entropy

MDPI

Article

Approximating Functions with Approximate Privacy for Applications in Signal Estimation and Learning

Naima Tasnim [1], Jafar Mohammadi [2], Anand D. Sarwate [3] and Hafiz Imtiaz [1,*]

[1] Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka P.O. Box 1205, Bangladesh; 0421062525@eee.buet.ac.bd
[2] Nokia, Werinherstraße 91, 81541 Munich, Germany; jafar.mohammadi@nokia.com
[3] Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, 94 Brett Road, Piscataway, NJ 08854-8058, USA; ads221@soe.rutgers.edu
* Correspondence: hafizimtiaz@eee.buet.ac.bd

Abstract: Large corporations, government entities and institutions such as hospitals and census bureaus routinely collect our personal and sensitive information for providing services. A key technological challenge is designing algorithms for these services that provide useful results, while simultaneously maintaining the privacy of the individuals whose data are being shared. Differential privacy (DP) is a cryptographically motivated and mathematically rigorous approach for addressing this challenge. Under DP, a randomized algorithm provides privacy guarantees by approximating the desired functionality, leading to a privacy–utility trade-off. Strong (pure DP) privacy guarantees are often costly in terms of utility. Motivated by the need for a more efficient mechanism with better privacy–utility trade-off, we propose Gaussian FM, an improvement to the functional mechanism (FM) that offers higher utility at the expense of a weakened (approximate) DP guarantee. We analytically show that the proposed Gaussian FM algorithm can offer orders of magnitude smaller noise compared to the existing FM algorithms. We further extend our Gaussian FM algorithm to decentralized-data settings by incorporating the CAPE protocol and propose capeFM. Our method can offer the same level of utility as its centralized counterparts for a range of parameter choices. We empirically show that our proposed algorithms outperform existing state-of-the-art approaches on synthetic and real datasets.

Keywords: differential privacy; functional mechanism; decentralized-data systems

check for updates

Citation: Tasnim, N.; Mohammadi, J.; Sarwate, A.D.; Imtiaz, H. Approximating Functions with Approximate Privacy for Applications in Signal Estimation and Learning. *Entropy* **2023**, *25*, 825. https://doi.org/10.3390/e25050825

Academic Editors: Eirik Rosnes and Hsuan-Yin Lin

Received: 21 March 2023
Revised: 16 April 2023
Accepted: 26 April 2023
Published: 22 May 2023

1. Introduction

Differential privacy (DP) [1] has emerged as a de facto standard for privacy-preserving technologies in research and practice due to the quantifiable privacy guarantee it provides. DP involves randomizing the outputs of an algorithm in such a way that the presence or absence of a single individual's information within a database does not significantly affect the outcome of the algorithm. DP typically introduces randomness in the form of additive noise, ensuring that an adversary cannot infer any information about a particular record with high confidence. The key challenge is to keep the performance or *utility* of the noisy algorithm close enough to the unperturbed one to be useful in practice [2].

In its pure form, DP measures privacy risk by a parameter ϵ, which can be interpreted as the *privacy budget*, that bounds the log-likelihood ratio of the output of a private algorithm under two datasets differing in a single individual's data. The smaller ϵ used, the greater the privacy ensured, but at the cost of worse performance. In privacy-preserving machine learning models, higher values of ϵ are generally chosen to achieve acceptable utility. However, setting ϵ to arbitrarily large values severely undermines privacy, although there are no hard threshold values for ϵ above which formal guarantees provided by DP become meaningless in practice [3]. In order to improve utility for a given privacy budget, a relaxed definition of differential privacy, referred to as (ϵ, δ)-DP, was proposed [4]. Under this privacy notion, a randomized algorithm is considered privacy-preserving if the privacy loss of the output is smaller than $\exp(\epsilon)$ with a high probability (i.e., with probability at least $1 - \delta$) [5].

Our current work is motivated by the necessity of a decentralized differentially private algorithm to efficiently solve practical signal estimation and learning problems that (i) offers better privacy–utility trade-off compared to existing approaches, and (ii) offers similar utility as the pooled-data (or centralized) scenario. Some noteworthy real-world examples of systems that may need such differentially private decentralized solutions include [6]: (i) medical research consortium of healthcare centers and labs, (ii) decentralized speech processing systems for learning model parameters for speaker recognition, (iii) multi-party cyber-physical systems. To this end, we first focus on improving the privacy–utility trade-off of a well known DP mechanism, called the *functional mechanism (FM)* [7]. The FM approach is more general and requires fewer assumptions on the objective function than other objective perturbation approaches [8,9].

The functional mechanism was originally proposed for "pure" ϵ-DP. However, it involves an additive noise with very large variance for datasets with even moderate ambient dimension, leading to a severe degradation in utility. We propose a natural "approximate" (ϵ, δ)-DP variant using Gaussian noise and show that the proposed *Gaussian FM* scheme significantly reduces the additive noise variance. A recent work by Ding et al. [10] proposed *relaxed FM* using the Extended Gaussian mechanism [11], which also guarantees approximate (ϵ, δ)-DP instead of pure DP. However, we will show analytically and empirically that, just like the original FM, the relaxed FM also suffers from prohibitively large noise variance even for moderate ambient dimensions. Our tighter sensitivity analysis for the Gaussian FM, which is different from the technique used in [10], allows us to achieve much better utility for the same privacy guarantee. We further extend the proposed Gaussian FM framework to the decentralized or "federated" learning setting using the CAPE protocol [6]. Our capeFM algorithm can offer the same level of utility as the centralized case over a range of parameters. Our empirical evaluation of the proposed algorithms on synthetic and real datasets demonstrates the superiority of the proposed schemes over the existing methods. We now review the relevant existing research works in this area before summarizing our contributions.

Related Works. There is a vast literature on the perturbation techniques to ensure DP in machine learning algorithms. The simplest method for ensuring that an algorithm satisfies DP is *input perturbation*, where noise is introduced to the input of the algorithm [2]. Another common approach is *output perturbation*, which obtains DP by adding noise to the output of the problem. In many machine learning algorithms, the underlying objective function is minimized with gradient descent. As the gradient is dependent on the privacy-sensitive data, randomization is introduced at each step of the gradient descent [9,12]. The amount of noise we need to add at each step depends on the *sensitivity* of the function to changes in its input [4]. *Objective perturbation* [8,9,13] is another state-of-the-art method to obtain DP, where noise is added to the underlying objective function of the machine learning algorithm, rather than its solutions. A newly proposed take on output perturbation [14] injects noise after model convergence, which imposes some additional constraints. In addition to optimization problems, Smith [15] proposed a general approach for computing summary statistics using the *sample-and-aggregate* framework and both the Laplace and Exponential mechanisms [16].

Zhang et al. originally proposed *functional mechanism (FM)* [7] as an extension to the Laplace mechanism. FM has been used in numerous studies to ensure DP in practical settings. Jorgensen et al. applied FM in personalized differential privacy (PDP) [17], where the privacy requirements are specified at the user-level, rather than by a single, global privacy parameter. FM has also been combined with homomorphic encryption [18] to obtain both data secrecy and output privacy, as well as with fairness-aware learning [10,19] in classification models. The work of Fredrikson et al. [20], which demonstrated privacy in pharmacogenetics using FM and other DP mechanisms, is of particular interest to us.

Pharmacogenetic models [21–24] contain sensitive clinical and genomic data that need to be protected. However, poor utility of differentially private pharmacogenetic models can expose patients to increased risk of disease. Fredrikson et al. [20] tested the efficacy of such models against attribute inference by using a model inversion technique. Their study shows that, although not explicitly designed to protect attribute privacy, DP can prevent attackers from accurately predicting genetic markers if ϵ is sufficiently small ($\leq$1). However, the small value of ϵ results in poor utility of the models due to excessive noise addition, leading them to conclude that when utility cannot be compromised much, *the existing methods do not give an ϵ for which state-of-the-art DP mechanisms can be reasonably employed*. As mentioned before, Ding et al. [10] recently proposed relaxed FM in an attempt to improve upon the original FM using the Extended Gaussian mechanism [11], which offered approximate DP guarantee.

DP algorithms provide different guarantees than Secure Multi-party Computation (SMC)-based methods. Several studies [25–27] applied a combination of SMC and DP for distributed learning. Gade and Vaidya [25] demonstrated one such method in which each site adds and subtracts arbitrary functions to confuse the adversary. Heikkilä et al. [26] also studied the relationship of additive noise and sample size in a distributed setting. In their model, S data holders communicate their data to M computation nodes to compute a function. Tajeddine et al. [27] used DP-SMC on vertically partitioned data, i.e., where data of the same participants are distributed across multiple parties or data holders. Bonawitz et al. [28] proposed a communication-efficient method for federated learning over a large number of mobile devices. More recently, Heikkilä et al. [29] considered DP in a cross-silo federated learning setting by combining it with additive homomorphic secure summation protocols. Xu et al. [30] investigated DP for multiparty learning in vertically partitioned data setting. Their proposed framework dissects the objective function into single-party and cross-party sub-functions, and applies functional mechanisms and secure aggregation to achieve the same utility as the centralized DP model. Inspired by the seminal work of Dwork et al. [31] that proposed distributed noise generation for preserving privacy, Imtiaz et al. [6] proposed the *Correlation Private Estimation (*CAPE*)* protocol. CAPE employs a similar principle as Anandan and Clifton [32] to *reduce* the noise added for DP in decentralized-data settings.

Our Contributions. As mentioned before, we are motivated by the necessity of a decentralized differentially private algorithm that injects a smaller amount of noise (compared to existing approaches) to efficiently solve practical signal estimation and learning problems. To that end, we first propose an improvement to the existing functional mechanism. We achieve this by performing a tighter characterization of the sensitivity analysis, which significantly reduces the additive noise variance. As we utilize the Gaussian mechanism [33] to ensure (ϵ,δ)-DP, we call our improved functional mechanism *Gaussian FM*. Using our novel sensitivity analysis, we show that the proposed Gaussian FM injects a much smaller amount of additive noise compared to the original FM [7] and the relaxed FM [10] algorithms. We empirically show the superiority of Gaussian FM in terms of privacy guarantee and utility by comparing it with the corresponding non-private algorithm, the original FM [7], the relaxed FM [10], the objective perturbation [8], and the noisy gradient descent [12] methods. Note that the original FM [7] and the objective perturbation [8] methods guarantee pure DP, whereas the other methods guarantee approximate DP. We compare our (ϵ,δ)-DP Gaussian FM with the pure DP algorithms as a means for investigating how much performance/utility gain one can achieve by trading off pure the DP guarantee with an approximate DP guarantee. Additionally, the noisy gradient descent method is a multi-round algorithm. Due to the composition theorem of differential privacy [33], the privacy budgets in multi-round algorithms accumulate across the number of iterations during training. In order to perform better accounting for the total privacy loss in the noisy gradient descent algorithm, we use Rényi differential privacy [34].

Considering the fact that machine learning algorithms are often used in decentralized/federated data settings, we adapt our proposed Gaussian FM algorithm to decentralized/federated data settings following the (CAPE) [6] protocol, and propose capeFM. In many signal processing and machine learning applications, where privacy regulations prevent sites from sharing the local raw data, joint learning across datasets can yield discoveries that are impossible to obtain from a single site. Motivated by scientific collaborations that are common in human health research, CAPE improves upon the conventional decentralized DP schemes and achieves the same level of utility as the pooled-data scenario in certain regimes. It has been shown [6] that CAPE can benefit computations with sensitivies satisfying some conditions. Many functions of interest in machine learning and deep neural networks have sensitivites that satisfy these conditions. Our proposed capeFM algorithm utilizes the Stone–Weierstrass theorem [35] to approximate a cost function in the decentralized-data setting and employs the CAPE protocol.

To summarize, the goal of our work is to improve the privacy–utility trade-off and reduce the amount of noise in the functional mechanism at the expense of approximate DP guarantee for applications of machine learning in decentralized/federated data settings, similar to those found in research consortia. Our main contributions are:

- We propose Gaussian FM as an improvement over the existing functional mechanism by performing a tighter sensitivity analysis. Our novel analysis has two major features: (i) the sensitivity parameters of the data-dependent (hence, privacy-sensitive) polynomial coefficients of the Stone–Weierstrass decomposition of the objective function are free of the dataset dimensionality; and (ii) the additive noise for privacy is tailored for the *order* of the polynomial coefficient of the Stone–Weierstrass decomposition of the objective function, rather than being the same for all coefficients. These features give our proposed Gaussian FM a significant advantage by offering much less noisy function computation compared to both the original FM [7] and the relaxed FM [10], as shown for linear and logistic regression problems. We also empirically validate this on real and synthetic data.
- We extend our Gaussian FM to decentralized/federated data settings to propose capeFM, a novel extension of the functional mechanism for decentralized-data. To this end, we note another significant advantage of our proposed Gaussian FM over the original FM: the Gaussian FM can be readily extended to decentralized/federated data settings by exploiting the fact that the sum of a number of Gaussian random variables is another Gaussian random variable, which is not true for Laplace random variables. We show that the proposed capeFM can achieve the same utility as the pooled-data scenario for some parameter choices. To the best of our knowledge, our work is the first functional mechanism for decentralized-data settings.
- We demonstrate the effectiveness of our algorithms with varying privacy and dataset parameters. Our privacy analysis and empirical results on real and synthetic datasets show that the proposed algorithms can achieve much better utility than the existing state-of-the-art algorithms.

2. Definitions and Preliminaries

Notation. We denote vectors, matrices, and scalars with bold lower case letters ($\mathbf{x}$), bold upper case letters ($\mathbf{X}$), and unbolded letters (N), respectively. We denote indices with lower case letters and they typically run from 1 to their upper case versions ($d \in 1, 2, \ldots, D \triangleq [D]$). The n-th column of a matrix $\mathbf{X}$ is denoted as $\mathbf{x}_n$. We denote the Euclidean (or $\mathcal{L}_2$) norm of a vector and the spectral norm of a matrix with $\|\cdot\|_2$. Finally, we denote the inner product of two matrices $\mathbf{A}$ and $\mathbf{B}$ as $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{A}^\top \mathbf{B})$.

2.1. Definitions

Definition 1 ((ϵ, δ)-Differential Privacy [4])**.** *Let us consider a domain $\mathbb{D}$ of datasets consisting of N records, and $D, D' \in \mathbb{D}$ where D and D' differ in a single record (neighboring datasets). Then,* *for all measurable $\mathbb{S} \subseteq \mathbb{T}$ and all neighboring data sets $D, D' \in \mathbb{D}$, an algorithm $\mathcal{A} : \mathbb{D} \mapsto \mathbb{T}$ provides (ϵ, δ)-differential privacy ((ϵ, δ)-DP) if*

$$\Pr[\mathcal{A}(D) \in \mathbb{S}] \leq \exp(\epsilon) \Pr[\mathcal{A}(D') \in \mathbb{S}] + \delta.$$

This definition is also known as bounded differential privacy (as opposed to unbounded differential privacy [1]). One way to interpret this is that an algorithm $\mathcal{A}$ satisfies (ϵ, δ)-DP if the probability distribution of the output of $\mathcal{A}$ does not change significantly if the input database is changed by one sample. That is to say, whether or not a particular individual takes part in a differentially private study, the outcome of the study is not changed by much. An adversary attempting to identify an individual will not be able to verify the individual's presence or absence in the study with high confidence. The privacy of the individual is thus preserved by plausible deniability. In the definition of DP, (ϵ, δ) are privacy parameters, where lower (ϵ, δ) ensure more privacy. The parameter δ can be interpreted as the probability that the algorithm fails to provide privacy risk ϵ. Note that (ϵ, δ)-DP is known as *approximate* differential privacy whereas ϵ-differential privacy (ϵ-DP) is known as *pure* differential privacy. In general, we denote approximate (bounded) differentially private algorithms with DP. An important feature of DP is that post-processing of the output does not change the privacy guarantee, as long as that post-processing does not use the original data [33]. Among the most commonly used mechanisms for formulating a DP algorithm are additive noise mechanisms such as the Gaussian [4] or Laplace [33] mechanisms, and random sampling using the exponential mechanism [16]. For additive noise mechanisms, the standard deviation of the additive noise is scaled to the *sensitivity* of the computation.

Definition 2 ($\mathcal{L}_p$-Sensitivity [4])**.** *Given neighboring datasets D and D', the $\mathcal{L}_p$-sensitivity of a vector-valued function $f(D)$ is*

$$\Delta := \max_{D, D'} \parallel f(D) - f(D') \parallel_p .$$

We focus on $p = 1$ and 2 in this paper.

Definition 3 (Gaussian Mechanism [33])**.** *Let $f : \mathbb{D} \mapsto \mathbb{R}^D$ be an arbitrary function with $\mathcal{L}_2$-sensitivity Δ. The Gaussian mechanism with parameter τ adds noise scaled to $\mathcal{N}(0, \tau^2)$ to each of the D entries of the output and satisfies (ϵ, δ)-differential privacy for $\epsilon \in (0, 1)$ if*

$$\tau \geq \frac{\Delta}{\epsilon} \sqrt{2 \log \frac{1.25}{\delta}}.$$

Note that, for any given (ϵ, δ) pair, we can calculate a noise variance τ^2 such that addition of a noise term drawn from $\mathcal{N}(0, \tau^2)$ guarantees (ϵ, δ)-differential privacy. There are infinitely many (ϵ, δ) pairs that yield the same τ^2. Therefore, we parameterize our methods using τ^2 [36] in this paper. We refer the reader to [37–39] for a broader discussion of privacy parameter ϵ.

Definition 4 (Rényi Differential Privacy (RDP) [34])**.** *A randomized mechanism $\mathcal{A} : \mathbb{D} \mapsto \mathbb{T}$ is (a, ϵ_r)-Rényi differentially private if, for any adjacent $D, D' \in \mathbb{D}$, the following holds:*

$$D_a(\mathcal{A}(D) \parallel \mathcal{A}(D')) \leq \epsilon_r$$

Here, $D_a(P(x) \parallel Q(x)) = \frac{1}{a-1} \log \mathbb{E}_{x \sim Q} \left(\frac{P(x)}{Q(x)} \right)^a$, and $P(x)$ and $Q(x)$ are probability density functions defined on $\mathbb{T}$.

