# Distributed Differentially Private Data Analytics via Secure Sketching

Jakob Burkhardt, Hannah Keller, Claudio Orlandi, Chris Schwiegelshohn
{jakob,hkeller,orlandi,schwiegelshohn}@cs.au.dk
Aarhus University
Denmark

## Abstract

We explore the use of distributed differentially-private computations across multiple servers, balancing the tradeoff between the error introduced by the differentially-private mechanism and the computational efficiency of the resulting distributed algorithm.

We introduce the *linear-transformation model*, where clients have access to a trusted platform capable of applying a public matrix to their inputs. Such computations can be securely distributed across multiple servers using simple and efficient secure multiparty computation techniques.

The linear-transformation model serves as an intermediate model between the highly expressive *central model* and the minimal *local model*. In the central model, clients have access to a trusted platform capable of applying any function to their inputs. However, this expressiveness comes at a cost, as it is often expensive to distribute such computations, leading to the central model typically being implemented by a single trusted server. In contrast, the local model assumes no trusted platform, which forces clients to add significant noise to their data. The linear-transformation model avoids the single point of failure for privacy present in the central model, while also mitigating the high noise required in the local model.

We demonstrate that linear transformations are very useful for differential privacy, allowing for the computation of linear sketches of input data. These sketches largely preserve utility for tasks such as private low-rank approximation and private ridge regression, while introducing only minimal error, critically independent of the number of clients. Previously, such accuracy had only been achieved in the more expressive central model.

## 1 Introduction

Differential Privacy (DP) [DMNS06] has become the de-facto standard for ensuring the privacy of individuals whose data is used in data analytics. DP offers strong, provable guarantees, such as composability and resilience to auxiliary information. In the classic *central model* of differential privacy, clients submit their data to a trusted central server, which processes the data and releases a (necessarily) noisy version of the result. Analysts can then use this result to perform queries on the data. Informally, differential privacy guarantees that the analyst cannot confidently determine whether any individual contributed their data, except with some small probability.

The central model requires trust that the server will not reveal any individual's data and cannot be compromised by external actors. However, universally trusted servers may not exist in practice. In situations where privacy is critical but trusting a single server is not feasible, alternative approaches are available: One such approach is to use secure multiparty computation (MPC) techniques to distribute the central server's computations across multiple servers. Under the assumption that a subset of these servers are honest, this setup guarantees the same level of privacy as the central model. However, MPC techniques often introduce significant computational overhead, especially when distributing complex or expressive computations.

Another option is to avoid relying on external servers altogether. In this case, each client must publish their data in a locally differentially private manner, by adding noise directly to their data. While this removes the need for a trusted server, it results in lower utility due to the increased noise. This approach is known as the *local model* of differential privacy [KLN$^+$08], an example of which is randomized response [War65].

Both the central and local models of differential privacy have strengths and drawbacks. The central model typically yields higher utility but requires strong trust in a single server or the costly use of MPC. The local model, by contrast, requires no trust in external parties but suffers from reduced utility. These tradeoffs have motivated significant research into finding a middle ground between these two models. One promising direction is to limit the expressiveness of the class of functions $\mathcal{F}$ that can be executed in a trusted manner. While this restriction allows for more efficient distributed implementation, it is only useful if the DP mechanism built around such functions can still achieve good accuracy.

A notable example of such restricted expressiveness is the class of functions known as *shuffles*, where a central entity (or a distributed system) randomly permutes the messages from clients before they reach the analysing server. This is known as the *shuffle model* [BEM$^+$17, EFM$^+$19, CSU$^+$19]. The shuffle model is motivated by cryptographic protocols (e.g., mix-nets), which allow for securely distributing the shuffle across multiple servers. The shuffle model has been used to implement many differentially-private mechanisms with reasonable utility [EFM$^+$19, CSU$^+$19]. Surprisingly, the shuffle model is more expressive than initially expected: with a trusted shuffler, any function can be securely computed in two rounds under the assumption of an honest-majority [BHNS20].

We therefore ask the following:

**Question 1.1.** *What is the least expressive class of functions $\mathcal{F}$ that needs to be securely implemented in order to achieve computationally efficient distributed differential privacy with utility comparable to that of the central model?*

### 1.1 Our Contribution

In this work, we investigate the power of the *linear-transformation model (LTM)* for differentially-private mechanisms. In this model, the clients only have access to a trusted platform for performing arbitrary linear transformations of their inputs. Linear functions can be distributed extremely efficiently with secure computations. However, it is also known that the expressiveness of this class of functions is strictly limited.

For a visualization of the architecture, we refer to Figure 1. We demonstrate the benefits of this linear transformation model (LTM) by showing that it can be used, in combinations with *linear sketches* to construct private summaries of a dataset with noise comparable to that of central differentially private mechanisms.

As a warm-up, we show how to use linear sketches to compute vector statistics as described in Section 3.1. For more advanced application, we consider problems from linear algebra such as *low rank approximation* and *regularized linear regression.* Note that in the LTM no noise is added centrally, and therefore clients (like in the local model) must add some noise to their data before sending it to the trusted platform. However, we show that thanks to the centrally applied linear transformation, we can significantly reduce the noise to be added (and therefore obtain better utility). Although it is not inherent to the model, all of our mechanisms require only a single round of communication from clients to server, akin to streaming algorithms.

In a nutshell, we obtain utility guarantees similar to the central model, while using significantly weaker assumptions that the central or shuffle model. A full overview of related work is given in Table 1.

*Low Rank Approximation.* We are given a data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$. Our goal is to compute an orthogonal projection $\mathbf{X} \in \mathbb{R}^{d \times r}$ minimizing the error $OPT = \min_{\mathbf{X}} \|\mathbf{A} - \mathbf{A}\mathbf{X}\mathbf{X}^T\|_F^2$, where $\|.\|_F$ denotes the Frobenius norm of a matrix. In the LTM, we can compute an $\hat{\mathbf{X}}$ differentially privately with error $\|A - \hat{X}\hat{X}^T A\|_F^2 \leq (1 + o(1)) \cdot (\|A - XX^T A\|_F^2 + \text{poly}(\varepsilon^{-1}, d, k, \log 1/\delta)$. Notably, this bound is independent of $n$, which so far has only been achieved in the more expressive central model.

*Ridge Regression.* We are given a data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, a target vector $\mathbf{b} \in \mathbb{R}^n$ and regularization parameter $\lambda > 0$. Our goal is to find $\mathbf{x} \in \mathbb{R}^d$ minimizing $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda \cdot \|\mathbf{x}\|^2$, where $\|.\|$ denotes the Euclidean norm of a vector. Each row of the data matrix, as well as the corresponding entry of $\mathbf{b}$ resides with a client. Assume that $\lambda \geq \text{poly}(\varepsilon^{-1}, d, \log 1/\delta)$ In the LTM, we can compute an $\hat{\mathbf{x}}$ differentially privately with error $(1 + o(1)) \cdot (\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda \cdot \|\mathbf{x}\|^2) + \text{poly}(\varepsilon^{-1}, d, \log 1/\delta)$. We note that this result is also, to the best of our knowledge, the first differentially private mechanism that yields a finite bound on the additive error even in the central model without making assumptions on $\mathbf{A}$ and $\mathbf{b}$.

*Comparison to state-of-the-art.* Table 1 provides a comparison of the utility bounds for the problems in the local, central, shuffle and LTM model. We obtain similar utility to mechanisms run in the more expressive central model, while at the same time we are able to distribute the evaluation of the mechanism and avoid the single point of failure of the central model. When comparing with the similar shuffle model for frequency estimation, it should be noted that the LTM is a strictly less expressive model. Indeed, it is possible to use the shuffle model to emulate the LTM, while the converse is not possible (remember that the matrix is public in the LTM, while the permutation is secret in the shuffle model). For low rank approximation/PCA, we are not aware of any work in the shuffle model. All of our mechanisms require only a single round of communication between users and servers, with one messages per user sent to each server in that round.

## 1.2 Related Work

There is substantial work on the shuffle model, which also aims at facilitating differential privacy in a distributed setting. We discuss the relationship between the LTM and the shuffle model in more detail in Section 3. For the problems we studied here, there is an abundance of prior work discussed as follows.

*Low Rank Approximation:* There are various ways in which one could formulate the low rank approximation problem. The setting which is most important to us is the seminal paper [DNMR14], who achieved a worst case additive error of the order $d\sqrt{k}$ for outputting an orthogonal projection matrix $V_k$ in the row space of $\mathbf{A}$. This bound is also optimal. In the local model where each client holds a row of the data matrix $\mathbf{A}$, [Upa18] gave an algorithm with an additive error of the order $\sqrt{n}$, which matches the lower bound by [BS15] up to lower order terms. To the best of our knowledge, no single round shuffle protocol improving over the local bounds is known.

Low rank approximation and its sister problem PCA, have seen substantial attention in settings more loosely related to our work. Much of the work on private low rank approximation [BDMN05, CSS12, DTTZ14, HR12, HR13, HP14, BDWY16, KT] considers the data to be fixed, often using Gaussian noise. In addition, a strong spectral gap assumption, typically of the form that the first singular value is substantially larger than the second, is crucial to the analysis. We note that all algorithms operating with this assumption do not yield worst case bounds.

When there is no spectral gap, we compare to the central model exponential mechanism approach from [CSS12], implemented by [KT]. [LKO22] also provide a solution to this problem; however, it is unfortunately computationally intractable.

*Ridge Regression.* The previous work most related to ours is due to [She19]. The author gave a mechanism that preserved the entire spectrum of a data matrix in a private manner and also showed that the returned regression vector $\hat{\mathbf{x}}$ is close to the optimal $\mathbf{x}$ assuming that the matrix is well conditioned.

Most previous work [KST12, CWZ21, BST14, WFS15, WGX18, WSX18, WHZ$^+$23, Wan18, VTJ22, MKFI22, LKO22, ZMW17] study private linear regression in the context of risk minimization, where the algorithm is given i.i.d. samples from some unknown distribution and aims at computing a solution with good out of sample performance. Even without privacy constraints risk minimization yields an additive error that depends on $n$. Another line of work studies varies loss functions, including regression in a Bayesian setting [MASN16, FGWC16, DNMR14]. We are not aware of any prior work on linear regression in the local model of differential privacy.

*Differential Privacy via Johnson-Lindenstrauss Transforms.* The Johnson-Lindenstrauss lemma has been previously used in differential privacy. [BBDS12] studies the DP guarantee a JL transform itself gives in the central model, in the context of cut queries and directional variance queries. [KKMM13] and [Sta21a] use JL transforms together with Gaussian noise in the context of differentially private Euclidean distance approximation, to decrease the error. This was subsequently improved in [Sta21b], using Laplace noise. The difference to our work is that the transform is used to reduce $d$ instead of $n$, by having every client apply it locally before adding Gaussian noise. In [Nik23] the authors make use of JL transforms to achieve private query release, by applying it before releasing a bundle of queries. [GKK+23] studies pairwise statistics in the local model of differential privacy and uses JL transforms to reduce client-sided dimensionality.

## 2 Preliminaries

*Notation.* Column vectors are written in bold lowercase letters $\mathbf{b}$ and matrices in bold uppercase letters $\mathbf{A}$. The transpose operator over vectors and matrices is $\mathbf{b}^T$ and $\mathbf{A}^T$. For any vector $\mathbf{b}$, we denote by $\|\mathbf{b}\| = \sqrt{\sum_i b_i^2}$ its $\ell_2$-norm. For any matrix $\mathbf{A}$, we denote by $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j \mathbf{A}_{i,j}^2}$ its Frobenius norm. We denote the inner product between vectors $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b}$. Two vectors $\mathbf{a}, \mathbf{b}$ are orthogonal if their inner product is 0. An orthogonal matrix $\mathbf{A}$ is a real matrix where the columns have unit Euclidean norm and are pairwise orthogonal.

*Subspace Preserving Sketches.* We consider subspace-preserving sketches in this work. Of particular interest for us are algorithms that compute a subspace approximation of $A$ without prior knowledge of $A$. Such algorithms are known as oblivious subspace embeddings. We will use a family of such embeddings known as *oblivious sparse norm-approximating projections* originally due to [NN13], but since improved upon in several subsequent works.

**Definition 2.1** (OSNAP [NN13].). Let $\mathcal{D}^{\mathsf{sketch}}_{m,n,s}$ be a distribution over random matrices $\{-1, 0, 1\}^{m \times n}$, where $s$ entries of each column, chosen uniformly and independently for each column, are set to $\pm 1$ with equal probability, and all other entries are set to 0. If $s = 1$, we write $\mathcal{D}^{\mathsf{sketch}}_{m,n}$. We say that $S \sim \mathcal{D}^{\mathsf{sketch}}_{m,n,s}$ is an $(\alpha, \beta, s, m)$ OSNAP for a $d$-dimensional subspace $W \in \mathbb{R}^n$ if with probability $1 - \beta$, for all $a \in W$ and some precision parameter $\alpha$

$$\left| \|Sa\|_2^2 - \|a\|_2^2 \right| \leq \alpha \cdot \|a\|^2.$$

We give bounds on available choices of $(\alpha, \beta, s, m)$ for subspaces of rank $k$ in table 5 in the appendix.

*Differential Privacy.* Differential privacy [DMNS06] offers privacy guarantees to individuals contributing their data to some randomized algorithm. We say that two datasets are neighboring if one can be obtained from the other by the the replacement of a single individual with another individual.

**Definition 2.2** (Differential Privacy in the central model [DMNS06]). Let $\varepsilon \geq 0$ and $\delta \in [0, 1]$. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is $(\varepsilon, \delta)$ differentially private, if for all neighboring data sets $x, x' \in \mathcal{X}$ and all outputs $S \subseteq \mathcal{Y}$ it holds that

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta,$$

where the probabilities are over the randomness of $\mathcal{M}$.

**Lemma 2.3** (The Gaussian Mechanism [DR14]). *Let $f : \mathcal{X} \to \mathbb{R}^k$ be a function and let $\varepsilon \geq 0$ and $\delta \in [0, 1]$. The Gaussian mechanism adds to each of the $k$ components of the output, noise sampled from $N(0, \sigma^2)$ with*

$$\sigma^2 \geq \frac{2(\Delta_2 f)^2 \ln(1.25/\delta)}{\varepsilon^2},$$

*where $\Delta_2 f = \max_{x \sim x'} \|f(x) - f(x')\|_2$ denotes the $\ell_2$ sensitivity of function $f$. The Gaussian mechanism is $(\varepsilon, \delta)$ differentially private.*

## 3 Privacy Guarantees in the LTM

We consider a model in which the trusted component can perform any public linear transformation of the inputs. The model includes three algorithms: (1) $R : \mathcal{X} \to \mathcal{Y}$ is a randomized encoder that takes a single user's data and outputs a randomized message, (2) $T : \mathcal{Y}^* \to \mathcal{Y}^*$ is the idealized trusted component that performs a public linear transformation of its inputs, and (3) $A : \mathcal{Y}^* \to \mathcal{Z}$ is an analysis function that takes the results messages and estimates some function from these messages.

