m}$ is created. The server privately estimates the range in which the means $\widehat{\beta}_i$ lie with high probability (Algorithm 7). In the second stage, each user projects its $\widehat{\beta}_i$ into the determined range from the first step. Then, all users send the LDP versions of their projected $\widehat{\beta}_i$ to the curator (Algorithm 8). Both steps are scalar operations. In the vector case, instead of applying them to each dimension separately, random rotation (Levy et al., 2021) is adopted to eliminate a superfluous factor of $\mathcal{O}(\sqrt{s})$. The full algorithm is summarized in Algorithm 9. We only consider pure differential privacy here and utilize Laplace noise instead of Gaussian in Girgis et al. (2022).

Algorithm 7 `Range`

Input: Scalars $\{y_i\}$, concentration radius τ, privacy budget ε.
`# user side`
All users divide the interval $[-1, 1]$ into $k = 1/\tau$ disjoint intervals, each with width 2τ. Let $\mathcal{T} := \{a_1, a_2, \ldots, a_k\}$ be the index set of middle points of intervals.
for y in $\{y_i\}$ **do**
  Compute $\nu = \arg\min_{a_j \in \mathcal{T}} |y - a_j|$.
  Uniformly sample $j \in [k]$.
  Compute $p = H_k^{\top j} \cdot e_\nu / \sqrt{k}$, where e_ν denotes the basis vector corresponding to ν and H_k is a size k Hadamard matrix.
  Compute vector z_i :

$$z_i = \begin{cases} +H_k^{\top j} \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} & \text{w.p. } \frac{1}{2} + \frac{\sqrt{k} \cdot p}{2} \frac{e^\varepsilon-1}{e^\varepsilon+1} \\ -H_k^{\top j} \cdot \frac{e^\varepsilon+1}{e^\varepsilon-1} & \text{w.p. } \frac{1}{2} - \frac{\sqrt{k} \cdot p}{2} \frac{e^\varepsilon-1}{e^\varepsilon+1} \end{cases}$$

end for
`# curator side`
$\overline{z} = \sum z_i$ and $\ell = \arg\max_j \overline{z}^j$.
Output: Bin $[a_\ell - 3\tau, a_\ell + 3\tau]$.

Let the standard Laplace random variable have probability density function $e^{-|x|}/2$ for $x \in \mathbb{R}$.

Algorithm 8 `Mean`

Input: Scalars $\{y_i\}_{i=1}^n$, concentration range $[a, b]$, privacy budget ε.
`# user side`
for i in $1, \cdots, n$ **do**
  Let $\tilde{y}_i = \Pi_{[a,b]} y_i + \text{Lap}(0, |b-a|/\varepsilon)$, where $\Pi_{[a,b]}$ is the projection onto $[a, b]$.
end for
`# curator side`
Output: $\sum \tilde{y}_i / n$.

Algorithm 9 `ULDPMean`

Input: Two groups of local coefficients $\mathcal{B}_1 = \{\beta_i\}_{i=1}^{n/2}$ and $\mathcal{B}_2 = \{\beta_i\}_{i=n/2}^{n}$, concentration radius τ, privacy budget ε.
Initialization: Let $D = \text{Diag}(w)$ and $U = H_s D/\sqrt{s}$, where $w_i \sim \text{Unif}\{-1, 1\}$ and H_s is a size s Hadamard matrix. Let z be a s dimensional zero vector.
`# histogram selection`
for ℓ in $1, \cdots, s$ **do**
  for β_i in $\mathcal{B}_1$ **do**
    $y_{\ell,i} = (U\beta_i)^\ell$.
  end for
  $R_\ell = \texttt{Range}(\{y_{\ell,i}\}_{i=1}^{n/2}, \tau, \varepsilon/s)$.
end for
`# coefficient estimation`
for $\beta_i \in \mathcal{B}_2$ **do**
  $\ell = i \bmod s$.
  $y_{\ell,i} = (U\beta_i)^\ell$.
end for
for j in $1, \cdots, s$ **do**
  $z^j = s \cdot \texttt{Mean}(\{y_{\ell,i} \text{ such that } \ell = j\}, R_j, \varepsilon)$.
end for
Output: $U^{-1} z$.

The following lemma is a modified version of Theorem 2 of Girgis et al. (2022) under pure differential privacy.