Analyzing the total privacy loss of a multi-round algorithm, each stage of which is DP, is a challenging task. It has been shown [34,40] that the advanced composition theorem [33] for (ϵ, δ)-differential privacy can be loose. Hence, we use RDP, which offers a much simpler composition rule that is shown to be tight. Here, we review the properties of RDP [34] that we utilize in our analysis in Section 3.

Proposition 1 (From RDP to Differential Privacy [34])**.** *If $\mathcal{A}$ is an (α, ϵ_r)-RDP mechanism, then it also satisfies $\left(\epsilon_r + \frac{\log \frac{1}{\delta_r}}{\alpha - 1}, \delta_r\right)$-differential privacy for any $0 < \delta_r < 1$.*

Proposition 2 (Composition of RDP [34])**.** *Let $\mathcal{A} : \mathbb{D} \mapsto \mathbb{T}_1$ be (α, ϵ_{r1})-RDP and $\mathcal{B} : \mathbb{D} \mapsto \mathbb{T}_2$ be (α, ϵ_{r2})-RDP. Then the mechanism defined as (X, Y), where $X \sim \mathcal{A}(D)$ and $Y \sim \mathcal{B}(X, D)$, satisfies $(\alpha, \epsilon_{r1} + \epsilon_{r2})$-RDP.*

Proposition 3 (RDP and Gaussian Mechanism [34])**.** *If $\mathcal{A}$ has $\mathcal{L}_2$-sensitivity 1, then the Gaussian mechanism $\mathbf{G}_\sigma \mathcal{A}(D) = \mathcal{A}(D) + E$, where $E \sim \mathcal{N}(0, \sigma^2)$ satisfies $\left(\alpha, \frac{\alpha}{2\sigma^2}\right)$-RDP. Additionally, a composition of T Gaussian mechanisms satisfies $\left(\alpha, \frac{\alpha T}{2\sigma^2}\right)$-RDP.*

Correlation Assisted Private Estimation (CAPE) [6]. As mentioned before, we utilize the CAPE protocol for developing capeFM. In Section 5.2 we describe the CAPE trust/collusion model in detail, and discuss how the correlated noise in a decentralized-data setting is used to reduce the excess noise introduced in conventional decentralized DP algorithms. We use the terms "distributed" and "decentralized" interchangeably in this paper. Note that the CAPE scheme, and consequently the proposed capeFM algorithm can be readily extended (see Section III.C of Imtiaz et al. [6]) for federated learning [29] settings.

The CAPE protocol considers a decentralized data setting with S sites and a central aggregator node in an "honest but curious" threat model [6]. For simplicity, we consider the symmetric setting: each site $s \in [S]$ holds a dataset of $N_s = \frac{N}{S}$ disjoint data samples, where the total number of samples across all sites is N. CAPE overcomes the utility degradation in conventional decentralized DP schemes and achieves the same noise variance as that of the pooled-data scenario in certain parameter regimes. The privacy of CAPE is given by Theorem 1 and the claim that the noise variance of the estimator is exactly the same as if all data were present at the aggregator is formalized in Lemma 1. Here, we review the relevant properties of the CAPE scheme for extending our proposed Gaussian FM to the decentralized-data setting. We refer the reader to Imtiaz et al. [6] for the proofs of these properties.

Theorem 1 (Privacy of CAPE scheme [6])**.** *In a decentralized data setting with $N_s = \frac{N}{S}$ and $\tau_s^2 = \tau^2$ for all sites $s \in [S]$, if at most $S_C = \lceil \frac{S}{3} \rceil - 1$ collude after execution, then CAPE guarantees (ϵ, δ)-differential privacy for each site, where (ϵ, δ) satisfy the relation $\delta = 2\frac{\sigma_z}{\epsilon - \mu_z}\phi\left(\frac{\epsilon - \mu_z}{\sigma_z}\right)$, $\epsilon \in (0, 1)$ and (μ_z, σ_z) are given by*

$$\mu_z = \frac{S^3}{2\tau^2 N^2 (1 + S)} \left(\frac{S - S_C + 2}{S - S_C} + \frac{\frac{9}{S - S_C} S_C^2}{S(1 + S) - 3S_C^2} \right),$$
$$\sigma_z = 2\mu_z.$$

Lemma 1 ([6])**.** *Consider the symmetric setting: $N_s = \frac{N}{S}$ and $\tau_s^2 = \tau^2$ for all sites $s \in [S]$. Let the variances of the noise terms e_s and g_s be $\tau_e^2 = \left(1 - \frac{1}{S}\right)\tau_s^2$ and $\tau_g^2 = \frac{\tau_s^2}{S}$, respectively. If we denote the variance of the additive noise (for preserving privacy) in the pooled-data scenario by τ_{pool}^2 and the variance of the estimator a_{cape} by τ_{cape}^2 then CAPE protocol achieves the same noise variance as the pooled-data scenario (i.e., $\tau_{pool}^2 = \tau_{cape}^2$).*

Proposition 4 (Performance improvement using CAPE [6])**.** *If the local noise variances are $\{\tau_s^2\}$ for $s \in [S]$ then the* CAPE *scheme provides a reduction $G = \frac{\tau_{conv}^2}{\tau_{cape}^2} = S$ in noise variance over the conventional decentralized DP scheme in the symmetric setting ($N_s = \frac{N}{S}$ and $\tau_s^2 = \tau^2\ \forall$ $s \in [S]$), where τ_{conv}^2 and τ_{cape}^2 are the noise variances of the final estimate at the aggregator in the conventional scheme and the* CAPE *scheme, respectively.*

Proposition 5 (Scope of CAPE [6])**.** *Consider a decentralized setting with $S > 1$ sites in which site $s \in [S]$ has a dataset D_s of N_s samples and $\sum_{s=1}^{S} N_s = N$. Suppose the sites are employing the* CAPE *scheme to compute a function $f(D)$ with $\mathcal{L}_2$-sensitivity $\Delta(N)$. Denote $\mathbf{n} = [N_1, N_2, \ldots, N_S]$ and observe the ratio $H(\mathbf{n}) = \frac{\tau_{cape}^2}{\tau_{pool}^2} = \frac{\sum_{s=1}^{S} \Delta^2(N_s)}{S^3 \Delta^2(N)}$. Then the* CAPE *protocol achieves $H(\mathbf{n}) = 1$, if (i) $\Delta\left(\frac{N}{S}\right) = S\Delta(N)$ for convex $\Delta(N)$; and (ii) $S^3\Delta^2(N) = \sum_{s=1}^{S} \Delta^2(N_s)$ for general $\Delta(N)$.*

2.2. Functional Mechanism [7]

In this section, we first review the existing functional mechanism through a regression model following [7] before describing our proposed improvement. Let $\mathbb{D}$ be a dataset that contains N samples of the form $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_n \in \mathbb{R}^D$ is the feature vector and $y_n \in \mathbb{R}$ is the response for $n \in [N]$. Without loss of generality, we assume for each sample that $\|\mathbf{x}_n\|_2 \leq 1$. The objective is to construct a regression model that enables one to predict any y_n based on $\mathbf{x}_n$. Depending on the regression model, the mapping function can be of various types. Without loss of generality, it can be parameterized with a D-dimensional vector $\mathbf{w}$ of real numbers. To evaluate whether $\mathbf{w}$ leads to an accurate model, a *cost function* f is defined to measure the deviation between the original and predicted values of y_n, given $\mathbf{w}$ as the model parameters. The optimal model parameter $\mathbf{w}^*$ is defined as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f_D(\mathbf{w}),$$

where the empirical average cost function is

$$f_D(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n, \mathbf{w}). \tag{1}$$

Note that $f_D(\mathbf{w})$ depends on the data samples. In cases where the data are privacy-sensitive, the empirical average cost function $f_D(\mathbf{w})$ (or any function computed from it, such as its gradient or the optimizer $\mathbf{w}^*$) may reveal private information about the members of the dataset. To make the model differentially private, one approach is to add noise to the gradients of the cost function at every iteration [12]. We refer to this approach as *noisy gradient descent* in this paper. Another approach is the to perturb the objective function [7–10]. In particular, the original FM [7] and the relaxed FM [10] use a randomized approximation of the objective function.

Now, recall that $\mathbf{w} \in \mathbb{R}^D$ contains the model parameters $\mathbf{w} = [w_1, w_2, \ldots, w_D]^\top$. We define $\phi(\mathbf{w}) = w_1^{c_1} w_2^{c_2} \ldots w_D^{c_D}$ for some $c_1, c_2, \ldots, c_D \in \mathbb{N}$. Let Φ_j denote the set of all $\phi(\mathbf{w})$ with degree $j \in \mathbb{N}$, i.e.,

$$\Phi_j = \left\{ w_1^{c_1} w_2^{c_2} \ldots w_D^{c_D} \,\middle|\, \sum_{d=1}^{D} c_d = j \right\}.$$

For example, $\Phi_0 = \{1\}$, $\Phi_1 = \{w_1, w_2, \ldots, w_D\}$, and $\Phi_2 = \{w_{d_1} w_{d_2} \mid d_1, d_2 \in [D]\}$. By the Stone–Weierstrass Theorem [35], any continuous and differentiable $f(\mathbf{x}_n, \mathbf{w})$ can be *always* written as a (potentially infinite) sum of monomials of $\{w_d\}$, i.e., for some $J \in [0, \infty)$, we have

$$f(\mathbf{x}_n, \mathbf{w}) = \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi n} \phi(\mathbf{w}),$$

where $\lambda_{\phi n} \in \mathbb{R}$ denotes the coefficient of $\phi(\mathbf{w})$ in the polynomial. Note that $\lambda_{\phi n}$ is a function of the n-th data sample. Consequently, the $f(\mathbf{x}_n, \mathbf{w})$ as expressed above depends on the model parameters through $\phi(\mathbf{w})$ and on the data samples through $\lambda_{\phi n}$. The expression for average cost in (1) can now be written as

$$f_D(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=0}^{J}\sum_{\phi\in\Phi_j}\lambda_{\phi n}\phi(\mathbf{w}) = \sum_{j=0}^{J}\sum_{\phi\in\Phi_j}\left(\frac{1}{N}\sum_{n=1}^{N}\lambda_{\phi n}\right)\phi(\mathbf{w}). \tag{2}$$

For regression analysis on two neighboring datasets $\mathbb{D}$ and $\mathbb{D}'$ differing in a single sample, the $\mathcal{L}_1$-sensitivity of the data-dependent term in (2) is computed as [7]:

$$\sum_{j=0}^{J}\sum_{\phi\in\Phi_j}\frac{1}{N}\left\|\left(\sum_{\mathbb{D}}\lambda_{\phi n} - \sum_{\mathbb{D}'}\lambda_{\phi n}\right)\right\|_1 \le \frac{2}{N}\max_n \sum_{j=0}^{J}\sum_{\phi\in\Phi_j}\|\lambda_{\phi n}\|_1 \triangleq \Delta^{fm}.$$

In FM, Zhang et al. [7] proposed to perturb $f_D(\mathbf{w})$ by injecting Laplace noise with variance $2\left(\frac{\Delta^{fm}}{\epsilon}\right)^2$ into each coefficient of the polynomial. FM achieves ϵ-DP by obtaining the optimal model parameters $\hat{\mathbf{w}}^*$ that minimize the noise-perturbed function $\hat{f}_D(\mathbf{w})$.

As mentioned before, decomposition such as (2) can be performed for any continuous and differentiable cost function $f(\mathbf{x}_n, \mathbf{w})$. However, depending on the complexity of $f(\mathbf{x}_n, \mathbf{w})$, the decomposition may be non-trivial. In Section 4, we show how such decomposition can be performed on linear regression and logistic regression problems, as illustrative examples.

3. Functional Mechanism with Approximate Differential Privacy: Gaussian FM

Zhang et al. [7] computed the $\mathcal{L}_1$-sensitivity Δ^{fm} of the data-dependent terms for linear regression and logistic regression problems. The Δ^{fm} is shown to be $\frac{2}{N}(1+D)^2$ for linear regression, and $\frac{1}{N}\left(\frac{D^2}{4}+3D\right)$ for logistic regression. We note that Δ^{fm} grows quadratically with the ambient dimension of the data samples, resulting in a excessively large amount of noise to be injected into the objective function. Additionally, Ding et al. [10] proposed relaxed FM, a "utility-enhancement scheme", by replacing the Laplace mechanism with the Extended Gaussian mechanism [11], and thus achieving slightly better utility than the original FM at the expense of an approximate DP guarantee instead of a pure DP guarantee. However, Ding et al. [10] showed that the $\mathcal{L}_2$-sensitivity of the data-dependent terms for the logistic regression problem is $\Delta^{rlx\text{-}fm} = \frac{1}{N}\sqrt{\frac{D^2}{16}+D}$. Additionally, using the technique outlined in [10], it can be shown that the $\mathcal{L}_2$-sensitivity of the data-dependent terms is $\Delta^{rlx\text{-}fm} = \frac{2}{N}\sqrt{1+4D+D^2}$ for the linear regression problem (please see Appendix A for details). For both cases, we observe that $\Delta^{rlx\text{-}fm}$ grows linearly with the ambient dimension of the data samples. Therefore, the privacy-preserving additive noise variances in both the original FM and relaxed FM schemes are data-dimensionality dependent, and therefore, can be prohibitively large even for moderate D. Moreover, both FM and relaxed FM schemes add the same amount of noise to each polynomial coefficient $\lambda_{\phi n}$ irrespective of the order j. With a tighter characterization, we show in Section 4 that the sensitivities of these coefficients are different for different order j. We reduce the amount of added noise by addressing these issues and performing a novel sensitivity analysis. The key points are as follows:

- Instead of computing the ϵ-DP approximation of the objective function using the Laplace mechanism, we use the Gaussian mechanism to compute the (ϵ, δ)-DP approximation of $f_D(\mathbf{w})$. This gives a weaker privacy guarantee than the *pure* differential privacy, but provides much better *utility*.
- Recall that the original FM achieves ϵ-DP by adding Laplace noise scaled to the $\mathcal{L}_1$-sensitivity of the data-dependent terms of the objective function $f_D(\mathbf{w})$ in (2). As we use the Gaussian mechanism, we require $\mathcal{L}_2$-sensitivity analysis. To compute the

$\mathcal{L}_2$-sensitivity of the data-dependent terms of the objective function $f_D(\mathbf{w})$ in (2), we first define an *array* Λ_j that contains $\frac{1}{N}\sum_{n=1}^{N}\lambda_{\phi n}$ as its entries for all $\phi(\mathbf{w}) \in \Phi_j$. The term "array" is used because the dimension of Λ_j depends on the cardinality of Φ_j. For example, for $j = 0$, Λ_0 is a scalar because $\Phi_0 = \{1\}$; for $j = 1$, Λ_1 can be expressed as a D-dimensional vector because $\Phi_1 = \{w_1, w_2, ..., w_D\}$; for $j = 2$, Λ_2 can be expressed as a $D \times D$ matrix because $\Phi_2 = \{w_{d_1} w_{d_2} \mid d_1, d_2 \in [D]\}$.

We rewrite the objective function as

$$f_D(\mathbf{w}) = \sum_{j=0}^{J}\sum_{\phi\in\Phi_j}\left(\frac{1}{N}\sum_{n=1}^{N}\lambda_{\phi n}\right)\phi(\mathbf{w}) = \sum_{j=0}^{J}\langle\Lambda_j, \bar{\phi}_j\rangle, \tag{3}$$

where $\bar{\phi}_j$ is the array containing all $\phi(\mathbf{w}) \in \Phi_j$ as its entries. Note that $\bar{\phi}_j$ and Λ_j have the same dimensions and number of elements. We define the $\mathcal{L}_2$-sensitivity of Λ_j as

$$\Delta_j = \max_{\mathbb{D},\mathbb{D}'}\|\Lambda_j^{\mathbb{D}} - \Lambda_j^{\mathbb{D}'}\|_2, \tag{4}$$

where $\Lambda_j^{\mathbb{D}}$ and $\Lambda_j^{\mathbb{D}'}$ are computed on neighboring datasets $\mathbb{D}$ and $\mathbb{D}'$, respectively. Following the Gaussian mechanism [33], we can calculate the (ϵ, δ) differentially private estimate of Λ_j, denoted $\hat{\Lambda}_j$ as

$$\hat{\Lambda}_j = \Lambda_j + e_j, \tag{5}$$

where the noise array e_j has the same dimension as Λ_j, and contains entries drawn i.i.d. from $\mathcal{N}(0, \tau_j^2)$ with $\tau_j = \frac{\Delta_j}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$. Finally, we have

$$\hat{f}_D(\mathbf{w}) = \sum_{j=0}^{J}\langle\hat{\Lambda}_j, \bar{\phi}_j\rangle. \tag{6}$$

As the function $f_D(\mathbf{w})$ depends on the data only through $\{\Lambda_j\}$, this computation satisfies (ϵ, δ)-differential privacy. Our proposed Gaussian FM is shown in detail in Algorithm 1.