Note that standard definitions of differential privacy in the shuffle model only require $T(R(x_1), ..., R(x_n))$ to be differentially private. This implicitly assumes that all clients are honest and do not collude with the adversary, in particular they are assumed not to leak the output of their randomizers publicly. This implies that existing definitions of differential privacy in the shuffle model could be satisfied even by (artificial) mechanisms in which a single client adds the whole noise while the others do not randomize their messages at all. This is clearly a very weak privacy guarantee: in a setting in which a large number of clients participate in a differentially private data analysis, it is unrealistic to assume that the adversary does not control even a single client. Luckily, to the best of our knowledge, no proposed mechanisms in the literature suffer from these vulnerabilities, still this counterexample shows that the existing definition is not robust enough, and we therefore formalize a notion of differential privacy where we explicitly tolerate that a bounded number of clients might collude with the adversary. Similar observations were also made by [TWM+23].

**Definition 3.1** (Trusted Computation Model for Differential Privacy). A tuple of algorithms $P = (R, T, A)$ is $(\varepsilon, \delta)$-differentially private given corrupt clients $C_{cor}$ if the output $\Pi_R(x_1, ..., x_n) = T(R(x_1), ..., R(x_n))$, as well as corrupted parties' randomizer output $R(x_i)$ for all $x_i \in C_{cor}$ satisfy $(\varepsilon, \delta)$-differential privacy.

This definition requires any possible subset of honest parties to jointly add enough randomness in $R$ to preserve the privacy guarantee, thus guaranteeing privacy even against a semi-honest adversary that is able to learn the outputs of an arbitrary subset of clients.

*Multi-Central Model of Differential Privacy.* [Ste20] introduce the multi-central model, which exactly defines this split-trust model that instantiates the trusted computation model above. They allow a semi-honest adversary that honestly follows the protocol to corrupt up to all but one servers and all but one client. We also operate in the same semi-honest setting, but define parameters $t$ and $t'$ for the number of tolerated server and client corruptions.

The view $\mathtt{View}^{R,\Pi,A}_{C_{cor},S_{cor}}(x)$ of the adversary consists of all the information available to the corrupted clients $C_{cor}$ and servers $S_{cor}$

| | Local | Shuffle | LTM (**this work**) | Central |
|---|---|---|---|---|
| Frequency Est | $\tilde{O}(\sqrt{n})$ [BNST20]<br>$\Omega(\sqrt{n})$ [BS15] | $\tilde{\Theta}(\varepsilon^{-1})$[GGK+21] | $\tilde{O}(\varepsilon^{-2})$ | $\tilde{\Theta}(\varepsilon^{-1})$ |
| Low-Rank Approximation<br>PCA | $\tilde{O}(\sqrt{n})$[Upa18]<br>$\Omega(\sqrt{n})$ [BS15] | N.A. | $\tilde{O}(d^3k^2\varepsilon^{-2}\log\frac{d}{\delta})$ | $\tilde{\Theta}(d\sqrt{k}\varepsilon^{-2})$[DTTZ14] |

**Table 1: SOTA utility bounds for various models for frequency estimation and low rank approximation.**

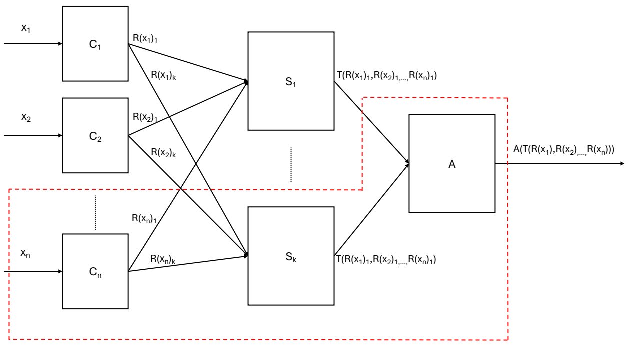
x1
C1
R(x1)1
R(x1)k
x2
C2
R(x2)1
R(x2)k
xn
Cn
R(xn)1
R(xn)k
S1
Sk
T(R(x1)1,R(x2)1,...,R(xn)1)
T(R(x1)1,R(x2)1,...,R(xn)1)
A
A(T(R(x1),R(x2),...,R(xn)))

**Figure 1: Adversary's View**

plus the final output of the honest, uncorrupted servers. This view excludes only internal information of the trustworthy servers and clients. The resulting protocol is $(\varepsilon, \delta)$-differentially private in the multi-central model if adversary's view is $(\varepsilon, \delta)$-indistinguishable from the output on a neighboring dataset.

Figure 1 further illustrates what is contained in the adversary's view. All clients randomize their input and send a single message to each server, who locally perform a linear transformation and release the result to some analyzer. If one server and one client are corrupted, the adversary's view contains all incoming and outgoing messages from that server and that client, or the incoming and outgoing edges from the client, server, and analyzer marked in red in the figure.

**Definition 3.2** (Instantiation of Trusted Computation Model for Differential Privacy with MPC)**.** Let $\Pi$ be a $k$-party MPC protocol that computes $f : \mathbb{R}^n \to \mathbb{R}$ with perfect security, tolerating $t$ corruptions. A tuple of algorithms $P = (R, \Pi, A)$ is $(\varepsilon, \delta)$-differentially private if for all coalitions $S_{cor}$ of up to $t < k$ corrupt servers and for all coalitions $C_{cor}$ of $t' < n$ corrupt clients and all neighboring datasets $x$ and $x'$:

$$\Pr[\mathtt{View}^{R,\Pi,A}_{S_{cor},C_{cor}}(x) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathtt{View}^{R,\Pi,A}_{S_{cor},C_{cor}}(x') \in S] + \delta$$

where the probability is over all the randomness in the algorithms $(R, \Pi, A)$.

With this definition, we can give our first result showing that differential privacy is retained given a bounded number of corruptions. The proof can be found in Appendix B.

**Lemma 3.3.** *If $P = (R, T, A)$ is $(\varepsilon, \delta)$-differentially private with $t'$ corrupt clients $C_{cor}$, and if $\Pi$ is a perfectly secure $k$-party MPC protocol that computes $T$ correctly while tolerating $t$ corrupt servers $S_{cor}$, then $P = (R, \Pi, A)$ is $(\varepsilon, \delta)$-differentially private as long as there are less than $t'$ corrupt clients and $t$ corrupt servers.*

## 3.1 Frequency Moments

To illustrate the possibilities inherent to the LTM, we consider a simple application by way of estimating frequency moments. For simplicity, assume in this section that all clients are trustworthy, i.e. $t' = 0$, but the arguments can be straightforwardly extended to deal with arbitrary values of $t'$. Here, each client is given a number $[-\Delta, \Delta]$ and our goal is to estimate $F_k = \sum_{i=1}^n |x_i|^k$. In the local model, even if all entries are either 0 or 1, it is not possible to estimate $F_1$ without incurring an additive error of the order $\sqrt{n}$, that is the estimated value $\tilde{F_1} = F_1 \pm O(\sqrt{n})$ for any privacy preserving local mechanism [BS15]. In contrast, simply computing $F_k$ and adding an appropriate amount of noise[1] yields an additive error that only depends on $\Delta$, $\varepsilon$ and $\delta$.

We now consider the LTM. Let $\mathbf{x}^k$ denote the vector of client entries with $\mathbf{x}_i^k = |x_i^k|$. We observe that $F_k = 1^T\mathbf{x}^k$. Suppose every client $i$ samples a Gaussian random variable $g_i$ with zero mean and variance $\sigma^2 \geq \frac{2\Delta^k \ln(1.25/\delta)}{n \cdot \varepsilon^2}$ and adds it to $|x_i^k|$. Then we have $\tilde{F_k} = 1^T(\mathbf{x}^k + g) = F_k + \sum_{i=1}^n g_i$. We now observe that $\sum_{i=1}^n g_i$ is Gaussian distributed with zero mean and variance $n \cdot \sigma^2 \geq \frac{2\Delta^k \ln(1.25/\delta)}{\varepsilon^2}$. Thus, assuming a trusted computation of $1^T g$, $\tilde{F_k}$ is a differentially private estimate of $F_k$ with error $\frac{2\Delta^k \ln(1.25/\delta)}{\varepsilon^2}$.

A key property used in this example (and throughout the paper), is that sums of Gaussian random variables are Gaussian distributed. A similar property also holds for Cauchy random variables, or random variables drawn from any stable distribution. Thus, differential privacy protocols based on the Gaussian or the Cauchy mechanism [NRS07] are natural baseline candidates for the LTM. Nevertheless, it is also possible to add noise following a distribution where the sum of random variables can be controlled. For example, the sum of negative binomial random variables follows a geometric distribution, which likewise can be used in differential private mechanisms [GKMP20]. Exploring these options in the context of the LTM is a interesting open problem.

## 3.2 Privacy Preserving Mechanism for Oblivious Sparse Johnson Lindenstrauss Transforms

We instantiate the trusted computation model for differential privacy using MPC, where the function $T$ performs a matrix multiplication and outputs a sketch of the input data. More specifically, we define the tuple of algorithms $(R_\sigma, T_S, A)$. $R_\sigma(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^{ds}$

[1] Using the Gaussian mechanism, $\sigma^2 = \rho \cdot \frac{\Delta^k}{\varepsilon^2} \log \delta^{-1}$ for some absolute constant $\rho$ is sufficient, though other mechanism can yield even better bounds.

**Algorithm 1** Linear Transformation Model

```
Input: Individual input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ representing values in $\mathbb{R}^d$

Parameters: noise variance $\sigma^2$, public sketching matrix $\mathrm{S} \in \mathbb{R}^{m \times n}$
Each client $i \in [n]$ then locally computes $R_\sigma(\mathbf{x}_i)$ on their input $\mathbf{x}_i$.
Each client secret shares the resulting value using a linear secret sharing scheme, and sends one share to each server.
The servers jointly compute function $T_{\mathrm{S}}$ on the resulting matrix, which results from concatenating all $n$ vector secret shares.
The resulting matrix product is be published and then taken as input to any analysis function $A$.
Output: Output $A(T_{\mathrm{S}}(R_\sigma(\mathbf{x}_1, \ldots, \mathbf{x}_n))$
```

sets the input value to $0^d$ if $n < 8m \ln(dm/\delta) + t'$. Otherwise, for all $i \in [s]$, sample $\mathbf{g}_i \leftarrow \mathcal{N}(0, \sigma^2)^d$ with variance $\sigma^2$, and add these sample vectors to the input vector, resulting in $s$ vectors $\mathbf{x}_i = \mathbf{x} + \mathbf{g}_i$. Each client then secret shares the resulting values using a linear secret sharing scheme and sends one share to each server. $T_{\mathrm{S}} : \mathbb{R}^{n \times ds} \to \mathbb{R}^{m \times d}$ has as a parameter sketching matrix $\mathrm{S} \leftarrow \mathcal{D}^{\mathsf{sketch}}_{m,n,s}$ and takes as input matrices $\mathbf{A}_i \in \mathbb{R}^{n \times d}$ for $i \in [s]$. Notice that S satisfying Definition 2.1 can be decomposed such that $\mathrm{S} = \frac{1}{\sqrt{s}} \sum_i^s \mathrm{S}_i$, where $\mathrm{S}_i \in \{-1, 0, 1\}^{m \times n}$ is a matrix with only one non-zero entry per column. $T_{\mathrm{S}}$ outputs the sum of matrix products $\frac{1}{\sqrt{s}} \sum_{i \in [s]} \mathrm{S}_i \mathbf{A}_i$. $A$ takes a matrix in $\mathbb{R}^{m \times d}$, as well as any public values, and outputs some data analysis.

Algorithm 1 describes how clients and servers jointly compute the linear transformation, allowing the transformed data to be published and used for analysis. We now evaluate the variance necessary to guarantee differential privacy in this model. We find that for sufficiently large $n$, we can apply the stability of Gaussian distributions to divide by the number $(n - t')/2m$ of honest clients expected to contribute to any entry of a column in the resulting noisy matrix product. The concentration bound, formulated in Lemma B.1 and used to bound this number of honest clients, contributes to the necessary value of $\delta$. Composition theorems can be applied on the columns of the resulting matrix product to yield the final differential privacy guarantee. We will later see that adding noise with this variance to data results in only a small multiplicative error in the resulting error for regularized linear regression and low rank approximation.

We are now ready to state our privacy guarantees in the LTM based on the Gaussian mechanism.

**Theorem 3.4.** *Let $\varepsilon \geq 0$, $\delta \in (0, 1)$, $t' < n$, $m \in [n]$, $s \in [n]$ and*

$$\sigma^2 = \frac{4s^2\eta^2 \ln(1.25s/(\delta/d - m\exp(\frac{-(n-s-t')}{8m})))m^2d^2}{\varepsilon^2(n - s - t')}$$

*Let $\mathrm{S} \sim \mathcal{D}^{\mathsf{sketch}}_{m,n,s}$ with $t$ non-zero entries per column. Then as long as input values are bounded above by $\eta$, tuple of algorithms $(R_\sigma, T_{\mathrm{S}}, A)$ is $(\varepsilon, \delta)$-differentially private in the trusted computation model for differential privacy with $t'$ corrupt clients $C'$.*

The proofs can be found in Appendix B. Corollary 3.5 below follows directly from Theorem 3.4 and Lemma 3.3.