Lemma C.4. *Let $\widehat{\beta}^*$ be the true underlying coefficient, and $\widehat{\beta}_i$s be the coefficients estimated by each user. Suppose $\mathbb{E}\left[\widehat{\beta}_i\right] = \beta^*$ and $\|\widehat{\beta}_i - \beta^*\|_2 \le \tau$ with probability $1 - 1/n^2$ for all i. Then with probability $1 - 1/n^2$, we have*

$$\left\| \frac{2}{n} \sum_{i=n/2+1}^{n} \widehat{\beta}_i - \texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2}, \{\widehat{\beta}_i\}_{i=n/2+1}^{n}, \tau, \varepsilon) \right\|_2^2 \lesssim \frac{s\tau^2 \log^2 n}{n\varepsilon^2}$$

Proof of Lemma C.4. We know the $\widehat{\beta}_i$s satisfy Definition 2 in Girgis et al. (2022) with parameter $(\tau, 1/n^2)$. By Levy et al. (2021), we have

$$\|U\widehat{\beta}_i - U\widehat{\beta}^*\|_\infty \lesssim \sqrt{\frac{\tau^2 \log sn^2}{s}}.$$

If we choose $\tau' \asymp \sqrt{\tau^2 \log sn^2/s} \asymp \sqrt{\tau^2 \log n/s}$, then y_i satisfy Definition 2 in Girgis et al. (2022) with parameter $(\tau', 1/n^2)$. Then the Lemma 1 of Girgis et al. (2022) implies that $\Pi_{[a,b]} y_i = y_i$ with probability $1 - 1/n^2$ in Algorithm 8. Then the least square error for `Mean` is

$$\left| \frac{2s}{n} \sum_{i=1}^{n/2s} \tilde{y}_i - \frac{2s}{n} \sum_{i=1}^{n/2s} y_i \right|^2 = \left| \frac{|b-a|s}{n\varepsilon} \sum_{i=1}^{n/2s} \gamma_i \right| \le \sqrt{\frac{288 s\tau'^2 \log n}{n\varepsilon^2}}$$

where the inequality follows from (2.18) in Wainwright (2019). Since $\|\cdot\|_2$ is upper bounded by $\sqrt{s}$ times infinity norm, there holds

$$\begin{aligned}&\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2\\=&\left\|\frac{2}{n}U\sum_{i=n/2+1}^{n}\widehat{\beta}_i-z\right\|_2^2\le s\cdot\left\|\frac{2}{n}U\sum_{i=n/2+1}^{n}\widehat{\beta}_i-z\right\|_\infty^2\le\frac{288s^2\tau'^2\log n}{n\varepsilon^2}\lesssim\frac{s\tau^2\log^2 n}{n\varepsilon^2}.\end{aligned}$$

□

The following lemma is the key technical result to prove Theorem 3.6.

Lemma C.5 (**Privacy and utility of Algorithm 9**). *Let $\widehat{\beta}^*$ be the true underlying coefficient, and $\widehat{\beta}_i$s be the coefficients estimated by each user. Then the algorithm 9 is ε-ULDP. Moreover, there exists some $\tau\asymp\sqrt{\log^2 n/m}$ such that, with probability $1-2/n^2$, we have*

$$\left\|\widehat{\beta}^*-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2\lesssim\frac{s^2\log^3 n}{nm\varepsilon^2}+\frac{s\log n}{nm}$$

Proof of Lemma C.5. We first show the privacy property of Algorithm 9. Since the users of `Range` and `Mean` do not across, it suffices to show that both of the algorithms are ε-ULDP. The privacy of `Range` follows from Lemma 1 of Girgis et al. (2022). The privacy of `Mean` is straightforward by property of Laplace mechanism, given that the sensitivity of $\Pi_{[a,b]}y$ is $|b-a|$. Now we prove the accuracy part. The squared error can be decomposed into two parts associating to private estimation error and non-private estimation error, respectively.

$$\begin{aligned}&\left\|\widehat{\beta}^*-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2\\\le&2\cdot\left(\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2+\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\widehat{\beta}^*\right\|_2^2\right).\end{aligned}$$