Theorem 2 (Privacy of the Gaussian FM (Algorithm 1))**.** *Consider Algorithm 1 with privacy parameters (ϵ, δ), and the empirical average cost function $f_D(\mathbf{w})$ represented as in (3). Then Algorithm 1 computes an (ϵ, δ) differentially private approximation $\hat{f}_D(\mathbf{w})$ to $f_D(\mathbf{w})$. Consequently, the minimizer $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$ satisfies (ϵ, δ)-differential privacy.*

Algorithm 1 Gaussian FM

```
Require: Data samples $(\mathbf{x}_n, y_n)$ for $n \in [N]$; cost function $f_D(\mathbf{w})$ represented as in (3);
    privacy parameters $(\epsilon, \delta)$.
for $0 \le j \le J$ do
    Compute $\Lambda_j$ as shown in Section 4
    Compute $\Delta_j = \max_{\mathbb{D},\mathbb{D}'} \|\Lambda_j^{\mathbb{D}} - \Lambda_j^{\mathbb{D}'}\|_2$
    Compute $\tau_j = \frac{\Delta_j}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$
    Compute $e_j \sim \mathcal{N}(0, \tau_j^2)$ with the same dimension as $\Lambda_j$
    Release $\hat{\Lambda}_j = \Lambda_j + e_j$
end for
Compute $\hat{f}_D(\mathbf{w}) = \sum_{j=0}^{J}\langle\hat{\Lambda}_j, \bar{\phi}_j\rangle$
return Perturbed objective function $\hat{f}_D(\mathbf{w})$
```

Proof. The proof of Theorem 2 follows from the fact that the function $\hat{f}_D(\mathbf{w})$ depends on the data samples only through $\{\hat{\Lambda}_j\}$. The computation of $\{\hat{\Lambda}_j\}$ is (ϵ, δ)-differentially private by the Gaussian mechanism [4,33]. Therefore, the release of $\hat{f}_D(\mathbf{w})$ satisfies (ϵ, δ)-differential privacy. One way to rationalize this is to consider that the probability of the event of selecting a particular set of $\{\hat{\Lambda}_j\}$ is the same as the event of formulating a function $\hat{f}_D(\mathbf{w})$ with that set of $\{\hat{\Lambda}_j\}$. Therefore, it suffices to consider the joint density of the $\{\hat{\Lambda}_j\}$ and find an upper bound on the ratio of the joint densities of the $\{\hat{\Lambda}_j\}$ under two neighboring datasets $\mathbb{D}$ and $\mathbb{D}'$. As we employ the Gaussian mechanism to compute $\{\hat{\Lambda}_j\}$, the ratio is upper bounded by $\exp(\epsilon)$ with probability at least $1 - \delta$. Therefore, the release of $\hat{f}_D(\mathbf{w})$ satisfies (ϵ, δ)-differential privacy. Furthermore, differential privacy is post-processing invariant. Therefore, the computation of the minimizer $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$ also satisfies (ϵ, δ)-differential privacy. □

Privacy Analysis of Noisy Gradient Descent [12] using Rényi Differential Privacy. One of the most crucial qualitative properties of DP is that it allows us to evaluate the cumulative privacy loss over multiple computations [33]. Cumulative, or total, privacy loss is different from (ϵ, δ)-DP in multi-round machine learning algorithms. In order to demonstrate the superior privacy guarantee of the proposed Gaussian FM, we compare it to the existing functional mechanism [7], the relaxed functional mechanism [10], the objective perturbation [8], and the noisy gradient descent [12] method. Note that, similar to objective perturbation, FM and relaxed FM, the proposed Gaussian FM injects randomness in a single round, and therefore does not require privacy accounting. However, the noisy gradient descent method involves addition of noise in each step the gradient is computed. That is, noise is added to the computed gradients of the parameters of the objective function during optimization. Since it is a multi-round algorithm, the overall ϵ used during optimization is different from the ϵ for every iteration. We follow the analysis procedure outlined in [6] for the privacy accounting of the noisy gradient descent algorithm. Note that Proposition 3 described in Section 2.1 is defined for functions with unit $\mathcal{L}_2$-sensitivity. Therefore, if a noise from $N(0, \tau^2)$ is added to a function with sensitivity Δ, then the resulting mechanism satisfies $(\alpha, \frac{\alpha}{2\frac{\tau^2}{\Delta^2}})$-RDP. Now, according to Proposition 3, the T-fold composition of Gaussian mechanisms satisfies $(\alpha, \frac{\alpha T}{2\frac{\tau^2}{\Delta^2}})$-RDP. Finally, according to Proposition 1, it also satisfies $(\epsilon_r + \frac{\log\frac{1}{\delta_r}}{\alpha - 1}, \delta_r)$-differential privacy for any $0 \leq \delta_r \leq 1$, where $\epsilon_r = \frac{\alpha T}{2\frac{\tau^2}{\Delta^2}}$. For a given value of δ_r, we can express the value of the optimal overall ϵ_{opt} as a function of α_{opt}:

$$\epsilon_{\text{opt}} = \frac{\alpha_{\text{opt}} T}{2\frac{\tau^2}{\Delta^2}} + \frac{\log\frac{1}{\delta_r}}{\alpha_{\text{opt}} - 1}, \tag{7}$$

where α_{opt} is given by

$$\alpha_{\text{opt}} = 1 + \sqrt{\frac{2}{T}\frac{\tau^2}{\Delta^2}\log\frac{1}{\delta_r}}. \tag{8}$$

We compute the overall ϵ following this procedure for the noisy gradient descent algorithm [12] in our experiments in Section 6.

4. Application of Gaussian FM in Regression Analysis

In this section, we demonstrate how our proposed Gaussian FM can be applied to linear and logistic regression problems to achieve (ϵ, δ)-DP. For both cases, we first decompose the objective function (i.e., the empirical average cost function) into a finite series of polynomials, inject noise into the coefficients (i.e., the only data-dependent components in the decomposition) using Gaussian mechanism, and finally minimize the (ϵ, δ)-differentially private objective function. As before, we assume that we have a dataset $\mathbb{D}$ with N samples of the form $(\mathbf{x}_n, y_n)$, where for each sample $n \in [N]$, the D-dimensional feature vector is

$\mathbf{x}_n = \begin{bmatrix} x_{n1} \; x_{n2} \; \cdots \; x_{nD} \end{bmatrix}^\top$ (normalized to ensure $\|\mathbf{x}_n\|_2 \leq 1$) and the corresponding output is y_n.

4.1. Linear Regression

For our linear regression problem, we assume $y_n \in [-1, 1]$. Let $\mathbf{w} \in \mathbb{R}^D$ be the parameter vector. The goal of linear regression is to find the optimal $\mathbf{w}^*$ so that $\mathbf{x}_n^\top \mathbf{w}^* \approx y_n$. The empirical average cost function is defined as

$$f_D(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \mathbf{x}_n^\top \mathbf{w} \right)^2. \tag{9}$$

Using simple algebra, this equation can be decomposed into a series of polynomials as

$$f_D(\mathbf{w}) = \left(\frac{1}{N} \sum_{n=1}^{N} y_n^2 \right) + \sum_{d=1}^{D} \left(-\frac{2}{N} \sum_{n=1}^{N} y_n x_{nd} \right) w_d + \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} \left(\frac{1}{N} \sum_{n=1}^{N} x_{nd_1} x_{nd_2} \right) w_{d_1} w_{d_2}.$$

As we intend to compute the differentially private minimizer $\hat{\mathbf{w}}^*$, we observe that the representation of $f_D(\mathbf{w})$ is of the form $f_D(\mathbf{w}) = \sum_{j=0}^{J} \langle \Lambda_j, \bar{\phi}_j \rangle$ with $J = 2$. The expressions for Λ_j are

$$\begin{aligned}
\Lambda_0 &= \frac{1}{N} \sum_{n=1}^{N} y_n^2, \\
\Lambda_1 &= -\frac{2}{N} \begin{bmatrix} \sum_{n=1}^{N} y_n x_{n1} \\ \sum_{n=1}^{N} y_n x_{n2} \\ \vdots \\ \sum_{n=1}^{N} y_n x_{nD} \end{bmatrix}, \\
\Lambda_2 &= \frac{1}{N} \begin{bmatrix} \sum_{n=1}^{N} x_{n1}^2 & \cdots & \sum_{n=1}^{N} x_{n1} x_{nD} \\ \vdots & \ddots & \vdots \\ \sum_{n=1}^{N} x_{nD} x_{n1} & \cdots & \sum_{n=1}^{N} x_{nD}^2 \end{bmatrix} = \frac{1}{N} \left(\mathbf{X} \mathbf{X}^\top \right).
\end{aligned}$$

Here, Λ_0 is a scalar, Λ_1 is a D-dimensional vector, and Λ_2 is a $D \times D$ symmetric matrix, since $\mathbf{X}$ is an $D \times N$ matrix containing $\mathbf{x}_n$ as its columns. The expressions for $\bar{\phi}_j$ are

$$\begin{aligned}
\bar{\phi}_0 &= 1, \\
\bar{\phi}_1 &= \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}, \\
\bar{\phi}_2 &= \begin{bmatrix} w_1^2 & w_1 w_2 & \cdots & w_1 w_D \\ w_2 w_1 & w_2^2 & \cdots & w_2 w_D \\ \vdots & \vdots & \ddots & \vdots \\ w_D w_1 & w_D w_2 & \cdots & w_D^2 \end{bmatrix}.
\end{aligned}$$

The next step is finding the sensitivities of $\{\Lambda_j\}$ using (4). Let $\mathbb{D}$ and $\mathbb{D}'$ be two neighboring datasets differing in only one sample, e.g., the last samples $(\mathbf{x}_N, y_N)$ and $(\mathbf{x}'_N, y'_N)$. Now, the $\mathcal{L}_2$-sensitivity of Λ_0 is

$$\begin{aligned}\Delta_0 &= \max_{\mathbb{D},\mathbb{D}'} \left\| \frac{1}{N}\sum_{n=1}^{N} y_n^2 - \frac{1}{N}\sum_{n=1}^{N} y_n'^2 \right\|_2 \\ &= \frac{1}{N}\max_{\mathbb{D},\mathbb{D}'} \|y_N^2 - y_N'^2\|_2 \\ &\le \frac{1}{N},\end{aligned}$$

since $y_n \in [-1,1]$ and hence $y_n^2 \in [0,1]$. Next, the $\mathcal{L}_2$-sensitivity of Λ_1 is

$$\begin{aligned}\Delta_1 &= \max_{\mathbb{D},\mathbb{D}'} \left\| -\frac{2}{N} y_N \mathbf{x}_N + \frac{2}{N} y_N' \mathbf{x}_N' \right\|_2 \\ &\le \frac{2}{N}\max_{\mathbb{D},\mathbb{D}'} \left(\|y_N\mathbf{x}_N\|_2 + \|y_N'\mathbf{x}_N'\|_2 \right) \\ &= \frac{2}{N}\max_{\mathbb{D},\mathbb{D}'} \left(|y_N|\|\mathbf{x}_N\|_2 + |y_N'|\|\mathbf{x}_N'\|_2 \right) \\ &\le \frac{4}{N},\end{aligned}$$

where the second line follows from the triangle inequality, and the last line follows from the assumptions that $y_n \in [-1,1]$ and $\|\mathbf{x}_n\|_2 \le 1$. Finally, the $\mathcal{L}_2$-sensitivity of Λ_2 is

$$\begin{aligned}\Delta_2 &= \max_{\mathbb{D},\mathbb{D}'} \left\| \frac{1}{N}(\mathbf{X}\mathbf{X}^\top) - \frac{1}{N}(\mathbf{X}'\mathbf{X}'^\top) \right\|_2 \\ &= \frac{1}{N}\max_{\mathbb{D},\mathbb{D}'} \|\mathbf{x}_N\mathbf{x}_N^\top - \mathbf{x}_N'\mathbf{x}_N'^\top\|_2 \\ &\le \frac{1}{N}.\end{aligned}$$

The proof of the inequality in the last line is as follows:

Proof. The term $\left(\mathbf{x}_N\mathbf{x}_N^\top - \mathbf{x}_N'\mathbf{x}_N'^\top\right)$ is a $D \times D$ symmetric matrix, whose norm can be expressed [41] as $\sup\left\{ \mathbf{u}^\top [\mathbf{x}_N\mathbf{x}_N^\top - \mathbf{x}_N'\mathbf{x}_N'^\top] \mathbf{v} \,\middle|\, u = v,\ \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1 \right\}$. It follows that

$$\begin{aligned}\|\mathbf{x}_N\mathbf{x}_N^\top - \mathbf{x}_N'\mathbf{x}_N'^\top\|_2 &= \sup\left\{ \mathbf{u}^\top\mathbf{x}_N\mathbf{x}_N^\top\mathbf{u} - \mathbf{u}^\top\mathbf{x}_N'\mathbf{x}_N'^\top\mathbf{u} \right\} \\ &= \sup\left\{ \left(\mathbf{x}_N^\top\mathbf{u}\right)^\top\left(\mathbf{x}_N^\top\mathbf{u}\right) - \left(\mathbf{x}_N'^\top\mathbf{u}\right)^\top\left(\mathbf{x}_N'^\top\mathbf{u}\right) \right\} \\ &= \sup\left\{ \|\mathbf{x}_N^\top\mathbf{u}\|_2^2 - \|\mathbf{x}_N'^\top\mathbf{u}\|_2^2 \right\} \\ &\le \sup\left\{ \|\mathbf{x}_N^\top\|_2^2\|\mathbf{u}\|_2^2 - \|\mathbf{x}_N'^\top\|_2^2\|\mathbf{u}\|_2^2 \right\} \le 1.\end{aligned}$$

□

After computing the $\mathcal{L}_2$-sensitivity of Λ_j for $j = 0, 1$, and 2, we can now compute the noise array $e_j \sim \mathcal{N}(0, \tau_j^2)$, where $\tau_j = \frac{\Delta_j}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$, and then compute $\{\hat{\Lambda}_j\}$ following (5). Using these, we can compute the (ϵ,δ) differentially private $\hat{f}_D(\mathbf{w})$ according to (6), and consequently, the minimizer $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$. Note that, unlike the existing FM and relaxed FM, the additive noise variances of our proposed Gaussian FM do not depend on the sample dimension D. More specifically, for the linear regression problem, the $\mathcal{L}_1$-sensitivity of the coefficients in FM [7] is $\Delta^{fm} = \frac{2}{N}(1+D)^2$ and the $\mathcal{L}_2$-sensitivity of the coefficients in relaxed FM [10] is $\Delta^{rlx\text{-}fm} = \frac{2}{N}\sqrt{1+4D+D^2}$ (see Appendix A for the proof).

Both of these sensitivities are orders of magnitude larger than Δ_j that we achieved for $j \in \{0, 1, 2\}$, and for practical values of D and N. Thus, the proposed Gaussian FM can offer the (ϵ, δ)-differentially private approximation $\hat{f}_D(\mathbf{w})$ with much less noise, which results in a (ϵ, δ)-differentially private model $\hat{\mathbf{w}}^*$ that is much closer to the true model $\mathbf{w}^*$. We show empirical validation on synthetic and real datasets in Section 6.

4.2. Logistic Regression

For the logistic regression problem, we assume $y_n \in \{0, 1\}$ to be the class labels. The class label is approximated using the *sigmoid* function defined as $f_{sig}(z) = \frac{1}{1+\exp(-z)}$. Let $\mathbf{w} \in \mathbb{R}^D$ be the parameter vector. The goal of logistic regression is to find the optimal $\mathbf{w}^*$ so that $f_{sig}(\mathbf{x}_n^\top \mathbf{w}^*) \approx y_n$. The empirical average cost function for logistic regression is defined as

$$\begin{aligned} f_D(\mathbf{w}) &= -\frac{1}{N}\sum_{n=1}^{N} y_n \log\left(f_{sig}(\mathbf{x}_n^\top \mathbf{w})\right) + (1-y_n)\log\left(1 - f_{sig}(\mathbf{x}_n^\top \mathbf{w})\right) \\ &= \frac{1}{N}\sum_{n=1}^{N} \log\left(1+\exp(\mathbf{x}_n^\top \mathbf{w})\right) - y_n \mathbf{x}_n^\top \mathbf{w}. \end{aligned} \tag{10}$$

Unlike linear regression, the simplified form of $f_D(\mathbf{w})$ in the second line cannot be represented with a finite series of polynomials. Zhang et al. [7] proposed an *approximate* polynomial form of $f_D(\mathbf{w})$ using Taylor series expansion, written as

$$\tilde{f}_D(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=0}^{2} \frac{f_1^{(k)}(0)}{k!}\left(\mathbf{x}_n^\top \mathbf{w}\right)^k - \frac{1}{N}\sum_{n=1}^{N} y_n \mathbf{x}_n^\top \mathbf{w}.$$

Using simple algebra and the values of $f_1^{(k)}(0)$ for $k = 0, 1$, and 2, i.e., $f_1^{(0)}(0) = \log 2$, $f_1^{(1)}(0) = \frac{1}{2}$, and $f_1^{(k)}(0) = \frac{1}{4}$, we obtain

$$\tilde{f}_D(\mathbf{w}) = \log 2 + \sum_{d=1}^{D}\left(\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2} - y_n\right)x_{nd}\right)w_d + \sum_{d_1=1}^{D}\sum_{d_2=1}^{D}\left(\frac{1}{8N}\sum_{n=1}^{N} x_{nd_1}x_{nd_2}\right)w_{d_1}w_{d_2}.$$

As before, we intend to compute the differentially private minimizer $\hat{\mathbf{w}}^*$, and we observe that the representation of $\tilde{f}_D(\mathbf{w})$ is of the form $f_D(\mathbf{w}) = \sum_{j=0}^{J}\langle \Lambda_j, \bar{\phi}_j \rangle$ with $J = 2$. The expressions for Λ_j are

$$\begin{aligned} \Lambda_0 &= \log 2, \\ \Lambda_1 &= \frac{1}{N}\begin{bmatrix} \sum_{n=1}^{N}\left(\frac{1}{2} - y_n\right)x_{n1} \\ \sum_{n=1}^{N}\left(\frac{1}{2} - y_n\right)x_{n2} \\ \vdots \\ \sum_{n=1}^{N}\left(\frac{1}{2} - y_n\right)x_{nD} \end{bmatrix}, \\ \Lambda_2 &= \frac{1}{8N}\begin{bmatrix} \sum_{n=1}^{N} x_{n1}^2 & \cdots & \sum_{n=1}^{N} x_{n1}x_{nD} \\ \vdots & \ddots & \vdots \\ \sum_{n=1}^{N} x_{nD}x_{n1} & \cdots & \sum_{n=1}^{N} x_{nD}^2 \end{bmatrix} = \frac{1}{8N}\left(\mathbf{X}\mathbf{X}^\top\right). \end{aligned}$$