**Corollary 3.5.** *Let $(R_\sigma, T_{\mathrm{S}}, A)$ be the tuple of algorithms above. If $T_{\mathrm{S}}$ is computed correctly using an MPC protocol $\Pi_{\mathrm{S}}$ with perfect security tolerating $t$ semi-honest corruptions $S_{cor}$, then the tuple of algorithms $(R_\sigma, \Pi_{\mathrm{S}}, A)$ is $(\varepsilon, \delta)$-differentially private with $t'$ corrupt clients $C_{cor}$.*

## 3.3 Cryptographic Assumptions and Relations to the Shuffle Model

Crucially, the execution of the linear transformation can be easily and securely distributed using simple cryptographic techniques for *secure multiparty computation* (MPC) such as *linear secret-sharing* (LSS). The guarantee offered by MPC is that no adversary corrupting all but one of the servers can learn more than the output of the computation performed. [DKM+06] is the first work to consider the combination of differential privacy and MPC; their focus is distributed noise generation, which requires interaction between servers. [CY23] provide lower bounds for non-interactive multi-server mechanisms. [DKN+23] provide an interactive MPC protocol to compute selection for distributed trust models of differential privacy. The linear transforms can be non-interactively computed locally by servers and are also known to all participating parties, that is, once initiated, the output of the protocol is fully deterministic. This allows to implement the LTM with a set of central servers under the minimal assumption that at least one honest server will not collude with all the others, without any requirement to fully trust any particular server. Using only simple LSS has several important consequences for both the efficiency and the security of the overall system: each client only needs to send a single message to each server, and servers do not need to communicate with each other, but only send a single message to the data analyst. Thus, the total workload is essentially that of the central model times the number of servers used. Finally, as LSS can be instantiated with information-theoretic security, the security of our system does not depend on any unproven mathematical assumption and immediately offers security even against powerful quantum adversaries. We describe the type of secret sharing scheme that can be used to securely compute linear transformations using MPC in Appendix E.

Secure aggregation and the shuffle model are two instantiations of intermediate trust models for differential privacy. Secure aggregation [GX17, BIK+17, MPBB19, AG21, TWM+23] is a special case of the LTM, where the linear transformation applied is a sum, which is useful in federated learning. The shuffle model [BEM+17, EFM+19, CSU+19] is motivated in practice by the existence of cryptographic protocols (e.g., mixnets) which allow to securely distribute a shuffle among a set of servers. However, implementing shuffles using cryptography is significantly more cumbersome than implementing linear transformations. In particular mixnets require: 1. servers to talk to each other over a chain, thus increasing the latency of the system (while in the LTM all servers can perform the computation in parallel and without talking to each other); 2. the use of computationally intensive public-key cryptography (while LSS only requires performing linear operations); 3. to use computational assumptions, (while LSS provides unconditional security). Finally, shuffles/mixnets are inherently randomized algorithms, while linear transformations are deterministic functions and as such are simpler to implement securely.[2]

## 4 Numerical Linear Algebra in LTM

In this section we give the utility guarantees of our mechanism. We start with giving generic distortion bounds relating the spectrum of the noisy, but private matrix to the the spectrum of the original non-private matrix. Parameterizations of this mechanism depend on the underlying class of subspace embeddings. In this utility analysis, we limit outselves to OSNAPs with $m$ and $s$ chosen according to [Coh16]; other trade-offs are possible and can be found in Appendix A. Applications to specific problems such as regression and low rank approximation are given towards the end of the section.

Here we present the formal utility statements for ridge regression and low rank approximation. All bounds have a dependency on $\sigma^2 n$. This term is in $\mathrm{poly}(d\varepsilon^{-1}\log(1/\delta))$ by our parametrization of the Gaussian mechanism. All proofs can be found in the Appendix C.

We will use the following spectral bounds for both linear regression as well as low rank approximation. We believe that there may be further applications and that the bounds themselves are therefore of independent interest.

**Lemma 4.1.** *Let $S \in \frac{1}{\sqrt{s}} \cdot \{-1,0,1\}^{m\times n}$ be an $(\alpha,\beta,m,s)$ OSNAP with $S = \frac{1}{\sqrt{s}}\sum_{i\in[s]} S_i$, where $S_i \in \{-1,0,1\}^{n\times m}$ has exactly one non-zero entry per row. Let $G = \sum_{i\in[s]} S_i G_i$ where every matrix $G_i \in \mathbb{R}^{n\times d}$ has independent Gaussian entries $\mathcal{N}(0,\sigma^2)$. Further, let $V$ be a set of $d$-dimensional vectors lying in a $k$-dimensional subspace. Then with probability at least $1-\beta$ for some absolute constant $\eta$*

$$\sup_{\mathbf{x}\in V} \frac{1}{s}\|\mathbf{G}\mathbf{x}\|_2^2 \leq \eta \cdot 2\frac{n}{m}\sigma^2 \cdot \|\mathbf{x}\|^2 \cdot (\sqrt{(k+\log 1/\beta)\cdot m} + (k+\log 1/\beta)).$$

*and*

$$\frac{1}{s}\|\mathbf{G}\|_F^2 \leq \eta \cdot 2n\sigma^2 \cdot d \cdot \log 1/\beta$$

We first begin with our utility results for low rank approximation.

**Theorem 4.2.** *Let $\varepsilon \geq 0$, $\delta \in (0,1)$, $t' < n$ and $\sigma$ be chosen as described in Theorem 3.4. Let $\mathbf{S} \sim \mathcal{D}^{\mathsf{sketch}}_{m,n}$. Define $A_k$ that performs rank $k$ approximation for data $\mathbf{A}$. Without noise, this error is $\mathbf{X}_{OPT} = \mathrm{argmin}_{\mathbf{X}\ rank\ k,\in\mathbb{R}^{n\times d}} \|\mathbf{A}-\mathbf{A}\mathbf{X}\mathbf{X}^T\|_F^2$, and after sketching with Gaussian noise of variance $\sigma^2$ is $\mathbf{X}' = \mathrm{argmin}_{\mathbf{X}\ rank\ k,\in\mathbb{R}^{n\times d}} \|(\mathbf{A}+\mathbf{G})-(\mathbf{A}+\mathbf{G})\mathbf{X}\mathbf{X}^T\|_F^2$. An instantiation of the LTM $(R_\sigma, \Pi_{\mathrm{S}}, A_k)$, which tolerates $t'$ corrupted clients, computes linear regression parameters for any $\alpha_{\mathrm{S}} > 0$, as well as any $\alpha > 0$ and sufficiently large constant $\eta$ with probability $1-\beta$ with additive and multiplicative errors:*

$$(1+O(\alpha_{\mathrm{S}}))\|\mathbf{A}-\mathbf{A}\mathbf{X}_{OPT}\mathbf{X}_{OPT}^T\|_F^2 + \tilde{O}\left(\frac{k^2d^3}{\alpha_{\mathrm{S}}^7}\log^3(1/\beta)\varepsilon^{-2}\log(1/\delta)\right)$$

Proof. We will use the following inequality. The applications to vector and matrix norms are straightforward corollaries.

**Lemma 4.3** (Generalized Triangle Inequality [BBC+19]). *For any two real numbers $a, b$ and any $\alpha > 0$*

$$|a^2 - b^2| \leq \alpha \cdot a^2 + \left(1+\frac{1}{\alpha}\right)\cdot(a-b)^2.$$

Let $\mathbf{X}'$ be the matrix returned by the mechanism. We have

$$\begin{aligned}\|\mathbf{A}-\mathbf{A}\mathbf{X}'\mathbf{X}'^T\|_F^2 &\leq (1+\alpha_{\mathrm{S}})\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{A}-\mathbf{S}_i\mathbf{A}\mathbf{X}'\mathbf{X}'^T\right\|_F^2\\ &\leq (1+\alpha_{\mathrm{S}})(1+\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)-\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)\mathbf{X}'\mathbf{X}'^T\right\|_F^2\\ &+ (1+\alpha_{\mathrm{S}})(1+1/\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i-\mathbf{S}_i\mathbf{G}_i\mathbf{X}'\mathbf{X}'^T\right\|_F^2\end{aligned}$$

where the first inequality follows from the subspace embedding property and the second inequality follows from Lemma 4.3. By optimality of $\mathbf{X}'$ for the low rank approximation problem on $\sum_{i\in[s]}\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)$, we then have

$$\begin{aligned}&\|\mathbf{A}-\mathbf{A}\mathbf{X}'\mathbf{X}'^T\|_F^2\\ &\leq (1+\alpha_{\mathrm{S}})(1+\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)-\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\right\|_F^2\\ &+ (1+\alpha_{\mathrm{S}})(1+1/\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i-\mathbf{S}_i\mathbf{G}_i\mathbf{X}'\mathbf{X}'^T\right\|_F^2\\ &\leq (1+\alpha)(1+\alpha_{\mathrm{S}})\Bigg((1+\alpha)(1+\alpha_{\mathrm{S}})\left\|\mathbf{A}-\mathbf{A}\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\right\|_F^2\\ &+ (1+1/\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i-\mathbf{S}_i\mathbf{G}_i\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\right\|_F^2\Bigg)\\ &+ (1+\alpha_{\mathrm{S}})(1+1/\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i-\mathbf{S}_i\mathbf{G}_i\mathbf{X}'\mathbf{X}'^T\right\|_F^2\end{aligned}$$

where the inequality follows from Lemma 4.3 and the subspace embedding property. Now notice that $\mathbb{I}-\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T$ and $\mathbb{I}-\mathbf{X}'\mathbf{X}'^T$ are orthogonal projections, multiplying by which cannot increase the Frobenius norm of a matrix. Therefore, for $\mathbf{X}=\mathbf{X}'$ or $\mathbf{X}=\mathbf{X}_{\mathrm{OPT}}$:

$$\begin{aligned}\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i-\mathbf{S}_i\mathbf{G}_i\mathbf{X}\mathbf{X}^T\right\|_F^2 &= \frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i(\mathbb{I}-\mathbf{X}\mathbf{X}^T)\right\|_F^2\\ &\leq \frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i\right\|_F^2\end{aligned}$$

Then:

$$\begin{aligned}\|\mathbf{A}-\mathbf{A}\mathbf{X}'\mathbf{X}'^T\|_F^2 &\leq (1+\alpha)^2(1+\alpha_{\mathrm{S}})^2\|\mathbf{A}-\mathbf{A}\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\|_F^2\\ &+ (1+\alpha_{\mathrm{S}})(1+1/\alpha)(2+\alpha)\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i\right\|_F^2\end{aligned}$$

[2] The linear transformations we use are random, but this randomness is not private, so we don't need to trust the servers not to leak it. Instead, linear transformations are sampled before secure computation.

Using Lemma 4.1, we have $\frac{1}{s}\left\|\sum_{i\in[s]}\mathbf{S}_i\mathbf{G}_i\right\|_F^2 \leq 2n\cdot\sigma^2\cdot d\log(1/\beta)$. To simplify the error, observe that for the sketching matrix by [Coh16] (see Table 5), we can set $s = \frac{\log k}{\alpha_{\mathrm{S}}}$ and $m = \frac{k\log k}{\alpha_{\mathrm{S}}^2}\log(1/\beta)$. Setting $\alpha = \alpha_S$ and the bound on the variance from Theorem 3.4 and plugging this in above, we can simplify the additive and multiplicative errors.

$$\begin{aligned}
&\|\mathbf{A}-\mathbf{A}\mathbf{X}'\mathbf{X}'^T\|_F^2\\
&\quad\leq (1+\alpha)^2(1+\alpha_{\mathrm{S}})^2\|\mathbf{A}-\mathbf{A}\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\|_F^2\\
&\qquad\qquad+\eta(1+\alpha_{\mathrm{S}})(\alpha+\alpha^{-1})\sigma^2\cdot n\cdot d\cdot\log\frac{1}{\tilde{\beta}}\\
&\quad\leq (1+O(\alpha_{\mathrm{S}}))\|\mathbf{A}-\mathbf{A}\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\|_F^2\\
&\qquad\qquad+\tilde{O}\left(\frac{k^2d^3}{\alpha_{\mathrm{S}}^7}\log^3(1/\beta)\varepsilon^{-2}\log(1/\delta)\right)
\end{aligned}$$

□

**Theorem 4.4.** *Let $\varepsilon \geq 0$, $\delta \in (0,1)$, $t' < n$ and $\sigma$ be chosen as described in Theorem 3.4. Let $\mathbf{S} \sim \mathcal{D}_{m,n}^{\mathsf{sketch}}$. Define $A_\lambda$ that performs linear regression on the sketched noisy inputs, outputting argmin$_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2+\lambda\|\mathbf{x}\|^2$, where the error is measured by min$_{\mathbf{x}}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2+\lambda\|\mathbf{x}\|^2$. An instantiation of the LTM $(R_\sigma, \Pi_{\mathrm{S}}, A_\lambda)$, which tolerates $t'$ corrupted clients, computes linear regression parameters for any $\alpha_{\mathrm{S}} > 0$ with probability $1-\beta$ with additive and multiplicative errors:*

$$\begin{aligned}
&\left(1+\tilde{O}\left(\alpha_S+\frac{d^3\alpha_S^{-5}\log^5\cdot\varepsilon^{-2}\log\frac{1}{\delta}}{\beta\lambda}+\frac{d^6\alpha_S^{-10}\log^{10}\cdot\varepsilon^{-4}\log^2\frac{1}{\delta}}{\beta\lambda^2}\right)\right)\\
&\cdot(\|\mathbf{A}\mathbf{x}_{OPT}-\mathbf{b}\|^2+\lambda\|\mathbf{x}_{OPT}\|^2)\\
&+\tilde{O}\left(\frac{d^3\alpha_S^{-5}\log^5\frac{1}{\beta}\cdot\varepsilon^{-2}\log\frac{1}{\delta}}{\lambda}+\frac{d^6\alpha_S^{-10}\log^{10}\frac{1}{\beta}\cdot\varepsilon^{-4}\log^2\frac{1}{\delta}}{\lambda^2}\right)
\end{aligned}$$

Note that we can obtain a constant multiplicative error as long as the regularization factor $\lambda$ depends on a sufficiently large polynomial in $d$, while being independent of $n$. Generally such a relationship still has good generalization properties when training a regression model. If $\lambda$ is not sufficiently large, any underlying sketching matrix will set $\alpha_{\mathrm{S}}$ to be a large constant, i.e. 1/3, that still guarantees validity of the sketch.

## 5 Experimental Evaluation

In contrast to the central model, utility in the local model decreases with the number of clients $n$. For ridge regression and low-rank approximation, our theoretical results shows that this performance decrease can be avoided in the LTM with the Gaussian mechanism. It is natural to ask for which parametrizations of the Gaussian mechanism the performance remains acceptable. Therefore, we evaluate empirically how the error of ridge regression and low-rank approximation develops as $n$ grows using synthetic benchmarks. The first part of the experiments demonstrates that, as the number of clients increases, the asymptotic error in the LTM is between the error in the central and local models. The second part of the experiments describes the minimal computational overhead introduced by the use of MPC for distributing the execution of the LTM.