We deal with private estimation error part first. From Proposition 3.5, we know the $\widehat{\beta}_i$s satisfy Lemma C.4 with $\tau=\sqrt{s\log n/m}$. Then we have

$$\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2\lesssim\frac{s^2\log^2 n\log n}{nm\varepsilon^2}.\tag{24}$$

If either conditions in Proposition 3.2 holds, the parameter becomes $\tau=\sqrt{s^*\log n/m}$ by Proposition 3.5, and the same analysis goes with s^* instead of s.

$$\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2},\{\widehat{\beta}_i\}_{i=n/2+1}^{n},\tau,\varepsilon)\right\|_2^2\lesssim\frac{ss^*\log^2 n\log n}{nm\varepsilon^2}.\tag{25}$$

Next, we bound the non-private estimation error. When $\widehat{\beta}_i$ is the OLS estimator, by its sub-Gaussianality, we have

$$\left\|\frac{2}{n}\sum_{i=n/2+1}^{n}\widehat{\beta}_i-\widehat{\beta}^*\right\|_2^2\lesssim\frac{s\log n}{nm}.\tag{26}$$

If either conditions in Proposition 3.2 holds, this becomes

$$\left\| \frac{2}{n} \sum_{i=n/2+1}^{n} \widehat{\beta}_i - \widehat{\beta}^* \right\|_2^2 \lesssim \frac{s^* \log n}{nm}. \tag{27}$$

Together, (24) and (26) lead to

$$\left\| \widehat{\beta}^* - \texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2}, \{\widehat{\beta}_i\}_{i=n/2+1}^{n}, \tau, \varepsilon) \right\|_2^2 \lesssim \frac{s^2 \log^3 n}{nm\varepsilon^2} + \frac{s \log n}{nm}.$$

The overall failure probability is at least $2/n^2$ since we utilized two high probability arguments. Similarly, (25) and (27) lead to

$$\left\| \widehat{\beta}^* - \texttt{ULDPMean}(\{\widehat{\beta}_i\}_{i=1}^{n/2}, \{\widehat{\beta}_i\}_{i=n/2+1}^{n}, \tau, \varepsilon) \right\|_2^2 \lesssim \frac{ss^* \log^3 n}{nm\varepsilon^2} + \frac{s^* \log n}{nm}.$$

□

C.2.3. Proof of Theorem 3.6

Proof of Theorem 3.6. By Lemma B.6 and C.5, both `HeavyHitter` and `ULDPMean` are ε-ULDP. Since their associated users do not cross, we have Algorithm 1 is also ε-ULDP. As for (*ii*), by Proposition 3.3, we know that all the non-zero variables of β^* is included in $\{\widehat{v}_1, \cdots, \widehat{v}_s\}$ with probability $1 - 1/n^2$. Thus, we have

$$\|\beta^* - \beta\|_2^2 = \left\| \widehat{\beta}^* - \widehat{\beta} \right\|_2^2.$$

Applying Lemma C.5, this leads to

$$\|\beta^* - \beta\|_2^2 \lesssim \frac{s^2 \log^3 n}{nm\varepsilon^2} + \frac{s \log n}{nm} \lesssim \frac{s^{*2} \log^3 n}{nm\varepsilon^2\alpha^2} + \frac{s^* \log n}{nm\alpha},$$

where in the last step we used Proposition 3.3. In the last, the overall failure probability of Proposition 3.3, Lemma B.6, and Lemma C.5 is at most $4/n^2$. □

D. Extension to Sparse Estimation

The full statement of Theorem 3.7 is as follows. We utilize Algorithm 1 while modifying the estimators $\widehat{\beta}_i$ and selectors $\mathcal{S}_i$ to accommodate the general problem.