Again, Λ_j is a scalar, a D-dimensional vector, and a $D \times D$ matrix for $j = 0, 1$, and 2, respectively. We can express $\bar{\phi}_j$ for $j = 0, 1$, and 2 the same way as we did for linear regression in Section 4.1. To compute the sensitivities of $\{\Lambda_j\}$ using (4), let $\mathbb{D}$ and $\mathbb{D}'$ be two neighboring datasets differing in only the last samples, which are $(\mathbf{x}_N, y_N)$ and $(\mathbf{x}'_N, y'_N)$, respectively. Now, the $\mathcal{L}_2$-sensitivity of Λ_0 is $\Delta_0 = \max_{\mathbb{D},\mathbb{D}'}\left\| \log 2 - \log 2 \right\|_2 = 0$. The $\mathcal{L}_2$-sensitivity of Λ_1 is

$$\begin{aligned}\Delta_1 &= \max_{\mathbb{D},\mathbb{D}'} \left\| \frac{1}{N}\left(\frac{1}{2} - y_N\right)\mathbf{x}_N - \frac{1}{N}\left(\frac{1}{2} - y'_N\right)\mathbf{x}'_N \right\|_2 \\ &\leq \frac{1}{N} \max_{\mathbb{D},\mathbb{D}'} \left(\left|\frac{1}{2} - y_N\right| \left\|\mathbf{x}_N\right\|_2 + \left|\frac{1}{2} - y'_N\right| \left\|\mathbf{x}'_N\right\|_2 \right) \\ &\leq \frac{1}{N},\end{aligned}$$

where $\left|\frac{1}{2} - y_N\right| \leq \frac{1}{2}$, since $y_n \in \{0, 1\}$, and $\|\mathbf{x}_n\|_2 \leq 1$. Finally, the $\mathcal{L}_2$-sensitivity of Λ_2 is

$$\begin{aligned}\Delta_2 &= \max_{\mathbb{D},\mathbb{D}'} \left\| \frac{1}{8N}\left(\mathbf{X}\mathbf{X}^\top\right) - \frac{1}{8N}\left(\mathbf{X}'\mathbf{X}'^\top\right) \right\|_2 \\ &= \frac{1}{8N} \max_{\mathbb{D},\mathbb{D}'} \|\mathbf{x}_N\mathbf{x}_N^\top - \mathbf{x}'_N\mathbf{x}'^\top_N\|_2 \\ &\leq \frac{1}{8N},\end{aligned}$$

where the inequality follows from the expression for the norm of a symmetric matrix, as shown in Section 4.1. After computing the $\mathcal{L}_2$-sensitivity of Λ_j for $j = 0, 1$, and 2, we can now compute the noise array $e_j \sim \mathcal{N}(0, \tau_j^2)$, where $\tau_j = \frac{\Delta_j}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$, and then compute $\{\hat{\Lambda}_j\}$ following (5). Using these, we can compute the (ϵ, δ) differentially-private $\hat{f}_D(\mathbf{w})$ according to (6), and consequently, the minimizer $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$. Again we note that the $\mathcal{L}_1$-sensitivity of the coefficients in FM [7] is $\Delta^{fm} = \frac{1}{N}\left(\frac{D^2}{4} + 3D\right)$ and the $\mathcal{L}_2$-sensitivity of the coefficients in relaxed FM [10] is $\Delta^{rlx\text{-}fm} = \frac{1}{N}\sqrt{\frac{D^2}{16} + D}$ for logistic regression. As in the case of linear regression, both of these sensitivities are orders of magnitude larger than Δ_j that we achieved for $j \in \{1, 2\}$, and for practical values of D and N. Since additive noise variances of our proposed Gaussian FM do not depend on the sample dimension D, we obtain $\hat{f}_D(\mathbf{w})$, the (ϵ, δ)-differentially private approximation to $\tilde{f}_D(\mathbf{w})$, with much less noise. As mentioned before, we validate our analysis empirically using synthetic and real datasets in Section 6.

4.3. Avoiding Unbounded Noisy Objective Functions

Our proposed Gaussian FM achieves (ϵ,δ)-DP by injecting noise drawn from a Gaussian distribution into the coefficients of the Stone–Weierstrass decomposition of the empirical average objective function. However, the injection of noise may render the objective function *unbounded*, which means there may not exist any optimal solution for the noisy objective function. As shown in Sections 4.1 and 4.2, the Stone–Weierstrass decomposition would transform the objective functions of linear and logistic regression problems into quadratic polynomials in our Gaussian FM. Let $\hat{f}_D(\mathbf{w}) = \mathbf{w}^\top\mathbf{M}\mathbf{w} + \alpha^\top\mathbf{w} + \beta$ be the matrix representation of the quadratic polynomial, where $\mathbf{M}$ is a symmetric and positive semi-definite matrix, α is a D-dimensional vector and β is a scalar. After injection of noise, the noisy objective function becomes $\hat{f}_D(\mathbf{w}) = \mathbf{w}^\top\hat{\mathbf{M}}\mathbf{w} + \hat{\alpha}^\top\mathbf{w} + \hat{\beta}$. In order to ensure that $\hat{f}_D(\mathbf{w})$ is bounded after introducing noise, it suffices to make sure $\hat{\mathbf{M}}$ is also symmetric and positive semi-definite [42].

We follow the seminal work of Dwork et al. [43] in our implementation—the symmetry of $\hat{\mathbf{M}}$ is ensured by constructing the noise matrix in such a way that noise is first drawn from the Gaussian distribution to form an upper triangular matrix, and the elements of the upper triangle part of the matrix (excluding the diagonal elements) are then copied to its lower triangle part. Adding the symmetric noise matrix to $\mathbf{M}$ results in a symmetric $\hat{\mathbf{M}}$. However, $\hat{f}_D(\mathbf{w})$ may still be unbounded if $\hat{\mathbf{M}}$ is not positive semi-definite. To resolve this, we perform eigen-decomposition of $\hat{\mathbf{M}}$ to obtain the eigenvalues and corresponding eigenvectors. We then project the eigenvalues onto the non-negative orthant. Let $\mathbf{Q}^\top\mathbf{S}\mathbf{Q}$ be the eigen-decomposition of $\hat{\mathbf{M}}$, where $\mathbf{Q}$ is a $D \times D$ matrix containing an eigenvector of $\hat{\mathbf{M}}$

in each row, and $\mathbf{S}$ is a diagonal matrix where the i-th diagonal element is the eigenvalue of $\hat{\mathbf{M}}$ corresponding to the eigenvector in the i-th row of $\mathbf{Q}$. We can write

$$\hat{f}_D(\mathbf{w}) = \mathbf{w}^\top(\mathbf{Q}^\top\mathbf{S}\mathbf{Q})\mathbf{w} + \hat{\alpha}^\top\mathbf{w} + \hat{\beta}.$$

If the i-th diagonal element of $\mathbf{S}$ is negative, we turn that entry to zero. After this projection onto the non-negative orthant, let the resulting matrix be $\hat{\mathbf{S}}$, where any i-th diagonal element is bigger than or equal to zero. The noisy objective function then becomes

$$\hat{f}_D(\mathbf{w}) = \mathbf{w}^\top(\mathbf{Q}^\top\hat{\mathbf{S}}\mathbf{Q})\mathbf{w} + \hat{\alpha}^\top\mathbf{w} + \hat{\beta},$$

where $(\mathbf{Q}^\top\hat{\mathbf{S}}\mathbf{Q})$ is symmetric positive semi-definite. Thus, $\hat{f}_D(\mathbf{w})$ is bounded. Since all of these are performed after the differentially-private noise addition, we can invoke the post-processing invariability of differential privacy and guarantee that $\hat{f}_D(\mathbf{w})$ is (ϵ,δ)-differentially private. Consequently, the minimizer $\hat{\mathbf{w}}^*$ also satisfies (ϵ,δ) differential privacy. Note that it is possible for all the eigenvalues of the differentially private estimate of the $\mathbf{M}$ matrix to be negative. We leave the solution to such cases for future work.

5. Extension of Gaussian FM to Decentralized-Data Setting: capeFM

In many signal processing and machine learning applications, the privacy-sensitive user data being collected/used are of decentralized nature. Training machine learning and neural-network-based models on such a huge amount of data is certainly lucrative from an algorithmic perspective, but privacy constraints often make it challenging to share such datasets with a central aggregator. However, training locally at one node/site is infeasible due to the number of samples in each node/site could be too small for meaningful model training. Decentralized DP can benefit such research work by allowing data owners to share information while maintaining local privacy. The conventional decentralized DP scheme, however, always results in a degradation in performance compared to that of the pooled-data scenario. In this section, we first describe the problem with conventional decentralized DP. Then we review the CAPE scheme [6] in brief, as we employ the CAPE scheme into our Gaussian FM to propose capeFM.

The Decentralized-data Setting. In line with our discussions in Section 2.2, let us consider a decentralized data setting with S sites and a central aggregator node. We assume an "honest but curious" threat model [6]: all parties follow the protocol honestly, but a subset are "curious" and can collude (maybe with an external adversary) to learn other sites' data/function outputs. For simplicity, we consider the symmetric setting: each site $s \in [S]$ holds a dataset $\mathbb{D}_s$ of $N_s = \frac{N}{S}$ disjoint data samples $(\mathbf{x}_{s,n}, y_{s,n})$, where the total number of samples across all sites is N, and $\mathbf{x}_{s,n} \in \mathbb{R}^D$. The cost incurred by the model parameters $\mathbf{w} \in \mathbb{R}^D$ due to one data sample is $f(\mathbf{x}_{s,n};\mathbf{w}) : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}$. We need to minimize the average cost to find the optimal $\mathbf{w}^*$. The empirical average cost for a particular $\mathbf{w}$ over all the samples is expressed as

$$f_D(\mathbf{w}) = \frac{1}{N}\sum_{s=1}^{S}\sum_{n=1}^{N_s} f(\mathbf{x}_{s,n};\mathbf{w}) = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{N_s}\sum_{n=1}^{N_s} f(\mathbf{x}_{s,n};\mathbf{w}).$$

According to (3), the above expression can be written as

$$f_D(\mathbf{w}) = \frac{1}{S}\sum_{s=1}^{S}\sum_{j=0}^{J}\langle\Lambda_j^s, \bar{\phi}_j\rangle = \sum_{j=0}^{J}\langle\Lambda_j, \bar{\phi}_j\rangle,$$

where Λ_j^s contains $\frac{1}{N_s}\sum_{n=1}^{N_s}\lambda_{\phi s,n}$ as its entries for all $\phi(\mathbf{w}) \in \Phi_j$ at site s, $\Lambda_j = \frac{1}{S}\sum_{s=1}^{S}\Lambda_j^s$, and $\bar{\phi}_j$ is the array containing all $\phi(\mathbf{w}) \in \Phi_j$ as its entries. Finally, we can compute the minimizer:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f_D(\mathbf{w}) = \arg\min_{\mathbf{w}} \sum_{j=0}^{J} \langle \Lambda_j, \bar{\phi}_j \rangle.$$

5.1. Problems with Conventional Decentralized DP Computations

In this section, we discuss the problems with conventional decentralized DP schemes [6]. Consider estimating the mean $f(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} x_n$ of N scalars $\mathbf{x} = [x_1, \ldots, x_{N-1}, x_N]^\top$, where each $x_n \in [0,1]$. The $\mathcal{L}_2$-sensitivity of the function $f(\mathbf{x})$ is $\frac{1}{N}$. Therefore, for computing the (ϵ, δ)-DP estimate of the average $a = f(\mathbf{x})$, we can follow the Gaussian mechanism [4] to release $\hat{a}_{pool} = a + e_{pool}$, where $e_{pool} \sim \mathcal{N}(0, \tau_{pool}^2)$ and $\tau_{pool} = \frac{1}{N\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$.

Suppose now that the N samples are equally distributed among S sites. An aggregator wishes to estimate and publish the mean of all the samples. For preserving privacy, the conventional DP approach is for each site s to release (or send to the aggregator node) an (ϵ, δ)-DP estimate of the function $a_s = f(\mathbf{x}_s)$ as: $\hat{a}_s = f(\mathbf{x}_s) + e_s$, where $e_s \sim \mathcal{N}(0, \tau_s^2)$ and $\tau_s = \frac{1}{N_s\epsilon}\sqrt{2\log\frac{1.25}{\delta}} = \frac{S}{N\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$. The aggregator can then compute the (ϵ, δ)-DP approximate average as

$$\hat{a}_{conv} = \frac{1}{S}\sum_{s=1}^{S} \hat{a}_s = \frac{1}{S}\sum_{s=1}^{S} a_s + \frac{1}{S}\sum_{s=1}^{S} e_s.$$

The variance of the estimator $\hat{a}_{conv}$ is $S \cdot \frac{\tau_s^2}{S^2} = \frac{\tau_s^2}{S} \triangleq \tau_{conv}^2$. We observe the ratio

$$\frac{\tau_{pool}^2}{\tau_{conv}^2} = \frac{\tau_s^2/S^2}{\tau_s^2/S} = \frac{1}{S}.$$

That is, the decentralized DP averaging scheme will always result in a poorer performance than the pooled-data case. Imtiaz et al. [6] proposed the CAPE protocol that improves the performance of such systems by assuming the availability of some reasonable resources.

5.2. Correlation Assisted Private Estimation (CAPE)

Trust/Collusion Model. In order to incorporate the CAPE scheme to our proposed Gaussian FM in a decentralized data setting, we assume a similar trust model as in [6]. As mentioned before, we assume all of the S sites and the central aggregator node to be honest-but-curious. That is, the sites and central node can collude with an adversary to learn about the data or function output of some other site. We assume that up to $S_C = \lceil \frac{S}{3} \rceil - 1$ sites, as well as the central node can collude with an adversary. In addition to having access to the outputs from each site and the aggregator, the adversary can know everything about the S_C colluding sites, including their private data. Denoting the non-colluding sites with S_H, we have $S = S_C + S_H$.

Correlated Noise and the CAPE Protocol. Imtiaz et al. [6] proposed a novel framework that ensures (ϵ, δ)-DP guarantee of the output from each site, while achieving the same noise level of the pooled-data scenario in the final output from the aggregator. In the CAPE scheme, each site $s \in [S]$ first generates two noise terms: $g_s \sim \mathcal{N}(0, \tau_g^2)$ locally, and $e_s \sim \mathcal{N}(0, \tau_e^2)$ jointly with all other sites such that $\sum_{s=1}^{S} e_s = 0$. The correlated noise term e_s is generated by employing the secure aggregation protocol (SecureAgg) by Bonawitz et al. [28], which utilizes Shamir's t-out-of-n secret sharing [44] and is communication-efficient. The procedure is outlined in Algorithm 2.

Algorithm 2 Generate Zero-Sum Noise

```
Require: Local noise variances $\{\tau_s^2\}$; security parameter $\lambda$; threshold value $t$
Each site generates $\hat{e}_s \sim \mathcal{N}(0, \tau_s^2)$
Aggregator computes $\sum_{s=1}^{S} \hat{e}_s$ according to SecureAgg($\lambda, t$) [28]
Aggregator broadcasts $\sum_{s=1}^{S} \hat{e}_s$ to all sites $s \in [S]$
Each site computes $e_s = \hat{e}_s - \frac{1}{S}\sum_{s'=1}^{S} \hat{e}_{s'}$
return $e_s$
```

Note that neither of the terms e_s and g_s has large enough variance to provide an acceptable (ϵ, δ)-DP guarantee. However, the variances of e_s and g_s are chosen in such a way that the noise $e_s + g_s$ is sufficient to ensure a stringent DP guarantee to $f(\mathbf{x}_s)$ at site s. We observe that the variance of e_s is given by $\tau_e^2 = \left(1 - \frac{1}{S}\right)\tau_s^2$ and the variance of g_s is set to $\tau_g^2 = \frac{\tau_s^2}{S}$ [6]. Considering the decentralized mean computation problem of Section 5.1, under the CAPE scheme, each site sends $\hat{a}_s = f(\mathbf{x}_s) + e_s + g_s$ to the aggregator. We can then compute the following at the aggregator

$$a_{cape} = \frac{1}{S}\sum_{s=1}^{S} \hat{a}_s = \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}_s) + \frac{1}{S}\sum_{s=1}^{S} g_s,$$

where we used $\sum_{s=1}^{S} e_s = 0$. The variance of the estimator a_{cape} is $\tau_{cape}^2 = S \cdot \frac{\tau_g^2}{S^2} = \tau_{pool}^2$, which is exactly the same as if all the data were present at the aggregator. This claim is formalized in Lemma 1 [6] in Section 2.1. That is, the CAPE protocol achieves the same noise variance as the pooled-data scenario in the symmetric decentralized-data setting.

5.3. Proposed Gaussian FM for Decentralized Data (capeFM)

For employing the CAPE scheme to extend our proposed Gaussian FM for decentralized-data setting, we need to generate the zero-sum noise. We can readily extend Algorithm 2 to generate array-valued zero-sum noise terms for each of the Λ_j terms of the decomposition (3). That is, according to the CAPE scheme, the sites generate the noise e_j^s using Algorithm 2, such that $\sum_{s=1}^{S} e_j^s = 0$ holds for all $j \in \{0, \ldots, J\}$. The sites also generate noise g_j^s with entries i.i.d. $\sim \mathcal{N}(0, {\tau_{jg}^{s}}^{2})$. The sites then compute the perturbed coefficient arrays locally as $\hat{\Lambda}_j^s = \Lambda_j^s + e_j^s + g_j^s$ for all $j \in \{0, \ldots, J\}$ and send $\hat{\Lambda}_j^s$ to the central aggregator. Note that e_j^s and g_j^s are arrays of the same dimension as Λ_j^s. Now, the aggregator simply computes the average of each coefficient term for all $j \in \{0, \ldots, J\}$ as

$$\hat{\Lambda}_j = \frac{1}{S}\sum_{s=1}^{S} \hat{\Lambda}_j^s = \frac{1}{S}\sum_{s=1}^{S} \Lambda_j^s + \frac{1}{S}\sum_{s=1}^{S} g_j^s,$$

because $\sum_s e_j^s = 0$. The aggregator then uses these $\{\hat{\Lambda}_j\}$ to compute $\hat{f}_D(\mathbf{w}) = \sum_{j=0}^{J} \langle \hat{\Lambda}_j, \bar{\phi}_j \rangle$ and release $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$. The privacy of capeFM follows directly from Theorem 1 and Theorem 2. It follows from Lemma 1 [6] that in the symmetric setting (i.e., $N_s = \frac{N}{S}$ and $\tau_j^s = \tau_j$ for all sites $s \in [S]$ and all $j \in \{0, 1, \ldots, J\}$), the noise variance achieved at the aggregator is the same as that of the pooled-data scenario. Additionally, the performance gain of capeFM over any conventional decentralized functional mechanism is given by Proposition 4. We refer to our proposed decentralized functional mechanism as capeFM, shown in Algorithm 3.