### 5.1 Utility

In this section we will first evaluate the utility of our model for low-rank approximation and then do the same for ridge regression.

*5.1.1 Low-Rank Approximation.* We here investigate the error when performing low-rank approximation. We evaluate the error for synthetic datasets as well as four real-world datasets.

*Setup.* Our mechanism in the LTM adds Gaussian noise sampled from $\mathcal{N}(0,\sigma^2)$ to every entry in input data matrix $\mathbf{A}$. The variance $\sigma^2$ depends on privacy parameters $\varepsilon$ and $\delta$ and is chosen proportional to $n^{-p}$ for $p \in [0,1]$. This enables us to interpolate between the local model ($p = 0$) and the LTM ($p = 1$). As a baseline for the central model, we implemented MOD-SULQ [CSS12], an approach where $\mathbf{A}^T\mathbf{A}$ is perturbed by adding Gaussian noise. For low-rank approximation the additive error plays a significant role (see Theorem 4.2), so we measure the error in terms of the excess risk

$$\psi = \frac{\|\mathbf{A}-\mathbf{A}\mathbf{X}'\mathbf{X}'^T\|_F^2-\|\mathbf{A}-\mathbf{A}\mathbf{X}_{\mathrm{OPT}}\mathbf{X}_{\mathrm{OPT}}^T\|_F^2}{n},$$

where $\mathbf{X}_{\mathrm{OPT}}$ denotes the optimal solution and $\mathbf{X}'$ denotes the solution after adding noise to the training data. We vary the privacy parameter $\varepsilon \in \{0.01, 0.03, 0.05, 0.1\}$ and $k$ based on the dataset at hand. For each dataset we measure $\psi$ for the central model and the Gaussian mechanism with $p \in \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0\}$, by reporting the average $\psi$ over 20 runs of the algorithms per dataset.

*Software and Hardware.* All mechanisms are implemented in Python 3.6.9, making use of numpy. Experiments were executed on an Ubuntu 18.04 LTS machine, with an Intel Core i7-10510U CPU clocked at 1.8GHz and 16GB of RAM.

*Datasets.* We evaluate our mechanism on synthetic and real datasets, enabling us to vary the number of sampled points $n$ and interpolating between datasets of different sizes.

The synthetic datasets have a large spectral gap from the $k$th to the $(k+1)$th singular value with random bases, which emphasizes the performance difference between the various private mechanisms. This is achieved by first producing a matrix $\mathbf{A}'$ where every entry is sampled from $\mathcal{N}(0,1)$ and then changing its singular values such that there are exactly $k$ big ones and the rest are small. More specifically, we set $\mathbf{A} = \mathbf{U}'\Sigma\mathbf{V}'$, where $\mathbf{U}'\Sigma'\mathbf{V}'$ is the SVD of $\mathbf{A}$ and $\Sigma$ is a diagonal matrix with the first $k$ values set to $\sqrt{n/k}$ and the rest set to $1/n$.

The parameters of the four real-world datasets from the UC Irvine Machine Learning Repository are given in Table 2. For more thorough descriptions on the datasets, see Appendix D.4.

*Results and Interpretation.* Figure 2 shows our results for two privacy regimes ($\varepsilon \in \{0.01, 0.05\}$) with synthetic data and Table 2 shows the error with real-world datasets. In all chosen parameter settings for synthetic data, we observe that as $n$ grows, the error in the LTM asymptotically approaches the error in the central model, both on real and synthetic datasets (see Figure 2 and Table 2). Table 2 shows that notably, on real-world datasets, our approach performs significantly better than the Gaussian mechanism in the local model and is closer to the central MOD-SULQ [CSS12] mechanism. For more combinations of parameters, see Appendix D.3.

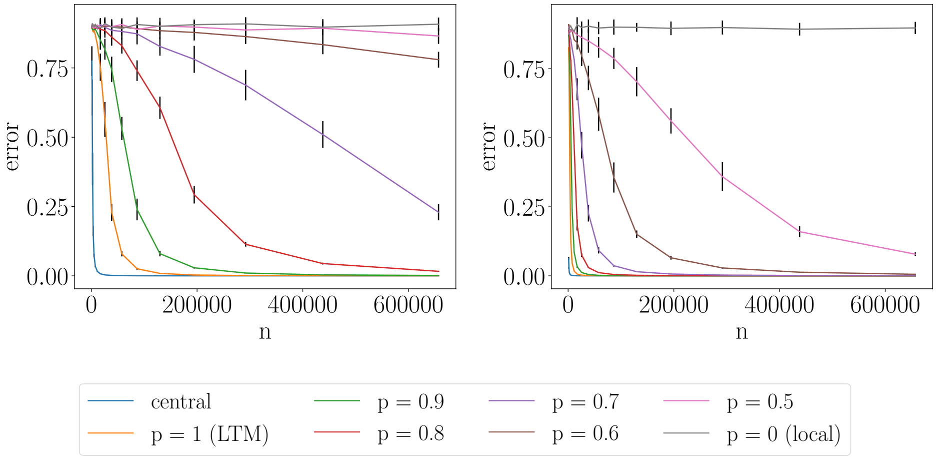

**Figure 2: Plots depicting the asymptotic behavior of error $\psi$ for $\varepsilon \in \{0.01, 0.05\}$ (left, right), with $d = 50$ and $k = 5$. The grey line depicts the error of the local mechanism and the orange one depicts our approach. The other lines resemble different values of $p$. The standard deviations are depicted by the vertical black lines.**

**Table 2: Experimental evaluation of error $\psi$ on real-world datasets, including standard deviations. Here we are in the setting where $\varepsilon = 0.03$.**

| **Dataset** | $n$ | $d$ | $k$ | **Local** ($\psi$) | **Our** ($\psi$) | **Central** ($\psi$) |
|---|---|---|---|---|---|---|
| Power [HB12] | 2049280 | 6 | 3 | $17.03 \pm 9.506$ | $(4.507 \pm 1.977) \times 10^{-8}$ | $(1.066 \pm 0.107) \times 10^{-10}$ |
| Elevation [Kau13] | 434874 | 2 | 1 | $0.162 \pm 0.198$ | $(5.894 \pm 7.537) \times 10^{-8}$ | $(8.027 \pm 9.154) \times 10^{-12}$ |
| Ethylene [Fon15] | 4178504 | 18 | 5 | $0.268 \pm 0.045$ | $(6.810 \pm 1.037) \times 10^{-8}$ | $(4.022 \pm 6.175) \times 10^{-10}$ |
| Songs [BM11] | 515345 | 89 | 15 | $43.51 \pm 2.543$ | $(7.853 \pm 0.490) \times 10^{-5}$ | $(1.895 \pm 0.147) \times 10^{-10}$ |

5.1.2 *Ridge Regression.* We now also investigate the error for the ridge regression problem due to its wide spectrum of possible error bounds from Theorem 4.4, depending on problem parameters, as well as due to its singular dependency on a multiplicative error. Therefore, we use approximation factor as an error measure:

$$\phi = \frac{\|\mathbf{A}\mathbf{x}' - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}'\|_2^2}{\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}_{\mathrm{OPT}}\|_2^2}$$

where $\mathbf{x}_{\mathrm{OPT}}$ denotes the optimal solution and $\mathbf{x}'$ denotes the solution after adding noise to the training data.

As we did for low-rank approximation, we evaluate the error for synthetic datasets, as well as for four real-world datasets. It is not possible to interpolate between two different datasets in order to gain insights into asymptotic behavior. For synthetic datasets, varying the number of sampled points $n$ allows us to interpolate between datasets of different sizes. Real-world datasets are useful to gauge the relative performance of our mechanisms, and to benchmark our synthetic datasets against the behavior of the mechanisms in practice.

We utilize the same software, hardware and real-world datasets as for low-rank approximation.

*Setup.* As a baseline for the central model, we implemented the so-called Sufficient Statistics Perturbation (SSP) algorithm [VS09]. Operating between SSP and a locally private algorithm, our LTM model approach adds Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ to every entry of $\mathbf{A}$ and $\mathbf{b}$. Our choice of $\sigma^2$ is proportionate to varying powers of $n^{-p}$, ranging from $p = 1$ (the LTM model) to $p = 0$ (the local model).

We generate synthetic data by first sampling every entry in $\mathbf{A}$ from $\mathcal{N}(0, 1)$, then sampling $\mathbf{x}_{\mathrm{OPT}}$ such that all entries are sampled from $\mathcal{N}(0, \mu^2)$, and finally setting $\mathbf{b} = \mathbf{A}\mathbf{x}_{\mathrm{OPT}}$. We vary parameters $d \in \{3, 10, 50\}$, $\varepsilon \in \{0.01, 0.03, 0.05, 0.1\}$, $\lambda \in \{1, 10, 100\}$, and $\mu^2 \in \{1, n, n^2\}$. For all combinations of those parameters, we generated 17 synthetic datasets of sizes $\{1000^{i \cdot 1.5} | i = 0, \ldots, 16\}$. For each of those we then measure $\phi$ for the SSP mechanism and the Gaussian mechanism with $p \in \{1, 0.9, 0.8, 0.7, 0.6, 0.5, 0\}$, by running the algorithms 30 times per dataset and reporting the average $\phi$.

In addition to using synthetic data-sets, we again also evaluated our mechanism for ridge regression on the same 4 real-world datasets as for low-rank approximation (see Table 2). For more thorough descriptions on those datasets, see Appendix D.4.

*Results and Interpretation.* Figure 3 provides our experimental results based on synthetic data, and Table 3 shows the error resulting from real-world datasets. As expected, our approach performs asymptotically better than the Gaussian mechanism in the local model, but worse than SSP in the central model. We also found that as $n$ increases, the error of our approach asymptotically approaches the error in the central model. For $p = 0.9, 0.8, 0.7, 0.6$, the

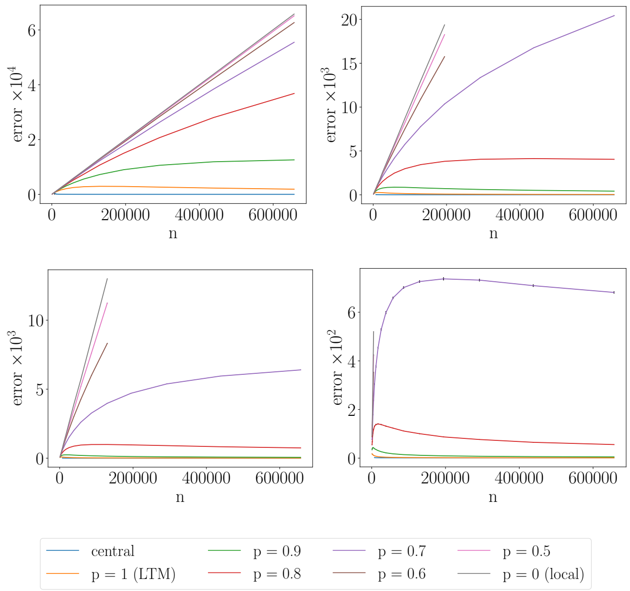

**Figure 3: Plots depicting the asymptotic behavior of error $\phi$ for $\varepsilon \in \{0.01, 0.03, 0.05, 0.1\}$ (top left, top right, bottom left, bottom right), with $d = 10$, $\lambda = 10$ and $\mu^2 = n$. The grey line depicts the error of the local mechanism, the blue line does it in the central model and the orange one depicts our approach. The other lines resemble different values of $p$. In most cases the standard deviations are so small, that it is not possible to see those.**

error eventually decreases for a sufficiently large $n$ for the privacy regimes we considered ($\varepsilon \in \{0.01, 0.03, 0.05, 0.1\}$).

For $p = 0.5$ though we saw a significant jump towards the local model in terms of asymptotic behaviour. In all the settings we considered, $p = 0.5$ showed an asymptotic increase of the error. We tested on synthetic datasets of up to $n = 40$ million, and the error also increases in this regime.

Table 3 shows the results of applying the local, our mechanism and SSP on real-world data. They include standard deviations, though for many of the results those are so small that they appear as 0 in the table. We found, that our mechanism performs significantly better than the Gaussian mechanism for local privacy.

See Appendix D.2 for error plots when considering more parameter combinations.

**Table 3: Experimental evaluation of error $\phi$ on real-world datasets, including standard deviations. Here we are in the setting where $\varepsilon = 0.03$ and $\lambda = 10$.**

| **Dataset** | Local | Our | Central |
|---|---|---|---|
| Power | 2.364 ± 0.007 | 1.055 ± 0.000 | 1.001 ± 0.001 |
| Elevation | 1.939 ± 0.008 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| Ethylene | 3.125 ± 0.002 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| Songs | 20.64 ± 0.016 | 1.000 ± 0.000 | 1.000 ± 0.000 |

## 5.2 Running Time

The only overhead of the LTM over the local model is the execution of the linear transform in MPC, and comes from distributing $\mathbf{A}$ among the servers and then performing the matrix multiplication $\mathbf{SA}$ in MPC, where $\mathbf{S} \in \mathbb{R}^{m \times n}$ is chosen according to [Coh16] with sparsity $s$.

We implemented our mechanism in a popular and easy to use MPC framework, the MP-SPDZ framework [Kel20], and run it on AWS t3.large instances. Our implementation makes use of additive secret sharing over the ring of integers modulo $2^{64}$, resulting in no communication between servers, and semi-honest security against an adversary that corrupts all but one servers. We fix the dimensionality $d = 10$ and $m = 50$, and vary the number of clients $n \in \{100000, 250000, 500000, 750000, 1000000\}$, the sparsity $s \in \{1, 10, 20, 30, 40, 50\}$ of the sketch (which dictates how many linear transformations we need to apply) and the number of servers $S \in \{2, 3\}$. The data matrix $\mathbf{A}$ is generated at random such that all entries are smaller than $2^{32}$ and the sketch $\mathbf{S}$ is generated according to our mechanism. Every parameter setting is then evaluated by running the protocol 10 times and averaging over the running times.

Table 4 provides running times per server and the total communication load for $s = 1$ using 2 servers. Notably, even with one million clients, the computation on each server lasts less than 2 seconds. This shows that securely distributing in the LTM using MPC techniques is a practically relevant approach which does not hinder the computational efficiency of the overall differentially-private data-analysis system. For running times with $S = 3$ servers and a varying sparsity $s$, see Appendix D.1.