Theorem D.1 (Formal version of Theorem 3.7). *Let data $\{X_i\}_{i=1}^n$ be generated by P_{β^*} for $\beta^* \in \Omega_{s,a}^d$. Suppose we have non-private estimators: (i) estimator $\tilde{\beta}_i$ with $\|\tilde{\beta}_i - \beta^*\|_2 \le \nu_1$ for all $1 \le i \le n/2$ and (ii) estimator $\widehat{\beta}_i$ on selected variables with $\mathbb{E}\left[\widehat{\beta}_i\right] = \widehat{\beta}^*$ and $\|\widehat{\beta}_i - \widehat{\beta}^*\|_2 \le \nu_2$ for all $n/2+1 \le i \le n$. Then there exist α-good selectors $\{\mathcal{S}_i\}_{i=1}^{n/2}$ with $\alpha \gtrsim s^*\sqrt{\log n \log d/n\varepsilon^2}$, that is $\Pr(v = \mathcal{S}_i(X_i)) \ge \alpha/s^*$ for $1 \le v \le s^*$ and $1 \le i \le n/2$. Suppose we let $\alpha/8s^* \le \rho \le \alpha/4s^*$, $\tau \asymp \sqrt{\nu^2\alpha \log n/s^*}$. Then, for any $a \gtrsim \nu_1$, Algorithm 1 is ε-ULDP and has an output β with*

$$\|\beta^* - \beta\|_2^2 \lesssim \frac{\nu_2^2}{n} + \frac{\nu_2^2 s^* \log^2 n}{n\varepsilon^2\alpha} \tag{28}$$

with probability at least $1 - 3/n^2$. Moreover, for ℓ_1 norm, there holds

$$\|\beta^* - \beta\|_1 \lesssim \sqrt{\frac{\nu_2^2 s^*}{n\alpha}} + \sqrt{\frac{\nu_2^2 s^{*2} \log^2 n}{n\varepsilon^2\alpha^2}} \tag{29}$$

with probability at least $1 - 3/n^2$.

Proof of Theorem D.1. The privacy guarantee follows from Theorem 3.6. Since $a \gtrsim \nu_1$, we can consistently select all true variables with proxy estimators. This implies we can have α-good selectors with $\alpha \gtrsim 1 \gtrsim s^*\sqrt{\log n \log d/n\varepsilon^2}$. By Proposition 3.3, we know that all the non-zero variables of β^* is selected with probability $1 - 1/n^2$. Thus, we have $\|\beta^* - \beta\|_2^2 = \left\|\widehat{\beta}^* - \widehat{\beta}\right\|_2^2$. Applying Lemma C.4, we have

$$\left\|\widehat{\beta}^* - \widehat{\beta}\right\|_2^2 \lesssim \left\|\widehat{\beta}^* - \frac{2}{n}\sum_{i=n/2+1}^{n} \hat{\beta}_i\right\|_2^2 + \frac{s\nu_2^2 \log^2 n}{n\varepsilon^2}.$$

Since $\widehat{\beta}_i$ are concentrated, it is sub-Gaussian. Thus, there holds

$$\left\|\widehat{\beta}^* - \widehat{\beta}\right\|_2^2 \lesssim \frac{\nu_2^2}{n} + \frac{s\nu_2^2 \log^2 n}{n\varepsilon^2} \lesssim \frac{\nu_2^2}{n} + \frac{s^*\nu_2^2 \log^2 n}{n\varepsilon^2\alpha}$$

where in the last step we used Proposition 3.3. In the last, the overall failure probability of Proposition 3.3, Lemma B.6, and Lemma C.4 is at most $3/n^2$. This yields (28). For (29), note that there is only $s \lesssim s^*/\alpha$ none zero elements. Using the difference between ℓ_1 and ℓ_2 norms, which is $\sqrt{s}$, yields (29). □

E. Additional Experiment Results

E.1. Implementation Details

For each model, we report the best result over its parameter grids, with the best result determined based on the average result of at least 30 replications. We do not perform any parameter selection (e.g. cross validation or validation set) since they are prohibitive under locally private setting (Ma & Yang, 2024; Ma et al., 2024a) or will cost too much privacy budget (Papernot & Steinke, 2021). The parameter grids size are selected based on running time so that each method costs equal amount of computation. Efficient methods receive a exhaustive parameter grid and can be properly tuned. Computation heavy methods receive a small grid with insensitive parameters set to default.