Algorithm 3 Proposed Decentralized Gaussian FM (capeFM)

```
Require: Data samples $(\mathbf{x}_{s,n}, y_{s,n})$ for $s \in [S]$; cost function $f_D(\mathbf{w})$ as in (3); local noise
    variances $\{\tau_j^2\}$ for all $j \in \{0, \dots, J\}$
for $0 \le s \le S$ do
    for $0 \le j \le J$ do
        Compute $\Lambda_j^s$ as shown in Section 4
        Generate $e_j^s$ according to Algorithm 2 (entrywise)
        Compute ${\tau_{jg}^s}^2 = \frac{{\tau_j^s}^2}{S}$
        Generate $g_j^s$ with entries i.i.d. $\sim \mathcal{N}(0, {\tau_{jg}^s}^2)$
        Compute $\hat{\Lambda}_j^s = \Lambda_j^s + e_j^s + g_j^s$
    end for
end for
At the central aggregator, compute for all $j \in \{0, \dots, J\}$: $\hat{\Lambda}_j = \frac{1}{S}\sum_{s=1}^{S} \hat{\Lambda}_j^s$
Compute $\hat{f}_D(\mathbf{w}) = \sum_{j=0}^{J} \langle \hat{\Lambda}_j, \bar{\phi}_j \rangle$
return Perturbed objective function $\hat{f}_D(\mathbf{w})$
```

5.4. Computation and Communication Overhead of capeFM

We analyze the computation and communication costs associated with the proposed capeFM algorithm according to [6,28] for the decentralized linear regression and logistic regression problems. At each iteration round, we need to generate the zero-sum noise terms e_j^s, which entails $O(S + D^2)$ communication complexity of the sites and $O(S^2 + SD^2)$ communication complexity of the aggregator [28]. Each site computes the noisy coefficient arrays Λ_j^s and sends those to the aggregator, incurring an $O(D^2)$ communication cost for the sites. Therefore, the total communication cost is $O(S + D^2)$ for the sites and $O(S^2 + SD^2)$ for the aggregator node. On the other hand, the zero-sum noise generation entails $O(S^2 + SD^2)$ computation cost at the sites and $O(S^2D^2)$ computation cost at the aggregator [28]. This is expected since the largest coefficient arrays we are computing/sending are $D \times D$ matrices in the decentralized setting. Note that we are not incorporating the computation cost of $\hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}} \hat{f}_D(\mathbf{w})$.

6. Experimental Results

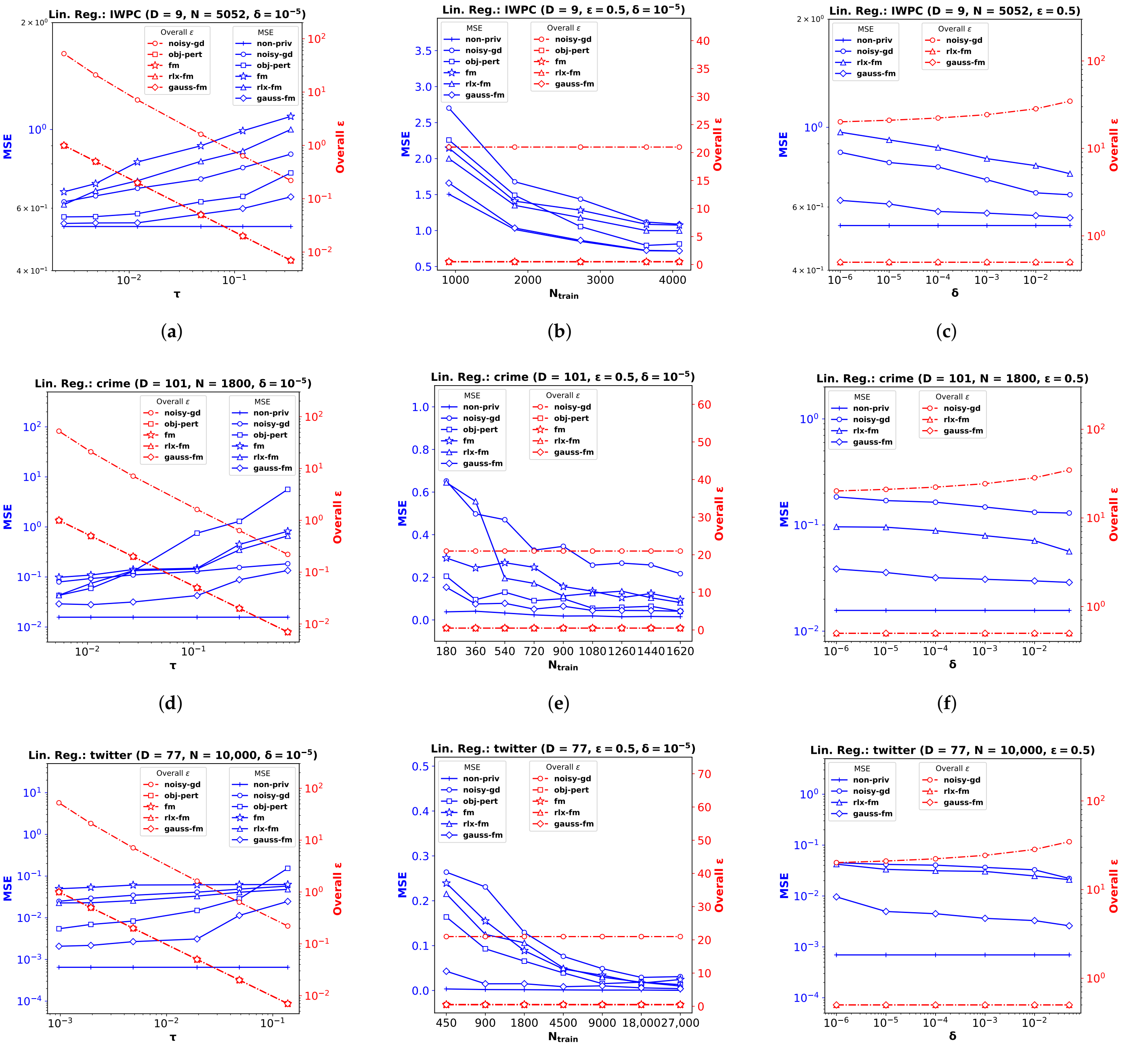
In this section, we empirically compare the performance of our proposed Gaussian FM algorithm (**gauss-fm**) with those of some state-of-the-art differentially private linear and logistic regression algorithms, namely noisy gradient descent (**noisy-gd**) [12], objective perturbation (**obj-pert**) [8], original functional mechanism (**fm**) [7], and relaxed functional mechanism (**rlx-fm**) [10]. We also compare the performance of these algorithms with non-private linear and logistic regression (**non-priv**). As mentioned before, we compute the overall ϵ using RDP for the multi-round **noisy-gd** algorithm. Additionally, we show how our proposed decentralized functional mechanism (**cape-fm**) can improve a decentralized computation if the target function has sensitivity satisfying the conditions of Proposition 5 in Section 2.1. We show the variation in performance with privacy parameters and number of training samples. For the decentralized setting, we further show the empirical performance comparison by varying the number of sites.

Performance Indices. For the linear regression task, we use the mean squared error (MSE) as the performance index. Let the test dataset be $\mathbb{D}_{\text{test}} = \{(\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y} : n \in [N_{\text{test}}]\}$. Then the MSE can be defined as: $\text{MSE} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} (\hat{y}_n - y_n)^2$, where $\hat{y}_n$ is the prediction from the algorithm. For the classification task, we use accuracy as the performance index. The accuracy can be defined as: $\text{Accuracy} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathcal{I}(\texttt{round}(\hat{y}_n) = y_n)$, where $\mathcal{I}(\cdot)$ is the indicator function, and $\hat{y}_n$ is the prediction from the algorithm. Note that, in addition to a small MSE or large accuracy, we want to attain a strict privacy guarantee, i.e., small overall (ϵ, δ) values. Recall from Section 3 that the overall ϵ for multi-shot algorithms is a function of the number of iterations, the target δ, the additive noise variance τ^2 and the $\mathcal{L}_2$ sensitivity Δ. To demonstrate the overall ϵ guarantee for a fixed target δ, we plotted the overall ϵ (with dotted red lines on the right y-axis) along with MSE/accuracy (with solid blue lines on the left y-axis) as a means for visualizing how the privacy–utility trade-off varies with different parameters. For a given privacy budget (or performance requirement), the user can use the overall ϵ plot on the right y-axis, shown with dotted lines, (or MSE/accuracy plot on the left y-axis, shown with solid lines) to find the required noise standard deviation τ on the x-axis and, thereby, find the corresponding performance (or overall ϵ). We compute the overall ϵ for the **noisy-gd** algorithm using the RDP technique shown in Section 3.

6.1. Linear Regression

For the linear regression problem, we perform experiments on three real datasets (and a synthetic dataset, as shown in Appendix B). The *pharmacogenetic* dataset was collected by the *International Warfarin Pharmacogenetics Consortium* (**IWPC**) [23] for the purpose of estimating personalized warfarin dose based on clinical and genotype information of a patient. The data used for this study have ambient dimension $D = 9$, and features are collected from $N = 5052$ patients. Out of the wide variety of numerical modeling methods used in [23], linear regression provided the most accurate dose estimates. Fredrikson et al. [20] later implemented an attack model assuming an adversary who employed an inference algorithm to discover the genotype of a target individual, and showed that an existing functional mechanism (**fm**) failed to provide a meaningful privacy guarantee to prevent such attacks. We perform privacy-preserving linear regression on the **IWPC** dataset (Figure 1a–c) to show the effectiveness of our proposed **gauss-fm** over **fm**, **rlx-fm**, and other existing approaches. Additionally, we use the *Communities and Crime* dataset (**crime**) [45], which has a larger dimensionality $D = 101$ (Figure 1d–f), and the *Buzz in Social Media* dataset (**twitter**) [46] with $D = 77$ and a large sample size $N = 10,000$ (Figure 1g–i). We refer the reader to [47] for a detailed description of these real datasets. For all the experiments, we pre-process the data so that the samples satisfy the assumptions $\|\mathbf{x}_n\|_2 \leq 1$ and $y_n \in [-1, 1]$ $\forall\ n \in [N]$. We divide each dataset into train and test partitions with a ratio of 90:10. We show the average performance over 10 independent runs.

Performance Comparison with Varying τ. We first investigate the variation of MSE with the DP additive noise standard deviation τ. We plot MSE against τ in Figure 1a,d,g. Recall from Definition 3 that, in the Gaussian mechanism, the noise is drawn from a Gaussian distribution with standard deviation $\tau = \frac{\Delta}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$. We keep δ fixed at 10^{-5}. Note that one can vary ϵ to vary τ. Since noise standard deviation is inversely proportional to ϵ, increasing ϵ means decreasing τ, i.e., smaller noise variance. We observe from the plots that smaller τ leads to smaller MSE for all DP algorithms, indicating better utility at the expense of higher privacy loss. It is evident from these MSE vs. τ plots that our proposed method **gauss-fm** has much smaller MSE compared to all the other methods for the same τ values for all datasets. The **obj-pert** and **fm** algorithms offer pure DP by trading off utility, whereas **gauss-fm** and **rlx-fm** algorithms offer approximate DP. Although **rlx-fm** improves upon **fm**, the excess noise due to linear dependence on data dimension D leads to higher MSE than **gauss-fm**. Our proposed **gauss-fm** outperforms all of these methods by reducing the additive noise with the novel sensitivity analysis as shown in Section 4. We recall that the overall privacy loss for **noisy-gd** is calculated using the RDP approach, since noise is injected into the gradients in every iteration during optimization, with target $\delta = 10^{-5}$. On the other hand, **gauss-fm**, **rlx-fm**, and **fm** add noise to the polynomial coefficients of the cost function $f_D(\mathbf{w})$ before optimization, and **obj-pert** injects noise into the regularized cost function [8]. We plot the total privacy loss for all of the algorithms against τ. We observe from the y-axis on the right that the total privacy loss of the multi-round **noisy-gd** is considerably higher than the single-shot algorithms.



Lin. Reg.: IWPC (D = 9, N = 5052, δ = 10^{-5})

Lin. Reg.: IWPC (D = 9, ε = 0.5, δ = 10^{-5})

Lin. Reg.: IWPC (D = 9, N = 5052, ε = 0.5)

(**a**) (**b**) (**c**)

Lin. Reg.: crime (D = 101, N = 1800, δ = 10^{-5})

Lin. Reg.: crime (D = 101, ε = 0.5, δ = 10^{-5})

Lin. Reg.: crime (D = 101, N = 1800, ε = 0.5)

(**d**) (**e**) (**f**)

Lin. Reg.: twitter (D = 77, N = 10,000, δ = 10^{-5})

Lin. Reg.: twitter (D = 77, ε = 0.5, δ = 10^{-5})

Lin. Reg.: twitter (D = 77, N = 10,000, ε = 0.5)

(**g**) (**h**) (**i**)

Figure 1. Linear regression performance comparison in terms of MSE and overall ϵ for *IWPC* $(D = 9)$, *crime* $(D = 101)$, and *twitter* $(D = 77)$ datasets with varying noise standard deviation τ in (**a**,**d**,**g**) the number of training samples N_{train} in (**b**,**e**,**h**), and privacy parameter δ in (**c**,**f**,**i**).

Performance Comparison with Varying N_{train}**.** Next, we investigate the variation of MSE with the number of training samples N_{train}. For this task, we shuffle and divide the total number of samples N into smaller partitions and perform the same pre-processing steps, while keeping the test partition untouched. We kept the values of the privacy parameters fixed: $\epsilon = 0.5$ and $\delta = 10^{-5}$. We plot MSE against N_{train} in Figure 1b,e,h. We observe that performance generally improves with the increase in N_{train}, which indicates that it is easier to ensure the same level of privacy when the training dataset cardinality is higher. We also observe from the MSE vs. N_{train} plots that our proposed method **gauss-fm** offers MSE very close to that of **non-priv** even for moderate sample sizes, outperforming **fm**, **rlx-fm**, **noisy-gd**, and **obj-pert**. Again, we compute the overall ϵ spent using RDP for **noisy-gd**, and show that the multi-round algorithm suffers from larger privacy loss. Recall from (7) in Section 3 that the overall ϵ depends on sensitivity Δ, and the number of iterations T. In the computation of $\frac{\tau^2}{\Delta^2}$, the number of training samples N_{train} is cancelled out. Thus, the overall ϵ depends only on T for **noisy-gd**. We keep T fixed at 1000 iterations for **noisy-gd** and observe that the overall privacy risk exceeds 20. Note that we set the value of the target δ_r in (7) to be equal to δ in our computations.

Performance Comparison with Varying δ. Recall that we can interpret the privacy parameter δ as the probability that an algorithm fails to provide privacy risk ϵ. The **obj-pert** and **fm** algorithms offer pure ϵ-DP, where the additional privacy parameter δ is zero. Hence, we compare our proposed **gauss-fm** method with the **rlx-fm** and **noisy-gd** methods, which also guarantee (ϵ,δ)-DP. In the Gaussian mechanism, δ is in the denominator of the logarithmic term within the square root in the expression of τ. Therefore, the noise variance τ^2 is not significantly changed by varying δ. We keep privacy parameter ϵ fixed at 0.5 and observe from the MSE vs. δ plots in Figure 1c,f,i show that the performance of our algorithm does not degrade much for smaller δ. For the **IWPC** dataset in Figure 1c, for a value of δ as small as 10^{-2} (indicating 1% probability of the algorithm failing to provide ϵ-differential privacy), the MSE of **gauss-fm** is almost the same as that of the **non-priv** case. For the other datasets, our proposed method also gives better performance and overall ϵ, and thus a better privacy–utility trade-off than **rlx-fm** and **noisy-gd**.

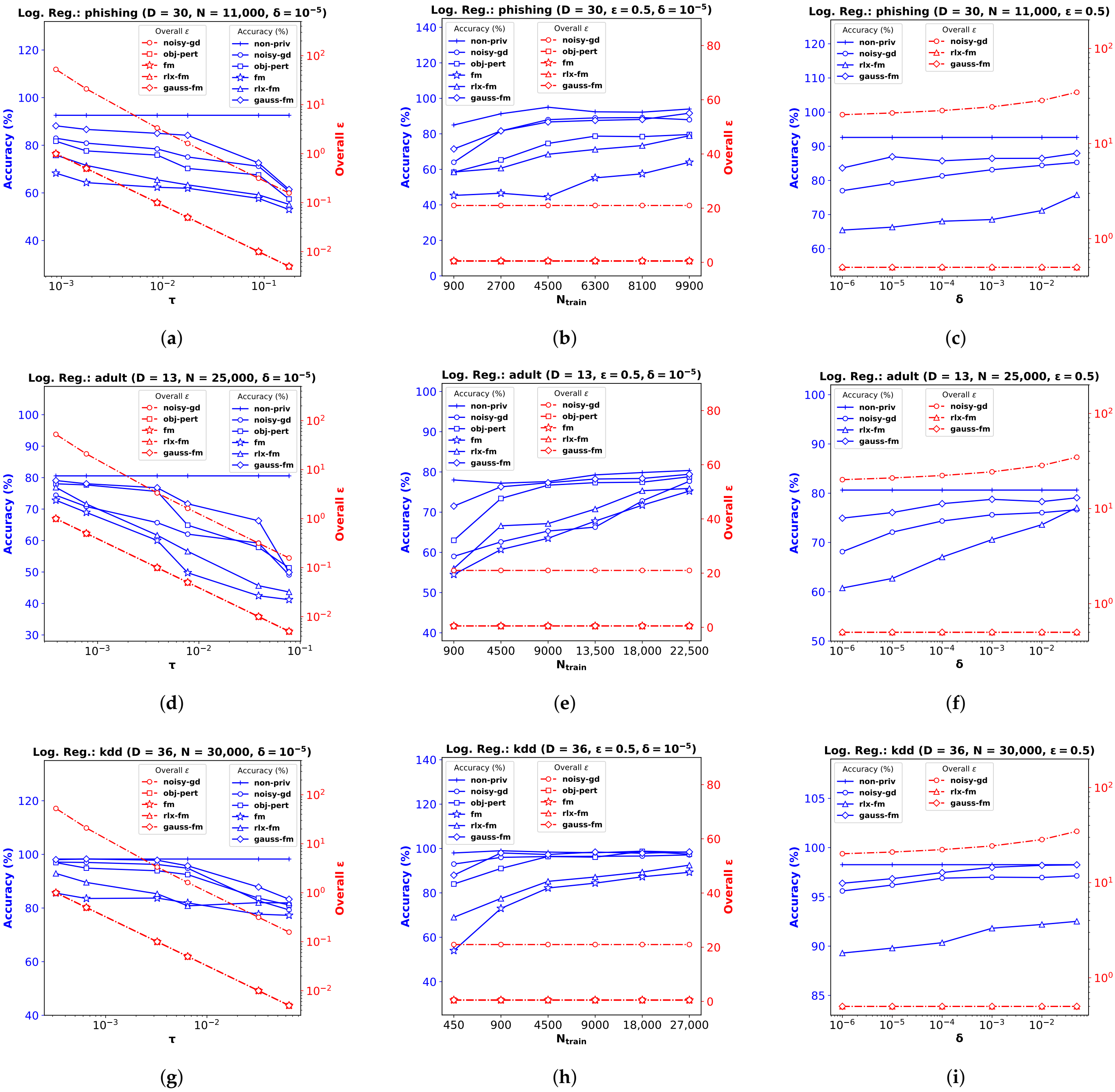
6.2. Logistic Regression

For the logistic regression problem, we again perform experiments on three real datasets (and a synthetic dataset, as shown in Appendix B): the *Phishing Websites* dataset (**phishing**) [47] with dimensionality $D = 30$ (Figure 2a–c), the *Census Income* dataset (**adult**) [47] with $D = 13$ (Figure 2d–f), and the *KDD Cup '99* dataset (**kdd**) [47] with $D = 36$ (Figure 2g–i). As before, we pre-process the data so that the feature vectors satisfy $\left\|\mathbf{x}_n\right\|_2 \leq 1$, and $y_n \in \left\{0,1\right\} \forall\, n \in [N]$. Note for **obj-pert** that the cost function is regularized and the labels are assumed to be $\left\{-1,1\right\}$ in [8]. We divide each dataset into train and test partitions with a ratio of 90:10. We use percent accuracy on the test dataset as the performance index for logistic regression, and show the average performance over 10 independent runs.