**Table 4: Computation cost $T_{\text{MPC}}$ and communication cost $C_{\text{MPC}}$ of the LTM using MPC for varying number of clients $n$. Here $m = 100$, $d = 10$, $s = 1$ and we are working with $S = 2$ servers.**

| $n$ | $T_{\text{MPC}}$ (sec) | $C_{\text{MPC}}$ (MB) |
|---|---|---|
| 100000 | 0.177 ± 0.028 | 16.016 |
| 250000 | 0.453 ± 0.036 | 40.016 |
| 500000 | 0.824 ± 0.101 | 80.016 |
| 750000 | 1.240 ± 0.157 | 120.016 |
| 1000000 | 1.683 ± 0.130 | 160.016 |

# 6 Conclusion

We propose the LTM as an MPC-inspired model that interpolates between local and central models for differential privacy. We highlight the power of this model by studying ridge regression and low rank approximation. Key applications of this model are sketching algorithms. An interesting question may be to see if sketching algorithms in general and more specific Johnson Lindenstrauss transforms can be used to obtain private mechanisms in this model. Such results will likely be powerful for designing private algorithms for a variety of data analysis problems.

## Acknowledgments

The research described in this paper has received funding from: the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 803096 (SPEC); the Danish Independent Research Council under Grant-ID DFF-2064-00016B (YOSO); DFF-3103-00077B (CryptoDigi) and the Danish Independent Research Council under Grant-ID 1051-001068 (ACBD).

## References

[Ach03] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[AG21] Apple and Google. Exposure Notification Privacy-preserving Analytics (ENPA). White paper, 2021.

[AHK06] Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, volume 4110 of *Lecture Notes in Computer Science*, pages 272–279. Springer, 2006.

[BBC+19] Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for $k$-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2019.

[BBDS12] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 410–419. IEEE Computer Society, 2012.

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, page 128–138, New York, NY, USA, 2005. Association for Computing Machinery.

[BDWY16] Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 284–309, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[BEM+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong Privacy for Analytics in the Crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP, 2017.

[BHNS20] Amos Beimel, Iftach Haitner, Kobbi Nissim, and Uri Stemmer. On the round complexity of the shuffle model. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part II*, page 683–712, Berlin, Heidelberg, 2020. Springer-Verlag.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS, 2017.

[BM11] T. Bertin-Mahieux. YearPredictionMSD. UCI Machine Learning Repository, 2011. DOI: https://doi.org/10.24432/C50K61.

[BNST20] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. Practical locally private heavy hitters. *J. Mach. Learn. Res.*, 21:16:1–16:42, 2020.

[BS15] Raef Bassily and Adam D. Smith. Local, private, efficient protocols for succinct histograms. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 127–135. ACM, 2015.

[BST14] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2014.

[CDN15] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing.* Cambridge University Press, 2015.

[Coh16] Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 278–287. SIAM, 2016.

[CSS12] Kamalika Chaudhuri, Anand D. Sarwate, and Kaushik Sinha. Near-optimal differentially private principal components. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 998–1006, 2012.

[CSU+19] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology – EUROCRYPT*. 2019.

[CW13] Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 81–90. ACM, 2013.

[CWZ21] T. Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825 – 2850, 2021.

[CY23] Albert Cheu and Chao Yan. Necessary Conditions in Multi-Server Differential Privacy. In *14th Innovations in Theoretical Computer Science Conference (ITCS)*, 2023.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT*, 2006.

[DKN+23] Ivan Damgård, Hannah Keller, Boel Nelson, Claudio Orlandi, and Rasmus Pagh. Differentially private selection from secure distributed computing. In *The Web Conference*, WWW, 2023.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 2006.

[DNMR14] Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Robust and private bayesian inference. In *Algorithmic Learning Theory*, pages 291–305, Cham, 2014. Springer International Publishing.

[DR14] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 2014.

[DTTZ14] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 11–20. ACM, 2014.

[EFM+19] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019.

[FGWC16] James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving bayesian data analysis. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, page 192–201, Arlington, Virginia, USA, 2016. AUAI Press.

[Fon15] Jordi Fonollosa. Gas sensor array under dynamic gas mixtures. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5WP4C.

[GGK+21] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 463–488. Springer, 2021.

[GKK+23] Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, and Adam Sealfon. On computing pairwise statistics with local differential privacy. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[GKMP20] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3505–3514. PMLR, 2020.

[GX17] Slawomir Goryczka and Li Xiong. A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[HB12] Georges Hebrail and Alice Berard. Individual household electric power consumption. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C58K54.

[HKL+23] Mikael Møller Høgsgaard, Lior Kamma, Kasper Green Larsen, Jelani Nelson, and Chris Schwiegelshohn. Sparse dimensionality reduction revisited. *CoRR*, abs/2302.06165, 2023.

[HP14] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[HR12] Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, page 1255–1268, New York, NY, USA, 2012. Association for Computing Machinery.

[HR13] Moritz Hardt and Aaron Roth. Beyond worst-case analysis in private singular vector computation. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 331–340, New York, NY, USA, 2013. Association for Computing Machinery.

[Kau13] Manohar Kaul. 3D Road Network (North Jutland, Denmark). UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5GP51.

[Kel20] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1575–1590. ACM, 2020.

[KKMM13] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. *J. Priv. Confidentiality*, 5(1), 2013.

[KLN+08] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS, 2008.

[KN12] Daniel M. Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1195–1206. SIAM, 2012.

[KST12] Daniel Kifer, Adam D. Smith, and Abhradeep Thakurta. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *The 25th Annual Conference on Learning Theory*, COLT, 2012.

[KT] Michael Kapralov and Kunal Talwar. *On differentially private low rank approximation*, pages 1395–1414.

[LBKW14] Yingyu Liang, Maria-Florina Balcan, Vandana Kanchanapally, and David P. Woodruff. Improved distributed principal component analysis. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3113–3121, 2014.

[LKO22] Xiyang Liu, Weihao Kong, and Sewoong Oh. Differential privacy and robust statistics in high dimensions. In *Conference on Learning Theory*, 2022.

[LM00] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pages 1302–1338, 2000.

[MASN16] Kentaro Minami, HItomi Arai, Issei Sato, and Hiroshi Nakagawa. Differential privacy without sensitivity. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[MKFI22] Jason Milionis, Alkis Kalavasis, Dimitris Fotakis, and Stratis Ioannidis. Differentially private regression with unbounded covariates. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 3242–3273. PMLR, 28–30 Mar 2022.

[MM13] Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 91–100. ACM, 2013.

[MPBB19] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[Nik23] Aleksandar Nikolov. Private query release via the johnson-lindenstrauss transform. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4982–5002. SIAM, 2023.

[NN13] Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2013.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84. ACM, 2007.

[Pis99] Gilles Pisier. *The volume of convex bodies and Banach space geometry.* Cambridge Tracts in Mathematics. 94, 1999.

[Sar06] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152. IEEE Computer Society, 2006.

[She19] Or Sheffet. Old techniques in differentially private linear regression. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 789–827. PMLR, 22–24 Mar 2019.

[Sta21a] Nina Mesing Stausholm. Improved differentially private euclidean distance approximation. In Leonid Libkin, Reinhard Pichler, and Paolo Guagliardo, editors, *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, pages 42–56. ACM, 2021.

[Sta21b] Nina Mesing Stausholm. Improved differentially private euclidean distance approximation. In Leonid Libkin, Reinhard Pichler, and Paolo Guagliardo, editors, *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, pages 42–56. ACM, 2021.

[Ste20] Thomas Steinke. Multi-central differential privacy, 2020.

[TWM+23] Kunal Talwar, Shan Wang, Audra McMillan, Vojta Jina, Vitaly Feldman, Bailey Basile, Aine Cahill, Yi Sheng Chan, Mike Chatzidakis, Junye Chen, Oliver Chick, Mona Chitnis, Suman Ganta, Yusuf Goren, Filip Granqvist, Kristine Guo, Frederic Jacobs, Omid Javidbakht, Albert Liu, Richard Low, Dan Mascenik, Steve Myers, David Park, Wonhee Park, Gianni Parsa, Tommy Pauly, Christian Priebe, Rehan Rishi, Guy Rothblum, Michael Scaria, Linmao Song, Congzheng Song, Karl Tarbe, Sebastian Vogt, Luke Winstrom, and Shundong Zhou. Samplable anonymous aggregation for private federated data analysis, 2023.

[Upa18] Jalaj Upadhyay. The price of privacy for low-rank factorization. In *Advances in Neural Information Processing Systems*, 2018.

[VS09] Duy Vu and Aleksandra B. Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *ICDM Workshops 2009, IEEE International Conference on Data Mining*, 2009.

[VTJ22] Prateek Varshney, Abhradeep Thakurta, and Prateek Jain. (nearly) optimal private linear regression for sub-gaussian data via adaptive clipping. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1126–1166. PMLR, 02–05 Jul 2022.

[Wan18] Yu-Xiang Wang. 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, UAI*, 2018.

[War65] Stanley L. Warner. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *Journal of the American Statistical Association*, 1965.

[WFS15] Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

[WGX18] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. In *Advances in Neural Information Processing Systems*, 2018.

[WHZ+23] Di Wang, Lijie Hu, Huanyu Zhang, Marco Gaboardi, and Jinhui Xu. Generalized linear models in non-interactive local differential privacy with public data. *Journal of Machine Learning Research*, 24(132):1–57, 2023.

[WSX18] Di Wang, Adam D. Smith, and Jinhui Xu. Noninteractive locally private learning of linear models via polynomial approximations. In *International Conference on Algorithmic Learning Theory*, 2018.

[ZMW17] Kai Zheng, Wenlong Mou, and Liwei Wang. Collect at once, use effectively: making non-interactive locally private learning possible. In *Proceedings of the 34th International Conference on Machine Learning*, ICML, 2017.

## A Sparse Oblivious Subspace Embeddings

Oblivious subspace embeddings (OSEs) [Sar06], allow for faster approximation algorithms for problems in linear algebra. Achlioptas [Ach03] provided the first Johnson-Lindenstrauss transform with some amount of sparsity. The first embedding with asymptotically smaller number of non-zeros than dense Johnson-Lindenstrauss transforms was probably due to [KN12], who also applied them in the context of subspace embeddings. In a remarkable result, the super-sparse version with only 1 non-zero entries per column, was first analyzed in [CW13] and improved independently in [MM13, NN13], by increasing the target dimension $m$. In [NN13], the sparse Johnson-Lindenstrauss transform [KN12] was studied with a wider range of parameters. Subsequent works [Coh16, HKL+23] further analyze and improve the relationship between $m$ and $s$, emphasizing different parameters. Table 5 provides the exact interplay between $m$ and $s$ in those works. [LBKW14] proposes an approach to boost the success probability $\beta$ of an OSE, which gives an alternative to increasing the target dimension.

When dealing with rank $k$ approximation, we will condition on

$$(1-\alpha_S)\|\mathbf{A}-\mathbf{A}\mathbf{X}\mathbf{X}^T\|_F^2 \leq \|\mathbf{S}(\mathbf{A}-\mathbf{A}\mathbf{X}\mathbf{X}^T)\|_F^2 \leq (1+\alpha_S)\|\mathbf{A}-\mathbf{A}\mathbf{X}\mathbf{X}^T\|_F^2$$

for all rank $k$ orthogonal matrices $\mathbf{X}$. When dealing with regression, we will condition on

$$(1-\alpha_S)\|\mathbf{A}\mathbf{x}'-\mathbf{b}\|^2 \leq \|\mathbf{S}(\mathbf{A}\mathbf{x}-\mathbf{b})\|^2 \leq (1+\alpha_S)\|\mathbf{A}\mathbf{x}'-\mathbf{b}\|^2$$

for all $\mathbf{x} \in \mathbb{R}^d$. For the parameters given here, this is true with the probability given in table 5 (assuming $k = d$ in the case of regression). For the case of regression in particular, it is sometimes beneficial to select $\alpha_S$ as large as possible. A sufficiently largest value of $\alpha_S$ such that $\mathbf{S}$ still provide a subspace embedding guarantee is $1/3$. Throughout this paper, we will sometimes bound $\alpha_S$ by 1.

We further will give our utility proofs using the sparsity/target dimension bounds from [Coh16]. Other tradeoffs are possible, but they have worse bounds for most ranges of parameters. The results in [MM13, NN13, HKL+23] give different trade-offs depending on which set of parameters are considered the most dominant. Notably, the $s = 1$ sketches of [MM13, NN13] result in an additive error of $\tilde{O}\left(\frac{k^4 d^3}{\alpha_S^4 \beta^2}\varepsilon^{-2}\log\frac{1}{\delta}\right)$ for low rank approximation and a term $\tilde{O}\left(\frac{d^7\varepsilon^{-2}\log\frac{1}{\delta}}{\alpha_S^5\beta^2\lambda} + \frac{d^{14}\varepsilon^{-4}\log^2\frac{1}{\delta}}{\alpha_S^6\beta^4\lambda^2}\right)$ in both the multiplicative and additive error. The utility guarantees using the bounds from [HKL+23] change only by logarithmic factors compared to those in Theorems 4.2 and 4.4.

## B Privacy Proofs

Proof Sketch of Lemma 3.3. This proof can be seen as analogous to the security proof of Theorem 4 in [TWM+23]. Recall that each server learns nothing more from a perfectly secure multi-party computation protocol other than what is implied by their own input and the output of the function being evaluated. Then there is a simulator $\mathtt{Sim}^{R,\Pi,A}_{S_{cor},S_{cor}=c}(x)$ whose output is $(\varepsilon, \delta)$ indistinguishable from the adversary's view $\mathtt{View}^{R,\Pi,A}_{S_{cor},C_{cor}}(x)$. Note that the secret shares that are part of the view of the corrupted servers in the protocol $\Pi$ are independent of the input values of the corrupted clients, and all can be easily simulated by random sampling.