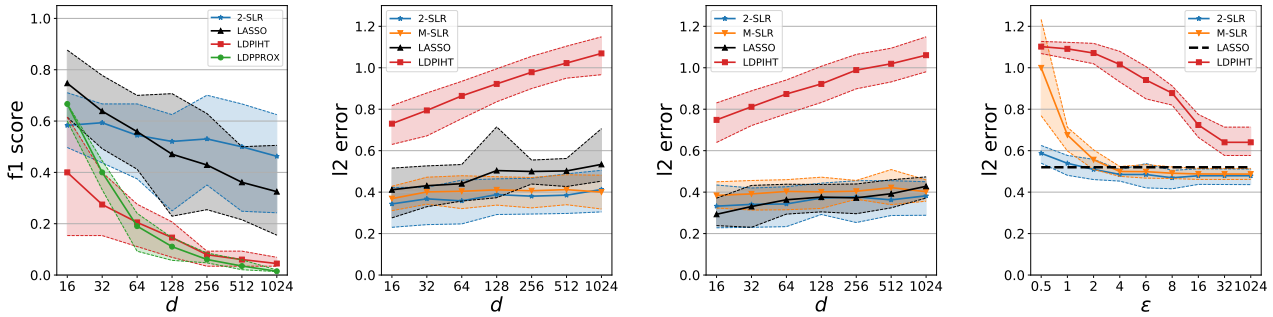
- For candidate variable selector of our methods, we adopt the Lasso estimator and identify its non-zero coefficients as the selected variables. Moreover, we conduct a feature screening (see Appendix B.1 for detail) for acceleration. The number of screened variables is set to 64. The number of selected variables s is selected in $\{2, 4, 8, 16\}$.
  - **2-SLR**: The two-round sparse linear regression protocol is implemented based on Algorithm 1. We select the range $[-B, B]$ in $B \in \{1, 2, 3\}$ and the concentration radius is decided by the number of bins, which is in $\{2, 4, 8, 16, 32\}$.
  - **M-SLR**: The multi-round sparse linear regression protocol is implemented based on Algorithm 6. We set $B = 3$ and select the number of bins in $\{2, 4, 8, 16, 32\}$. Moreover, we set the learning rate of the gradient to be $\eta_t = 0.1 \cdot (\frac{1+t}{2})^{0.2}$.
- **LDPPROX**: The non-interactive locally differentially private sparse linear regressor based on proxy estimator is implemented according to Algorithm 1 in Zhu et al. (2023). Due to the heavy computation burden, we set $r = \sqrt{d \cdot \log n}$, $\tau_1 = 4$, $\tau_2 = 8$. In simulation where we know $\min_{\beta^{*j} \neq 0} |\beta^{*j}| = 0.2$, we set $\lambda = 0.05$. In real data, we set λ to the 10-th lower quantile of the absolute fitted coefficients.
- **LDPIHT**: The locally differentially private iterative hard thresholding is implemented according to Algorithm 2 in Zhu et al. (2023). We select $T \in \{2, 5, 10, 20, 50\}$, $\eta \in \{0.01, 0.1, 1\}$, $\tau_1, \tau_2 \in \{2, 4, 8\}$, $k' \in \{5, 10, 20, 50\}$.
- **Lasso**: The conventional Lasso regressor is fitted using the *LassoCV* class in scikit-learn package (Pedregosa et al., 2011). We set $n_alphas = 300$, $max_iter = 3000$, and $tol = 10^{-4}$.

E.2. Additional Simulation Results

We present the additional result of the correlated marginal distribution data experiment omitted in the main text due to page limitation. The correlation of the first 50 dimensions are set to be exponentially decaying, i.e.

$$\mathrm{Cov}\left(X_{i,j}^k, X_{i,j}^{k'}\right) = 2^{-|k-k'|}.$$

We draw each $\sigma_{i,j}$ correlatedly from a standard Gaussian distribution. For β^*, we randomly select $s^* = 8$ coordinates in the first 50 dimensions to be 0.2 and let others be zero. We typically set $n = 400$, $m = 100$, $d = 256$, and $\varepsilon = 4$, while varying one of them to observe how the evaluated metric varies. We use squared error as evaluation of the estimated coefficient and F1 score as evaluation of the selected variables.



(a) d - F1 score. (b) d - ℓ_2 error with $m = 100$. (c) d - ℓ_2 error with $m = 200$. (d) ε - ℓ_2 error.

Figure 5. Experiments w.r.t. d and ε for correlated marginal. We plot the quantiles over 30 repetitions with 95% coverage. We exclude LDPPROX in some figures since it is highly unstable and do not fit into our plot scale.