Performance Comparison with Varying τ. We plot accuracy against the DP additive noise standard deviation τ in Figure 2a,d,g. We observe that accuracy degrades when the additive DP noise standard deviation τ increases, indicating a greater privacy guarantee at the cost of performance. When noise is too high, privacy-preserving logistic regression may not learn a meaningful $\mathbf{w}$ at all, and provide random results. Depending on the class distribution, this may not be obvious and the accuracy score may be misleading. We observe this for the **kdd** dataset in Figure 2g, where the classes are highly imbalanced, with ∼80% positive labels. Although the existing **fm** performs poorly on this dataset, our proposed **gauss-fm** provides significantly higher accuracy for all datasets, outperforming **fm**, as well as **rlx-fm**, **obj-pert**, and **noisy-gd**. As before, we observe the total privacy loss, i.e., overall ϵ spent, from the y-axis on the right.

Performance Comparison with Varying N_{train}. We perform the same steps described in Section 6.1 and observe the variation in performance with the number of training samples, N_{train} while keeping the privacy parameters fixed in Figure 2b,e,h. Accuracy generally improves with increasing N_{train}. We observe that the same DP algorithm does not perform equally well for different datasets. For example, **obj-pert** performs better than **noisy-gd** on the **adult** dataset (Figure 2e), whereas **noisy-gd** performs better than **obj-pert** on the **phishing** dataset (Figure 2b). In general, **fm** and **rlx-fm** suffer from too much noise due to the quadratic and linear dependence on D of their sensitivities, respectively. However, our proposed **gauss-fm** overcomes this issue and consistently achieves accuracy close to the **non-priv** case even for moderate sample sizes. We also show the overall privacy guarantee, as before.

Performance Comparison with Varying δ. Similar to the linear regression experiments shown in Section 6.1, we keep ϵ and N_{train} fixed for this task and vary the other privacy parameter δ. Figure 2c,f,i show that percent accuracy improves with increased δ. For sufficiently large δ (indicating 1–5% probability of the algorithm failing to provide ϵ privacy risk), **gauss-fm** accuracy can reach that of the **non-priv** algorithm in some datasets (e.g., Figure 2i). Although the accuracy of **noisy-gd** also improves, it comes at the cost of additional privacy risk, as shown in the overall ϵ vs. δ plots along the y-axes on the right. Due to the higher noise variance, **rlx-fm** achieves much inferior accuracy compared to both **gauss-fm** and **noisy-gd**.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

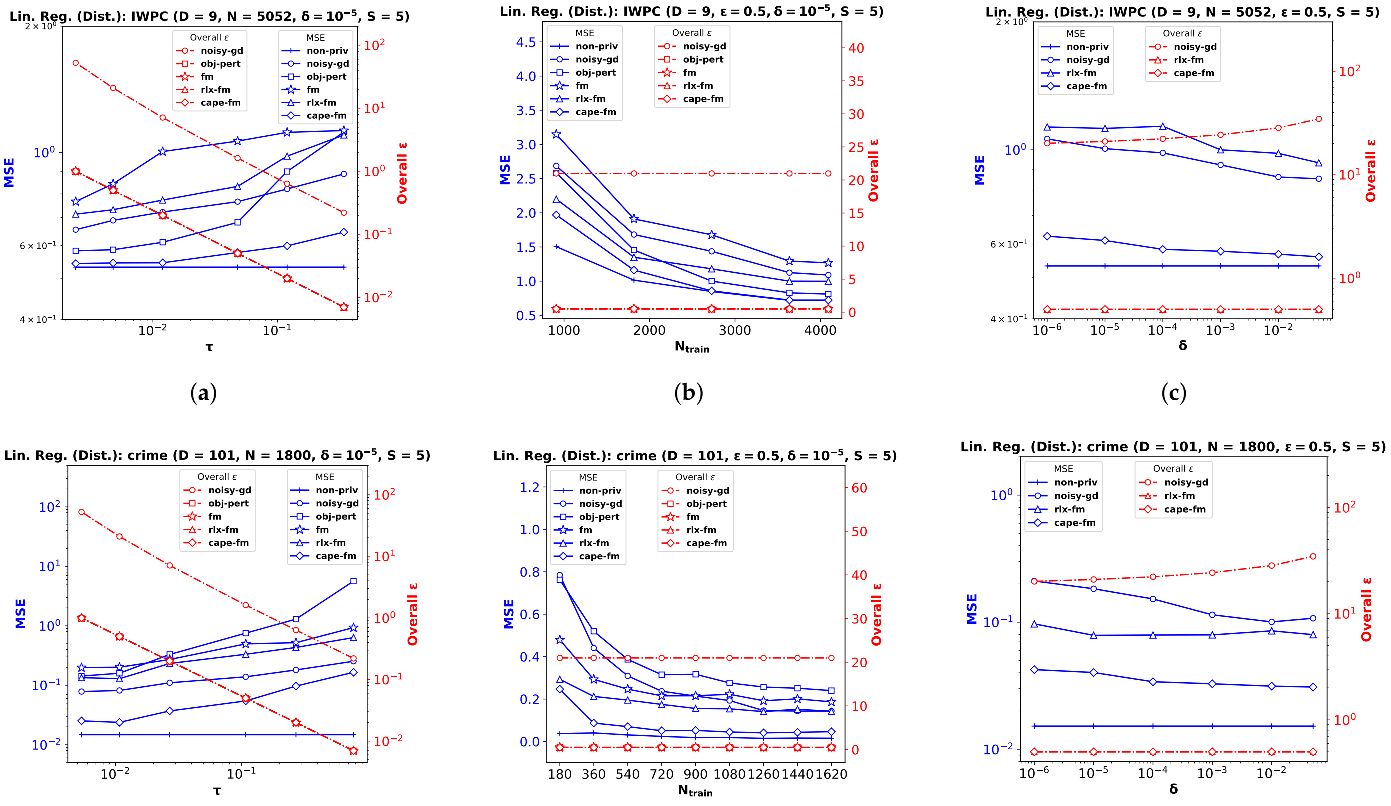
(**g**) (**h**) (**i**)

Figure 2. Logistic regression performance comparison in terms of accuracy and overall ϵ for *phishing* $(D = 30)$, *adult* $(D = 13)$, and *kdd* $(D = 36)$ datasets with varying noise standard deviation τ in (**a**,**d**,**g**), the number of training samples N_{train} in (**b**,**e**,**h**), and privacy parameter δ in (**c**,**f**,**i**).

6.3. Decentralized Functional Mechanism (capeFM)

In this section, we empirically show the effectiveness of capeFM, our proposed decentralized Gaussian FM which utilizes the CAPE [6] protocol. We implement differentially private linear and logistic regression for the decentralized-data setting using the same datasets described in Sections 6.1 and 6.2, respectively. Note that the IWPC [23] data were collected from 21 sites across 9 countries. After obtaining informed consent to use de-identified data from patients prior to the study, the Pharmacogenetics Knowledge Base has since made the dataset publicly available for research purpose. As mentioned before, the type of data contained in the IWPC dataset is similar to many other medical datasets containing private information [20].

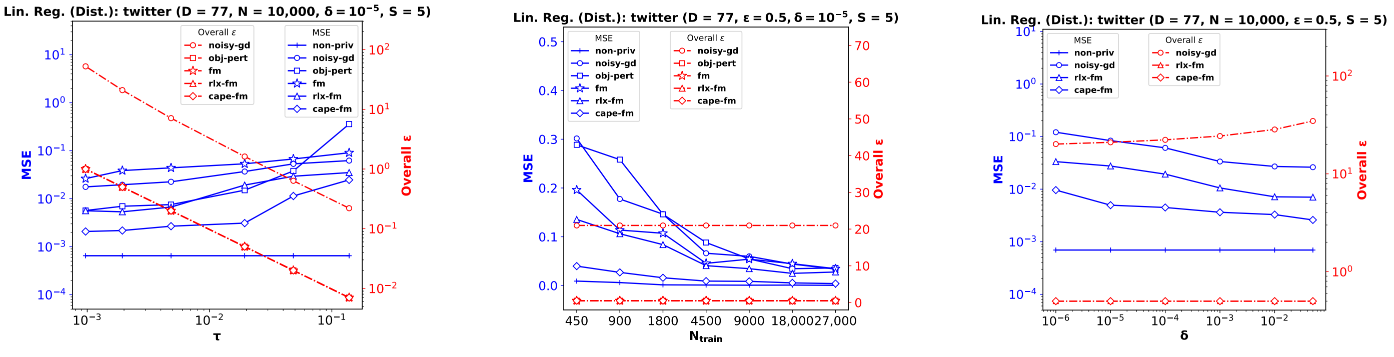
We implement our proposed **cape-fm** according to Algorithm 3, along with **fm**, **rlx-fm**, **obj-pert**, and **noisy-gd** according to the conventional decentralized DP approach. We compare the performance of these methods in Figures 3 and 4. Similar to the pooled-data scenario, we also compare performance of these algorithms with non-private linear and logistic regression (**non-priv**). For these experiments, we assume $N_s = \frac{N}{S}$ and $\tau_s = \tau$. Recall that the CAPE scheme achieves the same noise variance as the pooled-data scenario in the symmetric setting (see Lemma 1 [6] in Section 2.1). As our proposed capeFM algorithm follows the CAPE scheme, we attain the same advantages. When varying privacy parameters and N_{train}, we keep the number of sites S fixed. Additionally, we show the variation in performance due to change in the number of sites in Figure 5. We pre-process each dataset as before, and use MSE and percent accuracy on test dataset as performance indices of the decentralized linear and logistic regression problems, respectively.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 3. *Cont.*



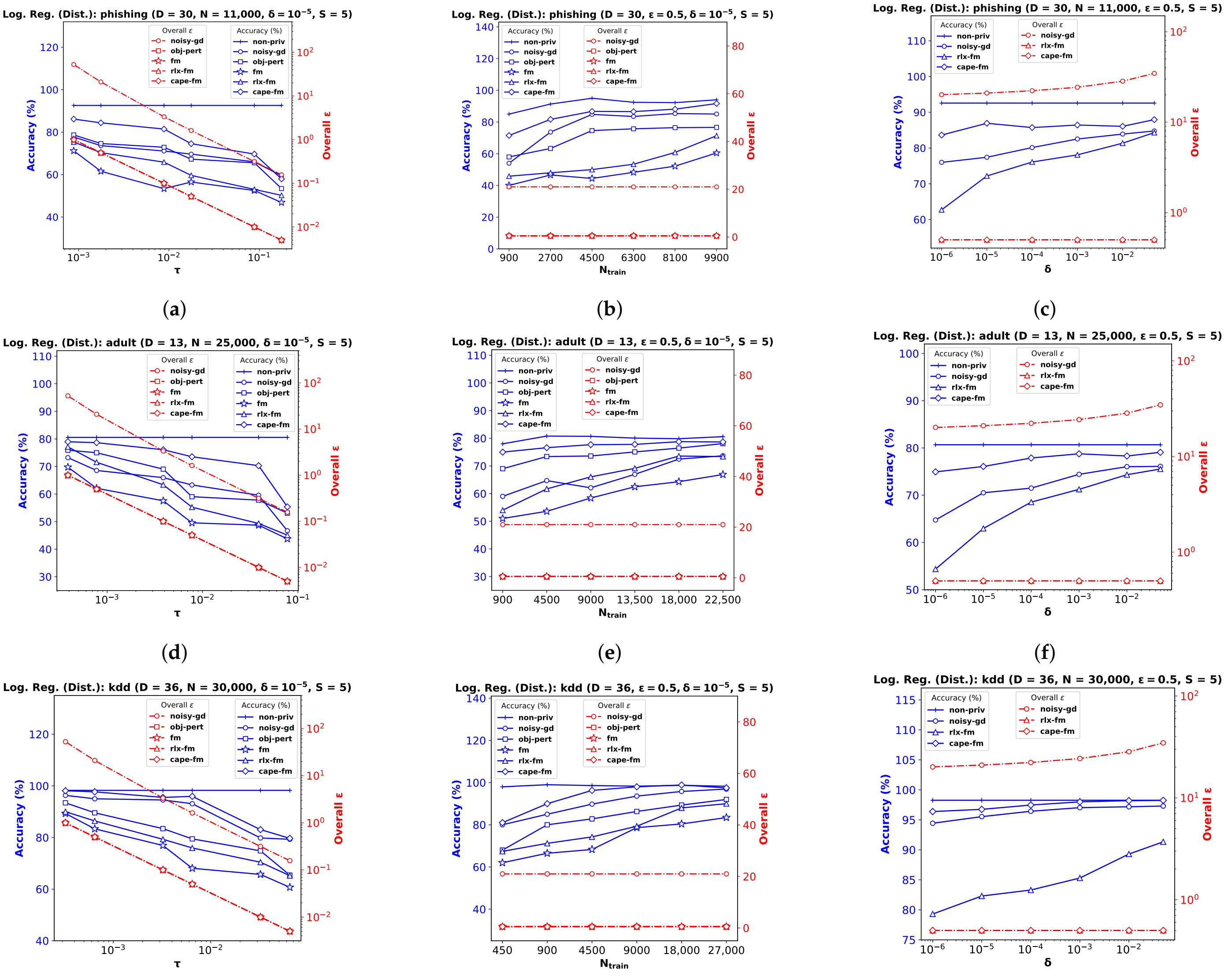
Lin. Reg. (Dist.): twitter (D = 77, N = 10,000, $\delta = 10^{-5}$, S = 5)

Lin. Reg. (Dist.): twitter (D = 77, ϵ = 0.5, $\delta = 10^{-5}$, S = 5)

Lin. Reg. (Dist.): twitter (D = 77, N = 10,000, ϵ = 0.5, S = 5)

(**g**) (**h**) (**i**)

Figure 3. Decentralized linear regression performance comparison in terms of MSE and overall ϵ for *IWPC* ($D = 9$), *crime* ($D = 101$), and *twitter* ($D = 77$) datasets with varying noise standard deviation τ in (**a**,**d**,**g**), the number of training samples N_{train} in (**b**,**e**,**h**), and privacy parameter δ in (**c**,**f**,**i**).