A bit more formally, let $x$ be an input dataset, where $x_i$ is the input value contributed by client $i$ and $\tilde{x}_i$ is the vector of input shares for server $i$ that serves as input to $\Pi$. There is a mapping from $x \in \mathcal{X}^n$ to $\tilde{x} \in \tilde{\mathcal{X}}^k$. Then for any two datasets $x, x'$ that differ in one entry $x_i$ with $i \notin C_{cor}$:

$$\begin{aligned}
&\Pr[\mathtt{View}^{R,\Pi,A}_{S_{cor},C_{cor}}(x) \in S] \\
&\leq \Pr[\mathtt{Sim}^{R,\Pi,A}_{S_{cor},C_{cor}}(f(R(x_1),\ldots,R(x_n),\{x_i\}_{i\in C_{cor}}) \in S] \\
&\leq e^{\varepsilon}\Pr[\mathtt{Sim}^{R,\Pi,A}_{S_{cor},C_{cor}}(f(R(x_1'),\ldots,R(x_n'),\{x_i'\}_{i\in C_{cor}}) \in S] + \delta \\
&= e^{\varepsilon}\Pr[\mathtt{View}^{R,\Pi,A}_{S_{cor},C_{cor}}(x') \in S] + \delta
\end{aligned}$$

□

**Lemma B.1.** *Let $\gamma \in (0,1)$, and let $\mathbf{S} \sim \mathcal{D}^{\mathsf{sketch}}_{m,n}$. Then the probability that $\mathbf{S}$ has at least $(1-\gamma)\frac{n}{m}$ and at most $(1+\gamma)\frac{n}{m}$ non-zero entries in every row is bounded by;*

$$\Pr\left(\exists i \in [m] : \left|X_i - \frac{n}{m}\right| > \gamma \cdot \frac{n}{m}\right) < 2m\exp\left(-\frac{\gamma^2 n}{2m}\right),$$

*where $X_i$ denotes the number of non-zero entries in row $i$.*

Proof. We only give a proof of the lower bound, as a proof of the upper bound is completely analogous. Let $X_{ij}$ denote the indicator random variable that is 1 if $\mathbf{S}_{ij}$ is non-zero and 0 otherwise. Note that each entry of $\mathbf{S}$ can be thought of as an independent Bernoulli random variable $X_{ij}$, and the number of non-zero entries in a row is the sum of $n$ of these random variables. Fixing a row $k \in [m]$, we can use a Chernoff bound for sums of Bernoulli random variables to get a concentration bound around the expected number of non-zero entries $X_k = \sum_{j=1}^n X_{kj}$ in row $k$. In order to do this, we first need the expected value of $X_k$;

$$\mathbb{E}[X_k] = \sum_{j=1}^n \Pr(X_{kj} = 1) = \frac{n}{m}.$$

Thus a Chernoff bound gives us that

$$\Pr\left(X_k < (1-\gamma)\frac{n}{m}\right) < \exp\left(-\frac{\gamma^2 n}{2m}\right).$$

Applying a union bound over all rows of $\mathbf{S}$ then gives us exactly what we were to prove;

$$\begin{aligned}
\Pr\left(\exists i \in [m] : X_i < (1-\gamma)\frac{n}{m}\right) &\leq \sum_{k=1}^m \Pr\left(X_k < (1-\gamma)\frac{n}{m}\right) \\
&< m\exp\left(-\frac{\gamma^2 n}{2m}\right).
\end{aligned}$$

□

To prove Theorem 3.4, we first prove a simpler version of the theorem below.

**Theorem B.2.** *Let $\varepsilon \geq 0$, $\delta \in (0,1)$, $m \in [n]$, $t' < n$ and*

$$\sigma^2 = \frac{4\eta^2\ln(1.25/(\delta/d - m\exp(-(n-t')/8m)))m^2d^2}{\varepsilon^2(n-t')}$$

*Let $\mathbf{S} \sim \mathcal{D}^{\mathsf{sketch}}_{m,n}$ with one non-zero entry per column. Then as long as input values are bounded above by $\eta$, tuple of algorithms $(R_\sigma, T_{\mathbf{S}}, A)$*

**Table 5: Trade-off between the target dimension $m$ and sparsity $s$ of sparse OSEs that are generated such that they have $s$ non-zeros entries per column. Here $\alpha$ denotes the accuracy of the OSE and $\beta$ denotes its fail probability. Variable $k$ is the parameter to $k$-rank approximation, which can be replaced by dimension $d$ for linear regression.**

| Paper | Target Dimension $m$ | Sparsity $s$ | With Probability |
|---|---|---|---|
| [CW13, MM13, NN13] | $O(\frac{k^2}{\alpha^2\beta})$ | 1 | $1-\beta$ |
| [Coh16] | $O(\frac{k\log(k/\beta)}{\alpha^2})$ | $O(\frac{\log(k/\beta)}{\alpha})$ | $1-\beta$ |
| [HKL$^+$23] | $O(\frac{k}{\alpha^2})$ | $O(\frac{1}{\alpha}\cdot(\frac{k}{\log(1/\alpha)}+k^{2/3}\log^{1/3}(k)))$ | $1-2^{-k^{2/3}}$ |

*is $(\varepsilon,\delta)$-differentially private in the trusted computation model for differential privacy with $t'$ corrupt clients $C'$.*

Proof of Theorem B.2. Let $\varepsilon, \delta, R_\sigma$ and $T_S$ be given as described in the theorem. We then need to prove that mechanism $\mathcal{M}:\mathbb{R}^{n\times d}\to\mathbb{R}^{m\times d}$ defined by $\mathcal{M}(\mathbf{A}) = T_S(R_\sigma(\mathbf{a}_1),\ldots,R_\sigma(\mathbf{a}_n))$ is $(\varepsilon,\delta)$ differentially private. If $\delta/d = \delta \le m\exp(-(n-t')/8m)$, then the $R_\sigma$ outputs $0^d$, which is entirely independent of the input and is therefore $(\varepsilon,\delta)$ differentially private.

Otherwise, $\mathcal{M}(\mathbf{A}) = T_S(R_\sigma(\mathbf{a}_1),\ldots,R_\sigma(\mathbf{a}_n)) = \mathbf{S}(\mathbf{A}+\mathbf{G})$ where $\mathbf{a}_i$ denotes row $i$ of $\mathbf{A}\in\mathbb{R}^{n\times d}$ and $\mathbf{G}\leftarrow\mathcal{N}(0,\sigma^2)^{n\times d}$, is $(\varepsilon,\delta)$ differentially private.

Consider first $d=1$ and let mechanism $\mathcal{M}_1:\mathbb{R}^n\to\mathbb{R}^m$ be defined as $\mathcal{M}_1(\mathbf{a}) = \mathbf{S}(\mathbf{a}+\mathbf{g})$ for $\mathbf{a}\in\mathbb{R}^n$, with $\mathbf{g}\leftarrow\mathcal{N}(0,\sigma^2)^n$.

This is because $\mathbf{S}$ is sampled such that it has only one non-zero entry per column; thus, the columns of $\mathbf{S}$ are orthogonal. This means that the entries of $(\mathbf{S}(\mathbf{a}+\mathbf{g}))_j$ and $(\mathbf{S}(\mathbf{a}+\mathbf{g}))_{j'}$ for $j\ne j'$, have disjoint support. Therefore, the privacy guarantee can be analyzed independently for each entry of $\mathbf{S}(\mathbf{a}+\mathbf{g})$. Then the overall privacy guarantee for $\mathcal{M}_1$ is the guarantee of $\mathbf{S}_i(\mathbf{a}+\mathbf{g})$, where $i$ is the index of the row in $\mathbf{S}$ with the fewest non-zero entries (i.e. the one where the least noise is added from $\mathbf{g}$).

Notice that the $\ell_2$-sensitivity of algorithm $T_S$ is $\Delta_2 T_S = \max_{\mathbf{A},\mathbf{A}'}\|\mathbf{SA}-\mathbf{SA}'\|_2 = 2\eta\sqrt{m}$, where $\mathbf{A},\mathbf{A}'\in\mathbb{R}^{n\times d}$ differ in a single row, and every entry of $\mathbf{A},\mathbf{A}'$ is bounded above by $\eta$.

If $\delta/d = \delta > m\exp(-(n-t')/8m)$, we sample the individual entries of $\mathbf{g}$ from $\mathcal{N}(0,\sigma^2)$ with

$$\sigma^2 = \frac{1}{n-t'}\cdot\frac{4m\eta^2\ln(1.25/(\delta - m\exp(-(n-t')/8m)))m}{\varepsilon^2}.$$

Denote by $E$ the event that the support of $\mathbf{S}(a+g)_j$ has at least $\frac{n-t'}{2m}$ uncorrupted clients, whose set of indices we denote $N$. Setting $\gamma = 1/2$, Lemma B.1 gives that $E$ occurs with probability at least $1-m\exp(-\frac{n-t'}{8m})$. In event $E$, due to the stability of Gaussians, we can formulate $\mathbf{S}(\mathbf{a}+\mathbf{g})_j = (\mathbf{Sa})_j + (\mathbf{Sg})_j$. Due to symmetry of Gaussians around the origin, $(\mathbf{Sg})_j$ is Gaussian distributed with mean zero and variance:

$$\begin{aligned}\tilde{\sigma}^2 &\ge \frac{n-t'}{2m}\sigma^2\\ &= \frac{n-t'}{2m}\cdot\frac{4m\eta^2\ln(1.25/(\delta-m\exp(-(n-t')/8m)))m}{\varepsilon^2(n-t')}\\ &= \frac{2m\eta^2\ln(1.25/(\delta-m\exp(-(n-t')/8m)))}{\varepsilon^2}.\end{aligned}$$

Then in event $E$, $\mathcal{M}_1$ is $(\varepsilon,\delta - m\exp(-(n-t')/8m)))$ differentially private using the Gaussian mechanism (Lemma 2.3). Since event $E$ does not occur with probability at most $m\exp(-(n-t')/8m)$, $\mathcal{M}_1$ is $(\varepsilon,\delta-m\exp(-(n-t')/8m)))$ differentially private except with probability $m\exp(-(n-t')/8m)$. Then $\mathcal{M}_1$ is $(\varepsilon,\delta)$-differentially private.

We now generalize to arbitrary choices of $d$. Note that now $\mathbf{S}_i(\mathbf{A}+\mathbf{G})$ is a d-dimensional vector. Sequential Composition (Lemma C.1) then gives that $\mathcal{M}$ is $(\varepsilon,\delta)$ differentially private, if we sample $\mathbf{G}$'s entries from $\mathcal{N}(0,\sigma^2)$ with

$$\sigma^2 = \frac{4\eta^2\ln(1.25/(\delta/d - m\exp(-(n-t')/8m)))m^2d^2}{\varepsilon^2(n-t')}$$

□

Proof of Theorem 3.4. Let $\varepsilon,\delta,R_\sigma$ and $T_S$ be given as described in the theorem. We then need to prove that mechanism $\mathcal{M}:\mathbb{R}^{n\times d}\to\mathbb{R}^{m\times d}$ defined by $\mathcal{M}(\mathbf{A}) = T_S(R_\sigma(\mathbf{a}_1),\ldots,R_\sigma(\mathbf{a}_n))$ is $(\varepsilon,\delta)$ differentially private. If $\delta/d = \delta \le m\exp(-(n-t')/8m)$, then the $R_\sigma$ outputs $0^d$, which is entirely independent of the input and is therefore $(\varepsilon,\delta)$ differentially private.

Otherwise, $\mathcal{M}(\mathbf{A}) = T_S(R_\sigma(\mathbf{a}_1),\ldots,R_\sigma(\mathbf{a}_n)) = \frac{1}{\sqrt{s}}\sum_{i\in[s]}\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)$ where $\mathbf{a}_i$ denotes row $i$ of $\mathbf{A}\in\mathbb{R}^{n\times d}$ and $\mathbf{G}_i\leftarrow\mathcal{N}(0,\sigma^2)^{n\times d}$. Here, $\mathbf{S}$ has been decomposed such that $\mathbf{S} = \frac{1}{\sqrt{s}}\sum_i^s\mathbf{S}_i$, where $\mathbf{S}_i\in\{-1,0,1\}^{m\times n}$ is a matrix with only one non-zero entry per column.

We argue $(\varepsilon/s,\delta/s)$-DP based on Theorem B.2. Note that the non-zero entry per column in each $\mathbf{S}_i$ is not sampled uniformly at random from all possible rows; rather, for $\mathbf{S}_i$ is sampled uniformly at random from the remaining $n-i$ rows (removing the one that was already chosen to be non-zero, sampling without replacement). Therefore, we replace $n$ with $n-s$[3] when applying Theorem B.2.

By sequential composition, releasing $\mathbf{S}_i(\mathbf{A}+\mathbf{G}_i)$ for all $i\in[s]$ satisfies $(\varepsilon,\delta)$-DP, and thus by post-processing, the scaled sum of these is also $(\varepsilon,\delta)$-DP.

□

## C Utility Proofs

We will use sequential composition of differentially private mechanisms.

**Lemma C.1** (Sequential Composition [DMNS06]). *Let $\mathcal{M}_i:\mathcal{X}\to\mathcal{Y}_i$ be an $(\varepsilon_i,\delta_i)$-differentially private mechanism for $i\in[k]$. Then mechanism $\mathcal{M}:\mathcal{X}\to\prod_{i=1}^k\mathcal{Y}_i$ defined as $\mathcal{M}(x) = (\mathcal{M}_1(x),\ldots,\mathcal{M}_k(x))$, is $(\sum_{i=1}^k\varepsilon_i,\sum_{i=1}^k\delta_i)$-differentially private.*

**Lemma C.2.** *Let $\mathbf{G}\in\mathbb{R}^{n\times d}$ such that all entries in $\mathbf{G}$ are independently sampled from a normal distribution with mean 0 and variance at most $\sigma^2$. Further, let $V$ be a set of $d$-dimensional vectors lying in a*

[3] Since $n \gg m > s$, subtracting $s$ will not have a large effect.

*$k$-dimensional subspace. Then with probability at least $1-\beta$ for some absolute constant $\eta$*

$$\begin{aligned}\sup_{\mathbf{x}\in V} \|\mathbf{G}\mathbf{x}\|_2^2 \leq n \cdot \sigma^2 \cdot \|\mathbf{x}\|^2 + \eta \cdot \sigma^2 \cdot \|\mathbf{x}\|^2 \\ \cdot (\sqrt{(k+\log 1/\beta)\cdot n} + (k+\log 1/\beta)).\end{aligned}$$

*Moreover, with probability at least $1-\beta$*

$$\|\mathbf{G}\|_F^2 \leq \eta \cdot \sigma^2 \cdot n \cdot d \cdot \log 1/\beta.$$

Proof. Denote by $\mathbf{G}_i$ the $i$th row of $\mathbf{G}$. We start by observing that $\mathbf{G}_i^T\mathbf{x}$ is Gaussian distributed with mean 0 and variance $\sigma^2 \cdot \|\mathbf{x}\|^2$. Thus, we focus our attention on controlling $\frac{\|\mathbf{G}\mathbf{x}\|^2}{\sigma^2\cdot\|\mathbf{x}\|^2}$, that is, we assume that $\mathbf{x}$ is a unit vector and that the entries of $\mathbf{G}$ are standard normal Gaussian random variables. The overall claim then follows by rescaling. Consider an $\varepsilon$-net $N_\varepsilon$ of $V$, that is, for every $\mathbf{x} \in V$ with unit norm there exists a vector $\mathbf{x}' \in N_\gamma$ with $\|\mathbf{x}-\mathbf{x}'\| \leq \varepsilon$. Such nets exist with $|N_\gamma| \leq \exp(\eta \cdot k \log 1/\gamma)$ for some absolute constant $\eta$, see [Pis99]. Suppose $\gamma = 1/4$. Using concentration bounds for sums of Gaussians (see for example Lemma 1 of [LM00]), we then have for any $\|x'\|$ with probability $1-\beta$

$$\begin{aligned}&\mathbb{P}\big[\exists \mathbf{x}' \in N_\gamma : \|\mathbf{G}\mathbf{x}'\|^2 - \mathbb{E}[\|\mathbf{G}\mathbf{x}'\|^2] \\ &\geq 2(\sqrt{(\log N_\gamma + \log 1/\beta)\cdot n} + (\log N_\gamma + \log 1/\beta))\big] \\ \leq& |N_\gamma| \cdot \exp(-(\eta \cdot k + \log 1/\beta)) \leq \beta.\end{aligned}$$

for some absolute constant $\eta$.