We conduct experiments with respect to d. We first analyze the variable selection performance. Due to the high sparsity, we use F1 score as the evaluation criterion. For $d \in \{16, 32, \cdots, 1024\}$, we compute the averaged F1 scores of the proposed candidate variable selection (represented by 2-SLR) and other methods. As depicted in Fig. 5(a), the overall performance of all methods deteriorates compared to that under the independent setting, whereas 2-SLR remains stable and maintains its advantage. When $d = 16$, the selection performances of Lasso and LDPPROX are slightly superior than the variables induced by other methods. However, as d increases, the variable selection performance of Lasso, LDPIHT and LDPPROX decreases sharply, while the F1 scores of 2-SLR only fluctuate slightly and become higher than those of other competitors when $d \geq 64$.

Then, we analyze the estimation performance. In 5(b) and 5(c), we plot the curve of ℓ_2 error with respect to d. Whether $m = 100$ or $m = 200$, the proposed methods are less sensitive to d compared to LDPIHT. The is also compatible with rate in (5) which scales with $\log d$. Compared to the independent case, the correlated case requires more local samples to achieve a consistent selection. Thus, thus difference in results for large m and small m is less apparent.

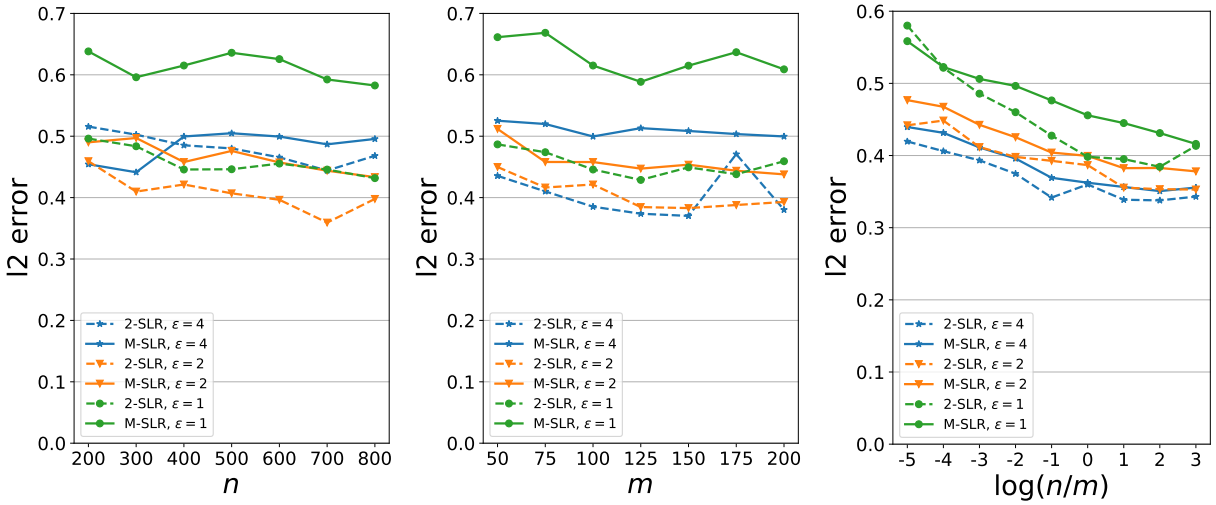
We examine the privacy-utility trade-offs by investigating performances under different εs. In 5(d), the error decreases as ε increases for all private methods as expected. Moreover, the error of 2-SLR is comparable to Lasso, while error of M-SLR quickly drops below Lasso at medium privacy region $\varepsilon \geq 4$. This again ensures the superiority of our methods compared to fitting Lasso using only local information.

Finally, we analyze the impact of sample sizes. In Figure 6(a) and 6(b), the ℓ_2 error decreases as both n and m increases for all ε, which confirms our theoretical claims. The error is generally higher than that in the independent case. The overall ℓ_2 curve is less sensitive to n and m. Moreover, we let $nm = 400 \times 100$ and vary the ratio n/m. In 4(c), we observe that, for each ε, the error of 2-SLR retains for $n/m \approx 1$, while increase slightly when either n or m is too small, which is compatible with Theorem 3.6. The performance of M-SLR is still sensitive to n becoming small.