Log. Reg. (Dist.): phishing (D = 30, N = 11,000, $\delta = 10^{-5}$, S = 5)

Log. Reg. (Dist.): phishing (D = 30, ϵ = 0.5, $\delta = 10^{-5}$, S = 5)

Log. Reg. (Dist.): phishing (D = 30, N = 11,000, ϵ = 0.5, S = 5)

(**a**) (**b**) (**c**)

Log. Reg. (Dist.): adult (D = 13, N = 25,000, $\delta = 10^{-5}$, S = 5)

Log. Reg. (Dist.): adult (D = 13, ϵ = 0.5, $\delta = 10^{-5}$, S = 5)

Log. Reg. (Dist.): adult (D = 13, N = 25,000, ϵ = 0.5, S = 5)

(**d**) (**e**) (**f**)

Log. Reg. (Dist.): kdd (D = 36, N = 30,000, $\delta = 10^{-5}$, S = 5)

Log. Reg. (Dist.): kdd (D = 36, ϵ = 0.5, $\delta = 10^{-5}$, S = 5)

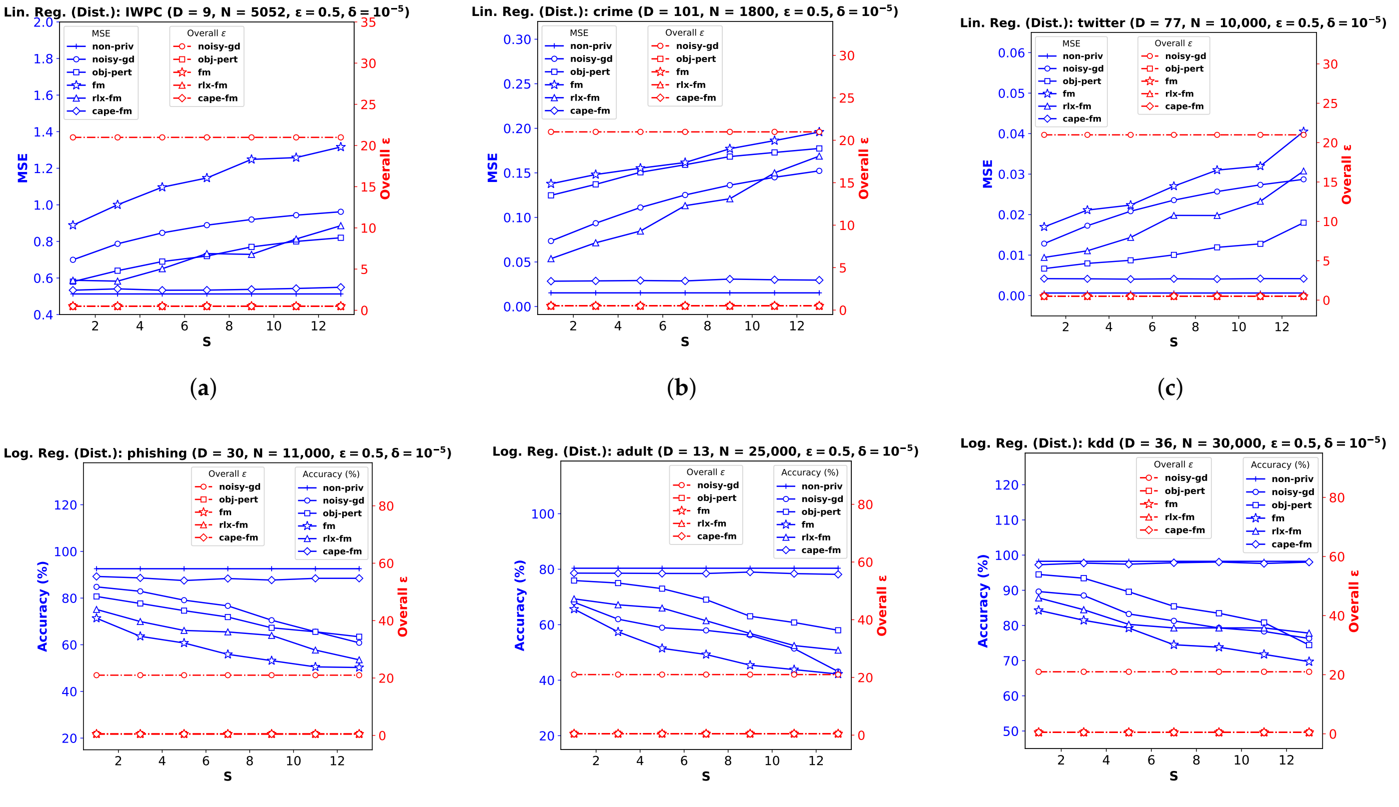
Log. Reg. (Dist.): kdd (D = 36, N = 30,000, ϵ = 0.5, S = 5)

(**g**) (**h**) (**i**)

Figure 4. Decentralized logistic regression performance comparison in terms of accuracy and overall ϵ for *phishing* ($D = 30$), *adult* ($D = 13$), and *kdd* ($D = 36$) datasets with varying noise standard deviation τ in (**a**,**d**,**g**), the number of training samples N_{train} in (**b**,**e**,**h**), and privacy parameter δ in (**c**,**f**,**i**).

Performance Comparison by Varying τ. For this experiment, we keep the total number of samples N, privacy parameter δ, and the number of sites S fixed. We observe from the plots (a), (d), and (g) in both Figures 3 and 4 that as τ increases, the performance degrades. The proposed **cape-fm** outperforms conventional decentralized **noisy-gd**, **obj-pert**, **fm**, and **rlx-fm** by a larger margin than the pooled-data case. The reason for this is that we can achieve a much smaller noise variance at the aggregator due to the correlated noise scheme detailed in Section 5.3. The utility of **cape-fm** thus stays the same as the centralized case in the decentralized-data setting, whereas the conventional scheme's utility always degrades by a factor of S (see Section 5.1). The overall ϵ usage vs. τ plots on the right y-axes for each site show that **noisy-gd** suffers from much higher privacy loss.

Performance Comparison by Varying N_{train}. We keep ϵ, δ, and S fixed while investigating variation in performance with respect to N_{train}. As the sensitivities we computed in Sections 4.1 and 4.2 are inversely proportional to the sample size, it is straightforward to infer that guaranteeing smaller privacy risk and higher utility is much easier when the sample size is large. Similar to the pooled-data cases in Sections 6.1 and 6.2, we again observe from the plots (b), (e), and (h) in both Figures 3 and 4 that, for sufficiently large $N_{train} = SN_{s,train}$, utility of **cape-fm** can reach that of the **non-priv** case. Note that the **non-priv** algorithms are the same as the pooled-data scenario, because if privacy is not a concern, all sites can send the data to aggregator for learning.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 5. Decentralized linear and logistic regression performance comparison and overall ϵ with varying number of sites S for the datasets (**a**) *IWPC* ($D = 9$), (**b**) *crime* ($D = 101$), (**c**) *twitter* ($D = 77$), (**d**) *phishing* ($D = 30$), (**e**) *adult* ($D = 13$), and (**f**) *kdd* ($D = 36$).

Performance Comparison by Varying δ. For this task, we keep ϵ, N_{train}, and S fixed. Note according to the CAPE scheme that the proposed **cape-fm** algorithm guarantees (ϵ, δ)-DP where (ϵ, δ) satisfy the relation $\delta = 2\frac{\sigma_z}{\epsilon - \mu_z}\phi\left(\frac{\epsilon - \mu_z}{\sigma_z}\right)$. Recall that δ is the probability that the algorithm fails to provide privacy risk ϵ, and that we assumed a fixed number of colluding sites $S_C = \lceil\frac{S}{3}\rceil - 1$. From the plots (c), (f), and (i) in both Figures 3 and 4, we observe that even for moderate values of δ, **cape-fm** easily outperforms **rlx-fm** and **noisy-gd**. Moreover, as seen from the overall ϵ plots, **noisy-gd** provides a much weaker privacy guarantee. Thus, our proposed **cape-fm** algorithm offers superior performance and privacy–utility trade-off in the decentralized setting.

Performance Comparison by Varying S. Finally, we investigate performance variation with the number of sites S, keeping the privacy and dataset parameters fixed. This automatically varies the number of samples N_s at each site $s \in [S]$, as we consider the symmetric setting. Figure 5a–c shows the results for decentralized linear regression, and Figure 5d–f shows the results for decentralized logistic regression. We observe that the variation in S does not affect the utility of **cape-fm**, as long as the number of colluding sites meets the condition $S_C \leq \lceil \frac{S}{3} \rceil - 1$. However, increasing S leads to significant degradation in performance for conventional decentralized DP mechanisms, since the additive noise variance increases as N_s decreases. We show additional experimental results on synthetic datasets in Appendix B.

7. Conclusions and Future Work

In this paper, we proposed Gaussian FM that offers a significant improvement over the existing FM to compute functions that are commonly used in signal processing and machine learning applications, satisfying differential privacy. Our improvement stems from a novel sensitivity analysis that resulted in an orders-of-magnitude reduction in the amount of noise added to the coefficients of the Stone–Weierstrass decomposition of the functions. We showed two common regression problems—linear and logistic regression—as examples to demonstrate our analyses. Additionally, we experimentally showed the superior privacy guarantee and utility of our proposed method over existing methods by varying privacy parameters and relevant dataset parameters for both synthetic and real datasets. We extended our Gaussian FM algorithm to decentralized data settings by taking advantage of a correlated noise protocol, CAPE, and proposed capeFM, which ensures the same utility as the pooled-data scenario in certain regimes. We empirically compared the performance of the proposed capeFM with that of existing and conventional algorithms for decentralized linear and logistic regression problems. In addition to varying privacy and dataset parameters, we showed performance comparison by varying the number of sites, which further proves the superior privacy guarantee and improved utility of our proposed method. For future work, we plan to extend our research to more complex algorithms and neural networks to ensure differential privacy on other challenging signal processing and machine learning problems.

Author Contributions: Conceptualization, methodology, formal analysis, N.T., J.M., A.D.S. and H.I.; software, data curation, N.T. and H.I.; supervision, H.I.; writing—original draft preparation, N.T.; writing—review and editing, H.I., J.M. and A.D.S.; funding acquisition, A.D.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work of A.D. Sarwate was funded in part by the US National Science Foundation under awards CNS-2148104 and CIF-1453432 and by the US National Institutes of Health under award 2R01DA040487-01A1.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The experimental data used to evaluate the performance of the algorithms proposed in this paper are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Comparison of Sensitivity and Noise Standard Deviation

To provide further details and rationale behind the superior performance of our proposed Gaussian FM (**gauss-fm**) over the original FM [7] (**fm**) and the relaxed FM [10] (**rlx-fm**) algorithms, we compare the additive noise standard deviation τ for each mechanism by varying the privacy parameter ϵ for different values of data dimension D. Recall that τ is scaled to the sensitivity of the data-dependent terms in the Stone–Weierstrass [35] decomposition of the objective function. The computed sensitivities for each of the three mechanisms are shown in Table A1.

Table A1. Comparison of sensitivities for various DP mechanisms.

		Δ^{fm}	$\Delta^{rlx\text{-}fm}$	$\Delta^{gauss\text{-}fm}$
Linear Regression	$j=0$	$\frac{2}{N}(1+D)^2$	$\frac{2}{N}\sqrt{1+4D+D^2}$	$\frac{1}{N}$
	$j=1$	$\frac{2}{N}(1+D)^2$	$\frac{2}{N}\sqrt{1+4D+D^2}$	$\frac{4}{N}$
	$j=2$	$\frac{2}{N}(1+D)^2$	$\frac{2}{N}\sqrt{1+4D+D^2}$	$\frac{1}{N}$
Logistic Regression	$j=1$	$\frac{1}{N}\left(\frac{D^2}{4}+3D\right)$	$\frac{1}{N}\sqrt{\frac{D^2}{16}+D}$	$\frac{1}{N}$
	$j=2$	$\frac{1}{N}\left(\frac{D^2}{4}+3D\right)$	$\frac{1}{N}\sqrt{\frac{D^2}{16}+D}$	$\frac{1}{8N}$

As mentioned before, the sensitivity terms for our proposed **gauss-fm** are tailored to the order j, and do not depend on the ambient dimension D. On the other hand, the sensitivity terms for both **fm** and **rlx-fm** depend on D. This results in injecting prohibitively large amounts of noise into the function computation. The proofs of the $\mathcal{L}_1$-sensitivity terms Δ^{fm} for **fm** are provided in [7], and the proof of the $\mathcal{L}_2$-sensitivity $\Delta^{rlx\text{-}fm}$ for **rlx-fm** for the logistic regression is shown in [10]. We can follow the similar procedure outlined by Ding et al. [10] to obtain the $\mathcal{L}_2$-sensitivity for the linear regression problem as $\frac{2}{N}\sqrt{1+4D+D^2}$. The proof is as follows:

Proof. Let the n-th sample of a dataset $\mathbb{D}$ be denoted by a tuple $t_n = (\mathbf{x}_n, y_n)$, where $\mathbf{x}_n \in \mathbb{R}^D$ is the feature vector and $y_n \in \mathbb{R}$ is the response for $n \in [N]$. Let us assume that two neighboring datasets $\mathbb{D}$ and $\mathbb{D}'$ differ in the last tuple t_N and $t_{N'}$. For linear regression we have

$$\begin{aligned} f_D(\mathbf{w}) &= \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^\top \mathbf{w}\right)^2 \\ &= \left(\frac{1}{N}\sum_{n=1}^{N} y_n^2\right) + \sum_{d=1}^{D}\left(-\frac{2}{N}\sum_{n=1}^{N} y_n x_{nd}\right) w_d + \sum_{d_1=1}^{D}\sum_{d_2=1}^{D}\left(\frac{1}{N}\sum_{n=1}^{N} x_{nd_1}x_{nd_2}\right) w_{d_1} w_{d_2} \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=0}^{2}\sum_{\phi\in\Phi_j} \lambda_{\phi t_n}\phi(\mathbf{w}), \end{aligned}$$

where $\{\lambda_{\phi t_n}\}_{\phi\in\Phi_0} =: \lambda_{0t_n} = y_n^2$; $\{\lambda_{\phi t_n}\}_{\phi\in\Phi_1} =: \lambda_{1t_n} = -2y_n\mathbf{x}_n$; and $\{\lambda_{\phi t_n}\}_{\phi\in\Phi_2} =: \lambda_{2t_n} = \mathbf{x}_n^2$. We denote $\mathcal{A}_1 = \left\{\frac{1}{N}\sum_{n=1}^{N}\lambda_{\phi t_n}\right\}_{\phi\in\cup_{j=0}^{2}\Phi_j}$ and $\mathcal{A}_2 = \left\{\frac{1}{N}\sum_{n=1}^{N}\lambda_{\phi t_{n'}}\right\}_{\phi\in\cup_{j=0}^{2}\Phi_j}$ as the set of polynomial coefficients of $f_D(\mathbf{w})$ and $f_{D'}(\mathbf{w})$. We also denote

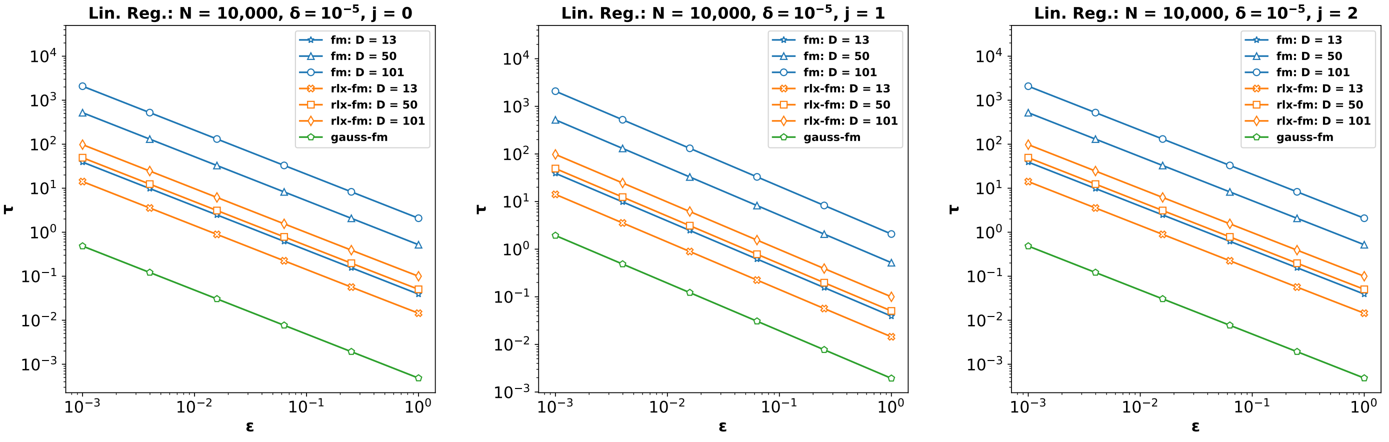
$$\mathcal{C} = \begin{pmatrix} y^2 \\ -2yx_{(1)} \\ \cdots \\ -2yx_{(D)} \\ x_{(1)}x_{(1)} \\ \cdots \\ x_{(D)}x_{(D)} \end{pmatrix} \in \mathbb{R}^{(1+D+D^2)\times 1},$$

where $x_{(c)}$ represents the c-th element in the feature vector $\mathbf{x}$. Now, the $\mathcal{L}_2$-sensitivity of linear regression for the relaxed FM algorithm can be expressed as

$$\begin{aligned}
\Delta_2 &= \left\|\mathcal{A}_1 - \mathcal{A}_2\right\|_2 \\
&= \left\| \left\{ \frac{1}{N}\sum_{n=1}^{N} \lambda_{\phi t_n} - \frac{1}{N}\sum_{n=1}^{N} \lambda_{\phi t_{n'}} \right\}_{\phi \in \cup_{j=0}^{2} \Phi_j} \right\|_2 \\
&= \frac{1}{N} \left\| \left\{ \lambda_{\phi t_N} - \lambda_{\phi t_{N'}} \right\}_{\phi \in \cup_{j=0}^{2} \Phi_j} \right\|_2 \\
&\leq \frac{2}{N} \max_{t=(\mathbf{x},y)} \left\|\mathcal{C}\right\|_2 \\
&= \frac{2}{N} \max_{t=(\mathbf{x},y)} \sqrt{y^2 + \sum_{d=1}^{D} \left(-2yx_{(d)}\right)^2 + \sum_{d_1=1}^{D}\sum_{d_2=2}^{D} \left(x_{(d_1)}x_{(d_2)}\right)^2} \\
&= \frac{2}{N}\sqrt{1 + 4D + D^2} \triangleq \Delta^{rlx\text{-}fm},
\end{aligned}$$

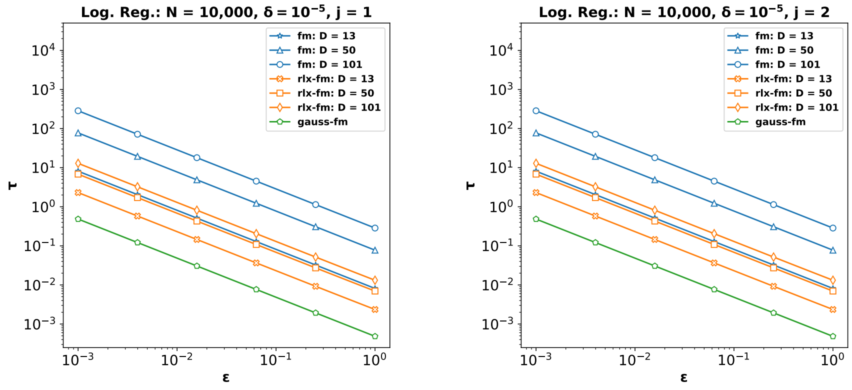
where t is an arbitrary tuple. □

We now empirically compare the additive noise standard deviation τ for **gauss-fm**, **fm**, and **rlx-fm**. In Figure A1, we show τ of the additive noise for the coefficient terms for different j and different data dimensionality D. We set the number of samples N = 10,000 and privacy parameter $\delta = 10^{-5}$. From the figure, we observe that the noise standard deviation for **gauss-fm** is significantly lower than the noise standard deviation for both **fm** and **rlx-fm** algorithms. We achieve this by our novel sensitivity analysis, which is tailored to different coefficient terms (i.e., the order j) as shown in Section 4 and Algorithm 1.