We now extend this argument to all vectors, using an argument from [AHK06] (Lemma 4 of that reference). Let $\mathbf{U}$ be an orthogonal basis of $V$. Then our goal is to control $\|\mathbf{G}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{U}^T\mathbf{G}^T\mathbf{G}\mathbf{U}x$. Define the matrix $\mathbf{B} := \mathbf{U}^T\mathbf{G}^T\mathbf{G}\mathbf{U} - n \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Note that $\mathbb{E}[\mathbf{U}^T\mathbf{G}^T\mathbf{G}\mathbf{U}] = n\cdot\mathbf{I}$ and that $\|\mathbf{B}\mathbf{x}\|$ is the deviation of $\|\mathbf{G}\mathbf{x}\|$ around its expectation. Let $\|B\|_{op} := \sup_{\mathbf{x}\in V} \|Bx\|$.

$$\begin{aligned}\|\mathbf{B}\|_{op} = \langle \mathbf{B}\mathbf{x}, \mathbf{x}\rangle &= \langle \mathbf{B}\mathbf{x}', \mathbf{x}'\rangle + \langle \mathbf{B}(\mathbf{x}'+\mathbf{x}), \mathbf{x}'-\mathbf{x}\rangle \\ \leq& 2(\sqrt{(\log N_\gamma + \log 1/\beta)\cdot n} + (\log N_\gamma + \log 1/\beta)) \\ &+ \|\mathbf{B}\|_{op} \cdot \|\mathbf{x}'+\mathbf{x}\|\|\mathbf{x}'-\mathbf{x}\| \\ \leq& 2(\sqrt{(\log N_\gamma + \log 1/\beta)\cdot n} + (\log N_\gamma + \log 1/\beta)) \\ &+ 2\gamma \cdot \|\mathbf{B}\|_{op} \\ \leq& 2(\sqrt{(\log N_\gamma + \log 1/\beta)\cdot n} + (\log N_\gamma + \log 1/\beta)) \\ &+ 1/2 \cdot \|\mathbf{B}\|_{op}\end{aligned}$$

Rearranging implies that $\|\mathbf{B}\|_{op} \leq 4(\sqrt{(\log N_\gamma + \log 1/\beta)\cdot n} + (\log N_\gamma + \log 1/\beta))$. The first claim now follows by rescaling $\eta$.

For the second claim, we observe that $\|\mathbf{G}\|_F^2$ has a Gaussian distribution with mean 0 and variance $n \cdot d \cdot \sigma^2$. The same concentration inequality we applied above also implies that the probability that $\|\mathbf{G}\|_F$ exceeds $2\sqrt{\sigma^2 \cdot n \cdot d \cdot \log 1/\beta}$ is at most $1-\beta$. □

Proof of Lemma 4.1. We first argue that $\mathbf{G}$ is Gaussian distributed. Each $G_i$ has independent Gaussian entries and multiplying a Gaussian with a random Rademacher does not change the distribution. Therefore, the entries of $S_iG_i$ are likewise Gaussian distributed, with mean 0 and variance at most $(1+\gamma)\frac{n}{m}\sigma^2 \leq 2\frac{n}{m}\sigma^2$ due to Lemma B.1. Concluding, the variance of the entries of $\mathbf{G} = \sum_{i\in[s]} S_iG_i$ is therefore at most $2s\frac{n}{m}\sigma^2$. Applying Lemma C.2, we then have with probability $1-\beta$

$$\begin{aligned}\sup_{\mathbf{x}\in V} \|\mathbf{G}\mathbf{x}\|_2^2 \leq 2\frac{n}{m}\sigma^2 \cdot \|\mathbf{x}\|^2 \\ + \eta \cdot 2s\frac{n}{m}\sigma^2 \cdot \|\mathbf{x}\|^2 \cdot (\sqrt{(k+\log 1/\beta)\cdot m} + (k+\log 1/\beta)).\end{aligned}$$

and

$$\|\mathbf{G}\|_F^2 \leq \eta \cdot 2sn\sigma^2 \cdot d \cdot \log 1/\beta$$

as desired □

Proof of Theorem 4.4. Let $\varepsilon$, $\delta$, $R_\sigma$ and $T_{\mathrm{S}}$ be given as in the theorem. We consider the output $A_{reg}(T_{\mathrm{S}}(R_\sigma(\mathbf{a}_1), \ldots, R_\sigma(\mathbf{a}_n)))$.

If $n \leq 8m\ln(dm/\delta) + t'$, then the $R_\sigma$ outputs $0^d$, then the optimal $\mathbf{x}' = 0^d$, leading to error $\|\mathbf{A}\mathbf{x}' - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}'\|_2^2 = \|\mathbf{b}\|^2 \leq \eta' n \leq \eta' m \ln(dm/\delta) + t'$.

Otherwise, $\mathcal{M}(\mathbf{A}) = T_{\mathrm{S}}(R_\sigma(\mathbf{a}_1), \ldots, R_\sigma(\mathbf{a}_n)) = \frac{1}{\sqrt{s}}\sum_{i\in[s]} \mathbf{S}_i(\mathbf{A} + \mathbf{G}_i)$ where $\mathbf{a}_i$ denotes row $i$ of $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{G} \leftarrow \mathcal{N}(0,\sigma^2)^{n\times d}$, is $(\varepsilon,\delta)$ differentially private.

To streamline the presentation, we give the analysis without corrupted clients. Adding corrupted clients merely changes the analysis along the same lines as Theorem 4.2.

We first the control the terms $\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{G}_i\mathbf{x}' - \mathbf{g}_i)\|^2$ and $\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{G}_i\mathbf{x}_{\mathrm{OPT}} - \mathbf{g}_i)\|^2$. Using Lemma 4.1 with an added coordinate of $-1$ to both $\mathbf{x}'$ and $\mathbf{x}_{\mathrm{OPT}}$, we get

$$\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{G}_i\mathbf{x} - \mathbf{g}_i)\|^2 \leq \eta \cdot \sigma^2 nd \log\frac{1}{\beta}(\|\mathbf{x}\|^2 + 1) \tag{1}$$

for either $\mathbf{x} = \mathbf{x}_{\mathrm{OPT}}$ or $\mathbf{x} = \mathbf{x}'$ and for a sufficiently large constant $\eta$. Then we have

$$\begin{aligned}&\|\mathbf{A}\mathbf{x}' - \mathbf{b}\|^2 + \lambda\|x'\|^2 \leq (1+\alpha_S)\|\mathbf{S}(\mathbf{A}\mathbf{x}' - \mathbf{b})\|^2 + \lambda\|x'\|^2 \\ =&(1+\alpha_S)\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{A}\mathbf{x}' - \mathbf{b})\|^2 + \lambda\|x'\|^2 \\ =&(1+\alpha_S)\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i((\mathbf{A} + \mathbf{G}_i - \mathbf{G}_i)\mathbf{x}' - \mathbf{b} + \mathbf{g}_i - \mathbf{g}_i)\|^2 + \lambda\|x'\|^2 \\ \leq&(1+\alpha_S)\frac{1}{s}\left((1+\alpha_S)\|\sum_{i\in[s]} \mathbf{S}_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}' - (\mathbf{b}+\mathbf{g}_i))\|^2\right. \\ &\left.+\left(1+\frac{1}{\alpha_S}\right)\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{G}_i\mathbf{x}' - \mathbf{g}_i)\|^2\right) + \lambda\|\mathbf{x}'\|^2 \\ \leq&(1+\alpha_S)^2 \cdot \left(\frac{1}{s}\|\sum_{i\in[s]} S_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}' - (\mathbf{b}+\mathbf{g}_i)\|^2 + \lambda \cdot \|\mathbf{x}'\|^2\right) \\ &+ 2\cdot\left(1+\frac{1}{\alpha_S}\right)\eta\sigma^2 nd\log\frac{1}{\beta}(\|\mathbf{x}'\|^2 + 1)\end{aligned} \tag{2}$$

where the second to last inequality follows by applying Lemma 4.3 and the final inequality follows from Equation 1. By optimality of $\mathbf{x}'$ for the instance $\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i(\mathbf{A}\mathbf{x}' + \mathbf{G}_i - (\mathbf{b}+\mathbf{g}_i))\|^2 + \lambda\|x'\|^2$, we then have

$$
\begin{aligned}
&\frac{1}{s}\|\sum_{i\in[s]} S_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}'-(\mathbf{b}+\mathbf{g}_i)\|^2+\lambda\cdot\|\mathbf{x}'\| \\
&\le \frac{1}{s}\|\sum_{i\in[s]} S_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i)\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2
\end{aligned}
$$

which likewise implies

$$
\|\mathbf{x}'\| \le \frac{1}{\lambda}\cdot\left(\frac{1}{s}\|\sum_{i\in[s]} S_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i))\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2\right).
$$

Insertion both bounds back into Equation 2, we obtain

$$
\begin{aligned}
&\|\mathbf{A}\mathbf{x}'-\mathbf{b}\|^2+\lambda\|x'\|^2 \\
&\le (1+\alpha_S)^2\left(\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i))\|^2+\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2\right) \\
&+2\left(1+\frac{1}{\alpha_S}\right)\eta\sigma^2 nd\log\frac{1}{\beta}+2\left(1+\frac{1}{\alpha_S}\right)\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda} \qquad (3) \\
&\cdot\left(\frac{1}{s}\|\sum_{i\in[s]} S_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i))\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2\right) \qquad (4)
\end{aligned}
$$

We now turn our attention to $\|\sum_{i\in[s]}\mathbf{S}_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i))\|^2$. Using Lemma 4.3 and Equation 1, we have

$$
\begin{aligned}
&\frac{1}{s}\|\sum_{i\in[s]} \mathbf{S}_i((\mathbf{A}+\mathbf{G}_i)\mathbf{x}_{\mathrm{OPT}}-(\mathbf{b}+\mathbf{g}_i))\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2 \\
&\le(1+\alpha_S)\frac{1}{s}\|\sum_{i\in[s]}\mathbf{S}_i(\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b})\|^2 \\
&\quad+\left(1+\frac{1}{\alpha_S}\right)\frac{1}{s}\|\sum_{i\in[s]}\mathbf{S}_i(\mathbf{G}_i\mathbf{x}_{\mathrm{OPT}}-\mathbf{g}_i)\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2 \\
&\le(1+\alpha_S)^2\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b}\|^2+\left(1+\frac{1}{\alpha_S}\right)\eta\sigma^2 nd\log\frac{1}{\beta}(\|\mathbf{x}_{\mathrm{OPT}}\|^2+1) \\
&\quad+\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2 \\
&\le(1+\alpha_S)^2\left(\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b}\|^2+\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2\right) \\
&\quad+\left(1+\frac{1}{\alpha_S}\right)\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}(\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2+\lambda) \\
&\le\left((1+\alpha_S)^2+\left(1+\frac{1}{\alpha_S}\right)\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}\right)\left(\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b}\|^2+\lambda\cdot\|\mathbf{x}_{\mathrm{OPT}}\|^2\right) \\
&\quad+\left(1+\frac{1}{\alpha_S}\right)\eta\sigma^2 nd\log\frac{1}{\beta}
\end{aligned}
$$

Inserting this into Equation 4 and collecting all the terms, we obtain

$$
\begin{aligned}
&\|\mathbf{A}\mathbf{x}'-\mathbf{b}\|^2+\lambda\|x'\|^2 \\
&\le\left((1+\alpha_S)^4+\frac{32}{\alpha_S}\left(\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}+\frac{2}{\alpha_S}\left(\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}\right)^2\right)\right) \qquad (5) \\
&\left(\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b}\|^2+\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2\right) \\
&+\frac{32}{\alpha_S}\left(\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}+\frac{2}{\alpha_S}\left(\frac{\eta\sigma^2 nd\log\frac{1}{\beta}}{\lambda}\right)^2\right)
\end{aligned}
$$

By our choice of $\sigma^2$ from Theorem 3.4 and using the sketching matrix of [Coh16] (see Table 5), we have $\sigma^2\cdot n\in O(d^2\log^4 d\cdot\alpha_S^{-4}\cdot\log^4\frac{1}{\beta}\varepsilon^{-2}\cdot\log\frac{1}{\delta})$. Thus, we have

$$
\begin{aligned}
&\|\mathbf{A}\mathbf{x}'-\mathbf{b}\|^2+\lambda\|x'\|^2 \\
&\le(\|\mathbf{A}\mathbf{x}_{\mathrm{OPT}}-\mathbf{b}\|^2+\lambda\|\mathbf{x}_{\mathrm{OPT}}\|^2)\cdot\Bigg(1+15\alpha_S \\
&\quad+\tilde{O}\left(\frac{d^3\alpha_S^{-5}\log^5\frac{1}{\beta}\cdot\varepsilon^{-2}\log\frac{1}{\delta}}{\lambda}+\frac{d^6\alpha_S^{-10}\log^{10}\frac{1}{\beta}\cdot\varepsilon^{-4}\log^2\frac{1}{\delta}}{\lambda^2}\right)\Bigg) \\
&\quad+\tilde{O}\left(\frac{d^3\alpha_S^{-5}\log^5\frac{1}{\beta}\cdot\varepsilon^{-2}\log\frac{1}{\delta}}{\lambda}+\frac{d^6\alpha_S^{-10}\log^{10}\frac{1}{\beta}\cdot\varepsilon^{-4}\log^2\frac{1}{\delta}}{\lambda^2}\right)
\end{aligned}
$$

□

As a final remark, if $\lambda$ is sufficiently large, one should choose $\alpha_S$ to be as small as possible, minimizing the tradeoff between $\alpha_S$ and $\frac{\alpha_S^{-5}}{\lambda}$. If $\lambda$ is not sufficiently large, one should choose $\alpha_S=1/3$ such that S is an oblivious subspace embedding, but the target dimension does not have a prohibitively large dependency on the sketch distortion.