E.3. Real Dataset Description

A summary of key information for these datasets after pre-processing can be found in Table 3. For user-specific sample partitioning, certain datasets come with predefined partitions, while others undergo random partitioning. Categorical features in the datasets are transformed into dummy variables, while each continuous feature is individually scaled to zero mean and unit variance. We also present additional information of the data sets including the data source and the pre-processing details.

`Airline`: The Airlines-Departure-Delay dataset originally comes from United States Department of Transportation and currently available on OpenML (LeDell, 2020), consists of 1,048,575 observations, including one target variable and 9 attributes pertaining to flight information. We partition samples into users based on the "Destination" variable, selecting 205 users with sample counts ranging from 200 to 400. Attributes such as "Origin" and "UniqueCarrier" are transformed into



(a) n - ℓ_2 error. (b) m - ℓ_2 error. (c) n/m - ℓ_2 error.

Figure 6. Experiments w.r.t. sample sizes for correlated marginal.

Table 3. Information of real datasets.

Dataset	Sample Partition	d	n	m	Area
`Airline`	Predefined	260	205	200-400	Social
`Loan`	Random	735	500	100	Business
`Mip`	Predefined	144	218	5	Computer Science
`Taxi`	Predefined	213	1200	189-200	Social
`Wine`	Random	41	60	100	Business
`Yolanda`	Random	100	800	200	Social

dummy variables, contributing to a total of 260 features in the "Airlines" dataset. Overall, the Airlines dataset contains $75,600$ samples.

`Loan`: The Loan-Default-Prediction dataset is obtained from the training set of the Kaggle Loan Default Prediction challenge (DrivenData, 2021a), which aims to reduce the consumption of economic capital and optimize on the risk to the financial investor. The original dataset comprises 55319 instances of 735 attributes We randomly select $50,000$ samples and partition the data into 500 groups, with each group containing 100 samples.

`Mip`: The MIP-2016-regression dataset, available on OpenML, comprises $1,090$ instances featuring 144 attributes and 1 output attribute (Bergdoll, 2019). Within this dataset, there are a total of 218 users, with each user possessing 5 samples.

`Taxi`: The Taxi dataset is obtained from the Differential Privacy Temporal Map Challenge (DrivenData, 2021b), which aims to develop algorithms that preserve data utility while guaranteeing individual privacy protection. The dataset contains quantitative and categorical information about taxi trips in Chicago, including time, distance, location, payment, and service provider. We partition the samples based on the unique identification number of taxis ($taxi_id$), resulting in 1200 taxis with sample counts ranging from 189 to 200. Other features include the time of each trip ($seconds$), the distance of each trip ($miles$), the time period during which each trip occurs($shift$), index of the zone where the trip starts (pca), index of the zone where the trip ends (dca), service provider ($company$), the method used to pay for the trip ($payment_type$) and amount of tips ($tips$) and fares ($fare$). We use the other variable to predict the fares of the fares ($fare$) of the trips. Attributes such as $shift$, pca,dca, $company$ and $payment_type$ are transformed into dummy variables, resulting in a total of 213 features in the Taxi dataset.

`Wine`: This dataset originates from the Wine Quality dataset (Cortez et al., 2009) on UCI Machine Learning Repository, which combines data from both the "red wine" and "white wine" datasets. The original dataset comprises 11 features associated with wine to predict the corresponding wine quality. In an effort to enhance dimensionality, Gaussian random noise in 30 dimensions has been incorporated. 6000 instances are collected in the dataset. The samples are randomly

partitioned among 60 users, with each user having 100 samples.

`Yolanda`: The Yolanda dataset (Guyon et al., 2019) contains 400000 instances of 100 attributes and 1 output attribute. We randomly select $160,000$ samples and distribute them into 800 groups, with each group containing 200 samples.

E.4. Additional Real Datasets Results

Table 4. Running time(seconds) on real datasets.

Datasets	Lasso	2-SLR	M-SLR	LDPPROX	LDPIHT
`Airline`	0.7	15.8	15.3	766.5	6.1
`Loan`	43.1	74.7	106.9	4124.0	30.6
`MIP`	0.1	0.1	2.3	5.5	4.3
`Taxi`	0.1	0.7	11.5	1569.1	7.0
`Wine`	0.2	0.7	2.9	3.5	2.6
`Yolanda`	0.5	0.4	12.5	365.8	4.4