(**a**) (**b**) (**c**)

Figure A1. Standard deviation τ of the additive noise for (**a**) $j = 0$, (**b**) $j = 1$, and (**c**) $j = 2$ for different values of dimensionality D for differentially private linear regression using **fm**, **rlx-fm**, and **gauss-fm**.



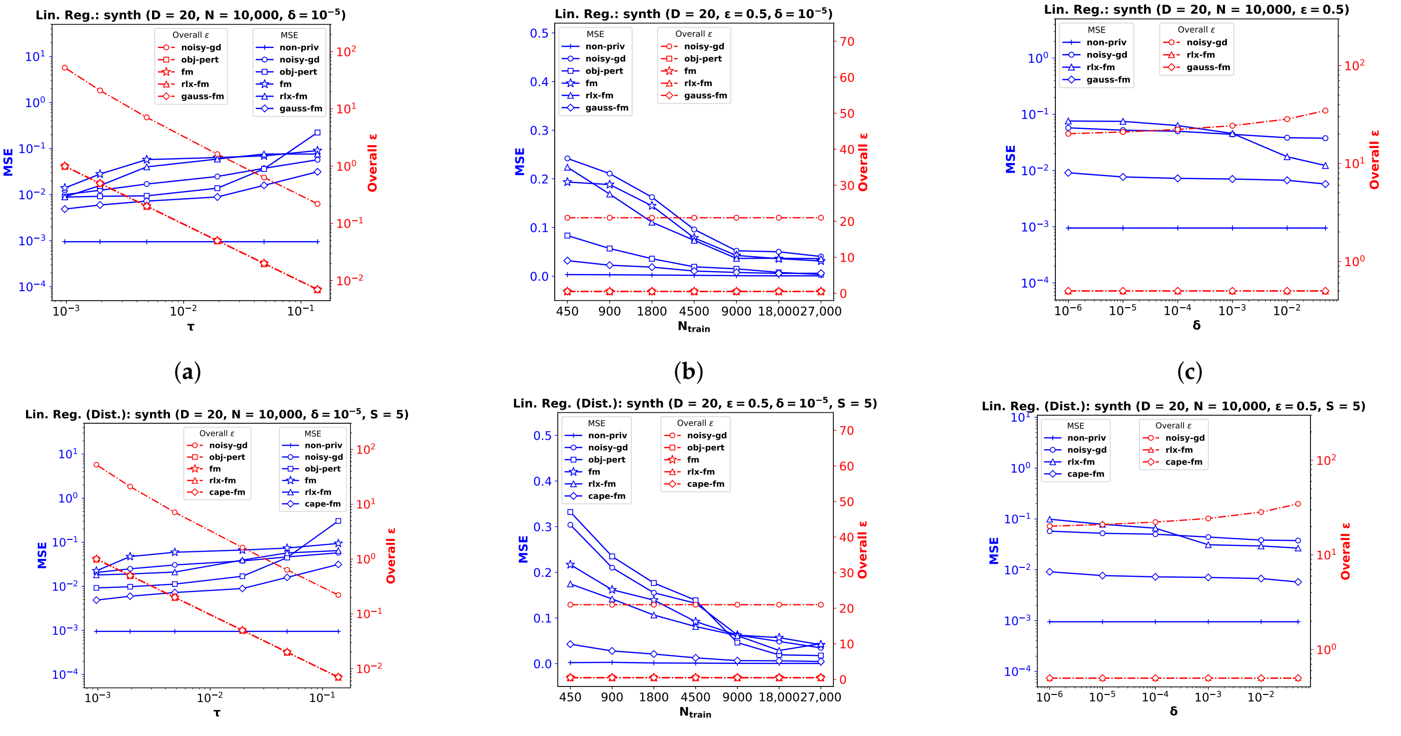
(**a**) (**b**)

Figure A2. Standard deviation τ of the additive noise for (**a**) $j = 1$ and (**b**) $j = 2$ for different values of dimensionality D for differentially private logistic regression using **fm**, **rlx-fm**, and **gauss-fm**.

Appendix B. Additional Experimental Results on Synthetic Data

In addition to the real datasets, we perform experiments on *synthetic* datasets while keeping the setup identical to the one described in Section 6. We generate random samples **X** and outputs **y** with dimensionality $D = 20$ for the linear regression problems in pooled-data (Figure A3a–c) and distributed-data settings ((Figure A3d–f). For logistic regression in pooled-data (Figure A3g–i) and distributed-data settings (Figure A3j–l), we generate another synthetic dataset with dimensionality $D = 50$ where outputs **y** are class labels.

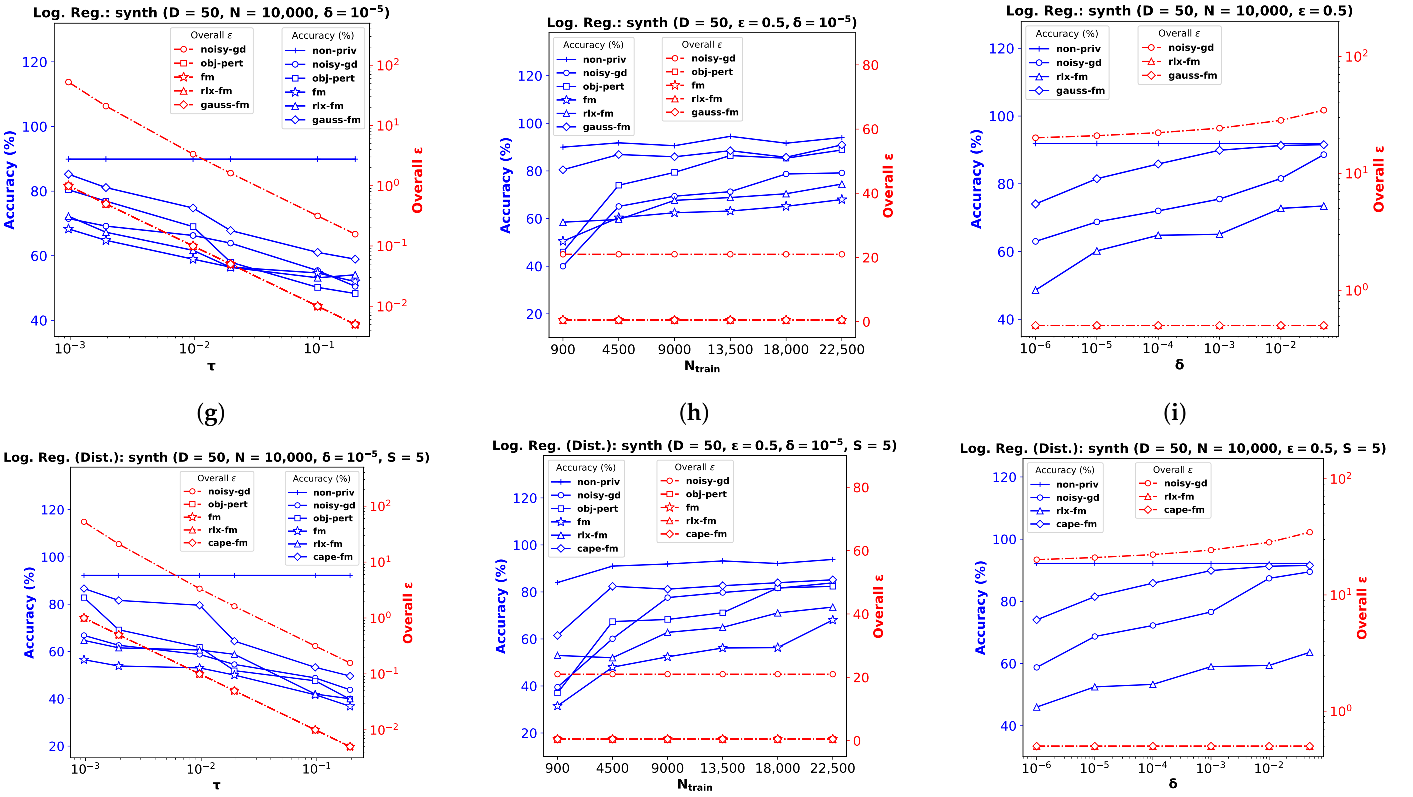
Similar to the results observed in Section 6, performance generally improves with lower noise variance and a weaker privacy guarantee. Our proposed **gauss-fm** and **cape-fm** algorithms consistently outperform existing **fm**, **rlx-fm**, **noisy-gd**, and **obj-pert** methods. We also show variation in performance with number of sites S in Figure A4. The empirical results verify that the utility of **cape-fm** does not degrade with increased S, and thus provides a better privacy guarantee over conventional decentralized DP schemes.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

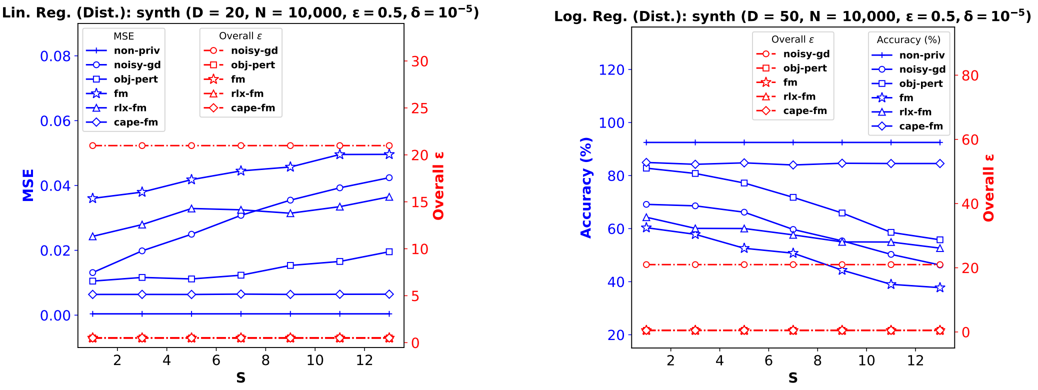
Figure A3. *Cont.*



(g) (h) (i)

(j) (k) (l)

Figure A3. Performance comparison and overall ϵ for *synthetic* datasets with varying noise standard deviation τ in (**a**,**d**,**g**,**j**) , number of training samples N_{train} in (**b**,**e**,**h**,**k**), and privacy parameter δ in (**c**,**f**,**i**,**l**).



(a) (b)

Figure A4. Decentralized linear and logistic regression performance comparison and overall ϵ with varying number of sites S for the datasets (**a**) *synth (D = 20)* and (**b**) *synth (D = 50)*.

References

1. Dwork, C. Differential Privacy. In *Automata, Languages and Programming. ICALP 2006*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4052, pp. 1–12.
2. Sarwate, A.D.; Chaudhuri, K. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Process. Mag.* **2013**, *30*, 86–94. [CrossRef] [PubMed]
3. Jayaraman, B.; Evans, D. Evaluating differentially private machine learning in practice. In Proceedings of the 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, USA, 14–16 August 2019; pp. 1895–1912.
4. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
5. Desfontaines, D.; Pejó, B. Sok: Differential privacies. *Proc. Priv. Enhancing Technol.* **2020**, *2020*, 288–313. [CrossRef]

6. Imtiaz, H.; Mohammadi, J.; Silva, R.; Baker, B.; Plis, S.M.; Sarwate, A.D.; Calhoun, V.D. A Correlated Noise-Assisted Decentralized Differentially Private Estimation Protocol, and its Application to fMRI Source Separation. *IEEE Trans. Signal Process.* **2021**, *69*, 6355–6370. [CrossRef] [PubMed]
7. Zhang, J.; Zhang, Z.; Xiao, X.; Yang, Y.; Winslett, M. Functional mechanism: Regression analysis under differential privacy. *arXiv* **2012**, arXiv:1208.0219.
8. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially private empirical risk minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109.
9. Bassily, R.; Smith, A.; Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 18–21 October 2014; pp. 464–473.
10. Ding, J.; Zhang, X.; Li, X.; Wang, J.; Yu, R.; Pan, M. Differentially private and fair classification via calibrated functional mechanism. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 622–629. [CrossRef]
11. Phan, N.; Vu, M.; Liu, Y.; Jin, R.; Dou, D.; Wu, X.; Thai, M.T. Heterogeneous Gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. *arXiv* **2019**, arXiv:1906.01444.
12. Song, S.; Chaudhuri, K.; Sarwate, A.D. Stochastic gradient descent with differentially private updates. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, Austin, TX, USA, 3–5 December 2013; pp. 245–248.
13. Nozari, E.; Tallapragada, P.; Cortés, J. Differentially private distributed convex optimization via objective perturbation. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 2061–2066.
14. Wu, X.; Li, F.; Kumar, A.; Chaudhuri, K.; Jha, S.; Naughton, J. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 1307–1322.
15. Smith, A. Privacy-preserving statistical estimation with optimal convergence rates. In Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, San Jose, CA, USA, 6–8 June 2011; pp. 813–822.
16. McSherry, F.; Talwar, K. Mechanism design via differential privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103.
17. Jorgensen, Z.; Yu, T.; Cormode, G. Conservative or liberal? Personalized differential privacy. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering, Seoul, Republic of Korea, 13–17 April 2015; pp. 1023–1034.
18. Aono, Y.; Hayashi, T.; Trieu Phong, L.; Wang, L. Scalable and secure logistic regression via homomorphic encryption. In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, 9–11 March 2016; pp. 142–144.
19. Xu, D.; Yuan, S.; Wu, X. Achieving differential privacy and fairness in logistic regression. In Proceedings of the Companion Proceedings of the 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 594–599.
20. Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; Ristenpart, T. Privacy in pharmacogenetics: An End-to-End case study of personalized Warfarin dosing. In Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), San Diego, CA, USA, 20–22 August 2014; pp. 17–32.
21. Anderson, J.L.; Horne, B.D.; Stevens, S.M.; Grove, A.S.; Barton, S.; Nicholas, Z.P.; Kahn, S.F.; May, H.T.; Samuelson, K.M.; Muhlestein, J.B.; et al. Randomized trial of genotype-guided versus standard Warfarin dosing in patients initiating oral anticoagulation. *Circulation* **2007**, *116*, 2563–2570. [CrossRef]
22. Fusaro, V.A.; Patil, P.; Chi, C.L.; Contant, C.F.; Tonellato, P.J. A systems approach to designing effective clinical trials using simulations. *Circulation* **2013**, *127*, 517–526. [CrossRef]
23. Consortium, I.W.P. Estimation of the Warfarin dose with clinical and pharmacogenetic data. *N. Engl. J. Med.* **2009**, *360*, 753–764.
24. Sconce, E.A.; Khan, T.I.; Wynne, H.A.; Avery, P.; Monkhouse, L.; King, B.P.; Wood, P.; Kesteven, P.; Daly, A.K.; Kamali, F. The impact of CYP2C9 and VKORC1 genetic polymorphism and patient characteristics upon Warfarin dose requirements: Proposal for a new dosing regimen. *Blood* **2005**, *106*, 2329–2333. [CrossRef] [PubMed]
25. Gade, S.; Vaidya, N.H. Private learning on networks. *arXiv* **2016**, arXiv:1612.05236.
26. Heikkilä, M.; Lagerspetz, E.; Kaski, S.; Shimizu, K.; Tarkoma, S.; Honkela, A. Differentially private Bayesian learning on distributed data. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3229–3238.
27. Tajeddine, R.; Jälkö, J.; Kaski, S.; Honkela, A. Privacy-preserving data sharing on vertically partitioned data. *arXiv* **2020**, arXiv:2010.09293.
28. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
29. Heikkilä, M.A.; Koskela, A.; Shimizu, K.; Kaski, S.; Honkela, A. Differentially private cross-silo federated learning. *arXiv* **2020**, arXiv:2007.05553.
30. Xu, D.; Yuan, S.; Wu, X. Achieving differential privacy in vertically partitioned multiparty learning. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 5474–5483.
31. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 486–503.

32. Anandan, B.; Clifton, C. Laplace noise generation for two-party computational differential privacy. In Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 21–23 July 2015; pp. 54–61.
33. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
34. Mironov, I. Rényi differential privacy. In Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275.
35. Rudin, W. *Principles of Mathematical Analysis*; International Series in Pure and Applied Mathematics; McGraw-Hill: New York, NY, USA, 1976.
36. Imtiaz, H.; Sarwate, A.D. Distributed differentially private algorithms for matrix and tensor factorization. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 1449–1464. [CrossRef]
37. Balle, B.; Wang, Y.X. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 394–403.
38. Holohan, N.; Antonatos, S.; Braghin, S.; Mac Aonghusa, P. The bounded Laplace mechanism in differential privacy. *arXiv* **2018**, arXiv:1808.10410.
39. Dong, J.; Roth, A.; Su, W.J. Gaussian differential privacy. *arXiv* **2019**, arXiv:1905.02383.
40. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.
41. Ergün, G. Random Matrix Theory. In *Encyclopedia of Complexity and Systems Science*; Meyers, R.A., Ed.; Springer: New York, NY, USA, 2009; pp. 7505–7520. [CrossRef]
42. Strang, G. *Introduction to Linear Algebra*; Wellesley-Cambridge Press: Wellesley, MA, USA, 1993; Volume 3.
43. Dwork, C.; Talwar, K.; Thakurta, A.; Zhang, L. Analyze Gauss: Optimal Bounds for Privacy-Preserving Principal Component Analysis. In Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, New York, NY, USA, 31 May–3 June 2014. [CrossRef]
44. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
45. Redmond, M.; Baveja, A. A data-driven software tool for enabling cooperative information sharing among police departments. *Eur. J. Oper. Res.* **2002**, *141*, 660–678. [CrossRef]
46. Kawala, F.; Douzal-Chouakria, A.; Gaussier, E.; Dimert, E. Prédictions d'activité dans les réseaux sociaux en ligne. In Proceedings of the 4ième Conférence sur les Modèles et l'Analyse des réseaux: Approches Mathématiques et Informatiques, Saint-Etienne, France, 16–18 October 2013; p. 16.
47. Dua, D.; Graff, C. UCI Machine Learning Repository, 2017. University of California, Irvine, School of Information and Computer Sciences. Available online: http://archive.ics.uci.edu/ml (accessed on 15 April 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.