# D Experimental Evaluation

In this section we provide the results of more combinations of parameters for our utility and running time experiments. We also provide more in depth descriptions of the real-world datasets we used.

## D.1 Running Time of LTM

Table 6 provides running times per server and the total communication load in our model for $s=1$ using 3 servers, and Table 7 shows those when varying $s$.

## D.2 Ridge Regression

On Figure 4, we provide plots for the same setting of parameters as on Figure 3 ($d=10$ and $\mu^2=n$). Though, we also vary $\lambda\in\{1,10,100\}$ here. The experiments show that varying $\lambda$ does not change the asymptotic behavior of any of the mechanisms we investigated, in the sense that a variance proportional to $n^{-0.6}$ or lower eventually had decreasing error, while a variance proportional to $n^{0.5}$ produced increasing errors. The increase of $\lambda$ does however produce a significantly lower error in all settings (note the

**Table 6: Computation cost $T_{\text{MPC}}$ and communication cost $C_{\text{MPC}}$ of the LTM using MPC for varying number of clients $n$. Here $m = 100$, $d = 10$, $s = 1$ and we are working with $S = 3$ servers.**

| $n$ | $T_{\text{MPC}}$ (sec) | $C_{\text{MPC}}$ (MB) |
|---|---|---|
| 100000 | 0.172 ± 0.019 | 24.024 |
| 250000 | 0.457 ± 0.058 | 60.024 |
| 500000 | 0.861 ± 0.102 | 120.024 |
| 750000 | 1.219 ± 0.128 | 180.024 |
| 1000000 | 1.641 ± 0.165 | 240.024 |

**Table 7: Computation cost $T_{\text{MPC}}$ and communication cost $C_{\text{MPC}}$ of the LTM using MPC for varying sparsity $s$. Here $m = 100$, $d = 10$, $n = 500000$ and we are working with $S = 3$ servers.**

| $s$ | $T_{\text{MPC}}$ (sec) | $C_{\text{MPC}}$ (MB) |
|---|---|---|
| 1 | 0.851 ± 0.099 | 80.016 |
| 10 | 6.863 ± 0.388 | 800.016 |
| 20 | 13.002 ± 0.445 | 1600.016 |
| 30 | 20.356 ± 1.171 | 2400.016 |
| 40 | 28.023 ± 0.660 | 3200.016 |
| 50 | 33.708 ± 1.384 | 4000.016 |

$y$-axes on the figure). Our theoretical results likewise predict this behaviour, as while the error bounds of Theorem 4.4 improve with increasing $\lambda$, they do not affect the dependency on $n$. Thus, we view the experiments as a confirmation that the theoretical bounds, while potentially improvable, express the correct asymptotic relationship between parameters and approximation bounds.

## D.3 Low-Rank Approximation

Figure 5 provides the error for low-rank approximation, where both $\varepsilon$ and $k$ are varied.

## D.4 Real-World Datasets

In addition to synthetic datasets, we also evaluated our mechanism for ridge regression on 4 datasets from the UC Irvine Machine Learning Repository. The following table provides their number of entries $n$ and dimensionality $d$. A more thorough description of the datasets is found below the table.

| **Dataset** | $n$ | $d$ |
|---|---|---|
| Power [HB12] | 2049280 | 6 |
| Elevation [Kau13] | 434874 | 2 |
| Ethylene [Fon15] | 4178504 | 18 |
| Songs [BM11] | 515345 | 89 |

- The first dataset consists of electric power consumption measurements in one household [HB12] and the feature we try to predict is sub_metering_3. We ignore the date and time features and the data points that had missing values. This leaves us with 6 features (plus the one we are predicting) and 2049280 data points.
- The Elevation dataset [Kau13] consists of 434874 open street map elevation measurements from North Jutland, Denmark. We predict the elevation from the longitude and latitude features.
- The Ethylene dataset [Fon15] contains recordings of sensors exposed to a mixture of gas. We trained on the part where the sensors were exposed to a mixtures of Ethylene and CO in air. The feature we are predicting is the last one TGS2620, which leaves us with $d = 18$ and $n = 4178504$.
- The Songs dataset [BM11], consists of 89 audio features that are meant to predict the release year of a song.

# E Distributed Computation of Linear Sketches

The linear transformation we consider is motivated by efficient secure distributed cryptographic tools from secure multi-party computation; in particular, we can use linear secret-sharing. We require this secret sharing scheme to be information-theoretically secure. We use an additive secret sharing scheme, such that clients $C_1, \ldots, C_n$ generate $k$ shares of their noisy inputs $x^1, \ldots, x^n$, where $k$ is the number of servers that compute the linear transformation. We use $[x^i]$ to denote a secret sharing of some client $C_i$'s noisy input $x^i$ for some $i \in [n]$. For some field $\mathbb{F}$ of size $p$ and some prime number $p$, $[x^i]$ consists of shares $x^i_1, \ldots, x^i_k \in \mathbb{F}$ such that $\sum_{j=1}^{k} x^i_j = x^i$. To split a secret into $k$ shares, a client can sample $k-1$ random field elements $x^i_1, \ldots, x^i_{k-1}$ and compute the last secret share as $x^i - \sum_{j=1}^{k-1} x^i_j$. Therefore, it also seems intuitive that an adversary that sees all but one of the shares knows nothing about the input. For every $j \in [k]$, server $S_j$ receives $\{x^i_j\}_{i \in n}$ from the respective parties. We define these shares such that the security of our distributed protocol does not rely on any computational assumption.

Since the linear transformation is public, servers can apply the transformation locally to their secret shares without any need to communicate with each other. Each server then reveals their shares of the resulting linear transformation, such that the linear sketch can be revealed in the clear. For more details on MPC based on secret sharing, see e.g., [CDN15].

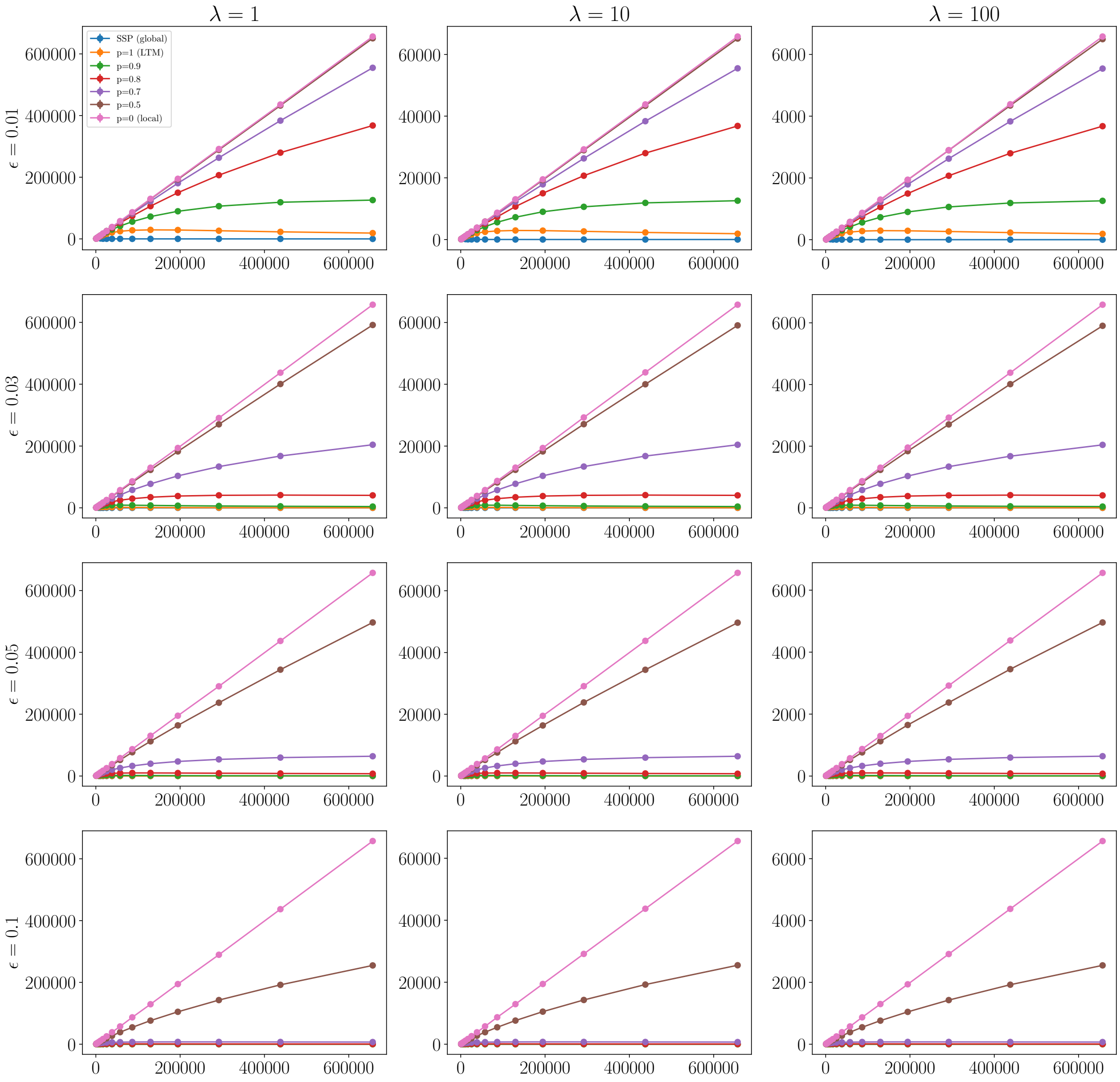

**Figure 4: Plots depicting the error $\phi$ as $n$ increases, for $\varepsilon \in \{0.01, 0.03, 0.05, 0.1\}$ (top to bottom) and $\lambda \in \{1, 10, 100\}$ (left to right). The middle row depicts the same choice of parameters as Figure 3.**

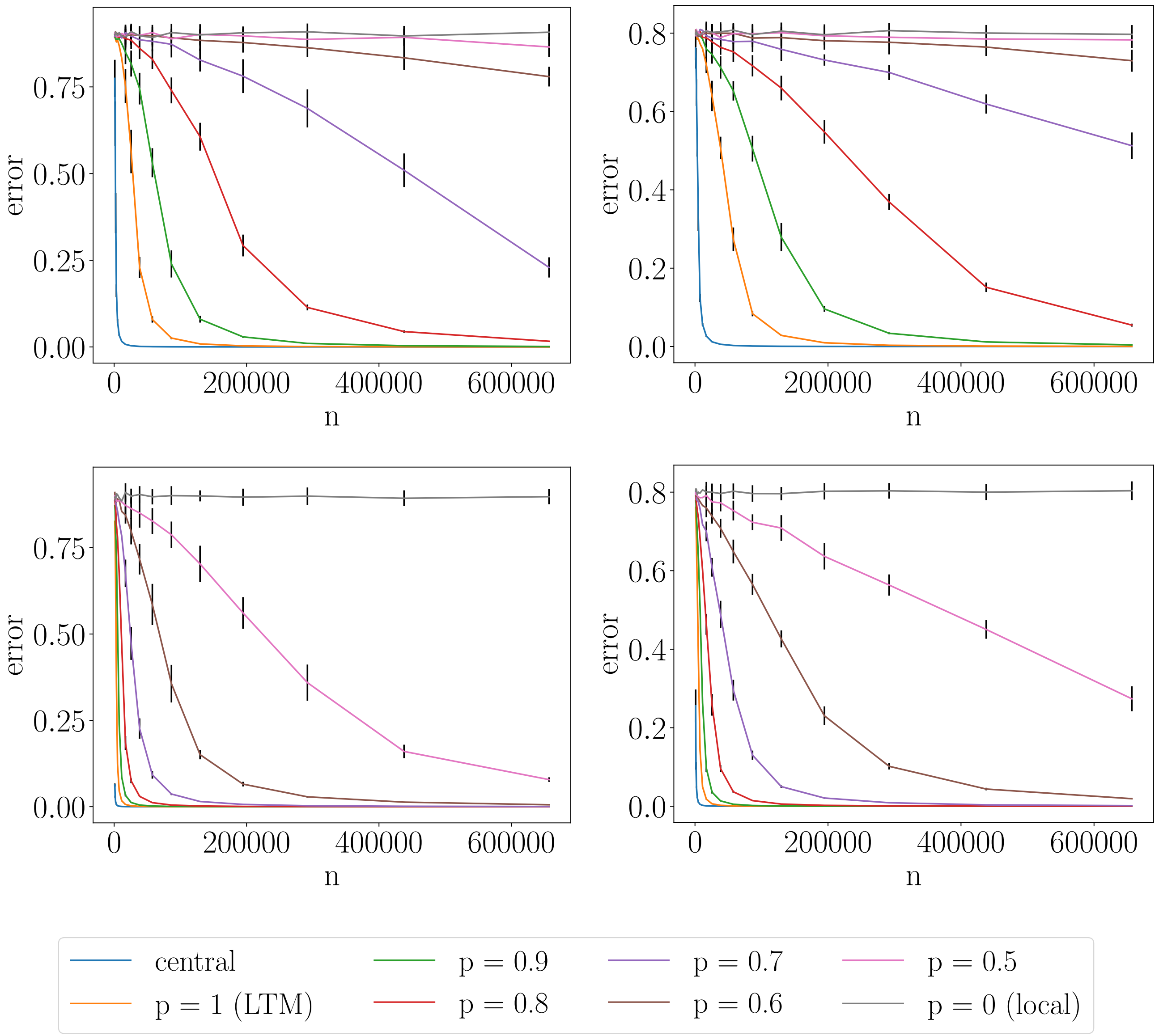

**Figure 5: Plots depicting the asymptotic behavior of error $\psi$ for $\varepsilon \in \{0.01, 0.05\}$ (top, bottom) and $k \in \{5, 10\}$ (left, right), with $d = 50$. The grey line depicts the error of the local mechanism and the orange one depicts our approach. The other lines resemble different values of $p$. The standard deviations are depicted by the vertical black lines.**