sciendo

Proceedings on Privacy Enhancing Technologies ; 2019 (1):26–46

Thee Chanyaswad*, Changchang Liu, and Prateek Mittal

RON-Gauss: Enhancing Utility in Non-Interactive Private Data Release

Abstract: A key challenge facing the design of differential privacy in the non-interactive setting is to maintain the utility of the released data. To overcome this challenge, we utilize the *Diaconis-Freedman-Meckes (DFM) effect*, which states that most projections of high-dimensional data are nearly Gaussian. Hence, we propose the *RON-Gauss* model that leverages the novel combination of dimensionality reduction via random orthonormal (RON) projection and the Gaussian generative model for synthesizing differentially-private data. We analyze how RON-Gauss benefits from the DFM effect, and present multiple algorithms for a range of machine learning applications, including both unsupervised and supervised learning. Furthermore, we rigorously prove that (a) our algorithms satisfy the strong ϵ-differential privacy guarantee, and (b) RON projection can lower the level of perturbation required for differential privacy. Finally, we illustrate the effectiveness of RON-Gauss under three common machine learning applications – clustering, classification, and regression – on three large real-world datasets. Our empirical results show that (a) RON-Gauss outperforms previous approaches by up to an order of magnitude, and (b) loss in utility compared to the non-private real data is small. Thus, RON-Gauss can serve as a key enabler for real-world deployment of privacy-preserving data release.

Keywords: differential privacy, non-interactive private data release, random orthonormal projection, Gaussian generative model, Diaconis-Freedman-Meckes effect

DOI 10.2478/popets-2019-0003
Received 2018-05-31; revised 2018-09-15; accepted 2018-09-16.

***Corresponding Author: Thee Chanyaswad:** Princeton University, E-mail: tc7@princeton.edu (Currently at KBTG Machine Learning Team, Thailand, E-mail: theerachai.c@kbtg.tech)
Changchang Liu: Princeton University, E-mail: cl12@princeton.edu (Currently at IBM T. J. Watson Research Center, E-mail: changchang.liu33@ibm.com)
Prateek Mittal: Princeton University, E-mail: pmittal@princeton.edu

1 Introduction

In an era of big data and machine learning, our digital society is generating a considerable amount of personal data at every moment. These data can be sensitive, and as a result, significant privacy concerns arise. Even with the use of anonymization mechanisms, privacy leakage can still occur, as exemplified by Narayanan et al. [91], Calandrino et al. [24], Barbaro and Zeller [8], Haeberlen et al. [53], and Backes et al. [4]. These privacy leaks have motivated the design of formal privacy analysis. To this end, *differential privacy (DP)* has become the gold standard for a rigorous privacy guarantee [15, 35–37, 39]. Many mechanisms have been proposed to comply with differential privacy [15, 26, 28, 36, 39, 41, 57, 76, 84, 93, 128], and various implementations of differentially-private systems have been presented in the literature [1, 14, 44, 45, 51, 74, 79, 112, 117, 129].

There are two settings under differential privacy – interactive and non-interactive [35]. Among the two, the non-interactive setting has traditionally been more challenging to implement due to the fact that the perturbation required is often too high for the published data to be truly useful [11, 21, 40, 118]. However, this setting is still attractive, as there are incentives for the data collectors to release the data in order to seek outside expertise, e.g. the Netflix prize [97], and OpenSNP [116]. Concurrently, there are incentives for researchers to obtain the data in their entirety, as existing software frameworks for data analytics could be directly used [11, 80]. Particularly, in the current era when machine learning has become the ubiquitous tool in data analysis, non-interactive data release would allow virtually instant compatibility with existing learning algorithms. For these reasons, we aim to design a non-interactive differentially-private (DP) data release system.

In this work, we draw inspiration from the *Diaconis-Freedman-Meckes (DFM) effect* [87], which shows that, under suitable conditions, most projections of high-dimensional data are nearly *Gaussian*. This effect suggests that, although finding an accurate model for a high-dimensional dataset is generally hard [18][122, chapter 7], its projection onto a low-dimensional space may be modeled well by the Gaussian model. With

respect to the application of non-interactive DP data release, this is particularly important because, in DP, simple statistics can generally be privately learned *accurately* [38, 42], while privately learning the database accurately is generally much more difficult [21, 118, 124].

To apply the DFM effect to the non-interactive DP data release, we combine two previously non-intersecting methods – *dimensionality reduction (DR)* [13, 66, 68, 78, 128, 137] and *parametric generative model* [11, 59, 81, 94, 98, 106, 134]. Although each method has independently been explored for the application of non-interactive private data release, without properly combining the two, the DFM effect has not been fully utilized. As we show in this work, *combining the two* to utilize the DFM effect can lead to significant gain in the utility of the released data. Specifically, we closely investigate the DFM theorem by Meckes [87] and propose the *RON-Gauss* model for non-interactive private data release, which combines two techniques – *random orthonormal (RON) projection* and the *Gaussian generative model*. The first component is the DR technique used for two purposes: reducing the sensitivity (similar to previous works [13, 66, 68, 78, 128, 137]) and triggering the DFM effect (which is a first, to the best of our knowledge). The second component is the parametric model used to capture the Gaussian nature of the projection and to allow an accurate DP data modeling.

We present three algorithms for RON-Gauss that can be applied to a wide range of machine learning applications, including both *unsupervised* and *supervised* learning. The supervised learning application, in particular, provides an additional challenge on the conservation of the training label through the sanitization process. Unlike many previous works, RON-Gauss ensures the integrity of the training label of the sanitized data. We rigorously prove that all of our three algorithms preserve the strong ϵ-differential privacy guarantee. Moreover, to show the general applicability of our idea, we extend the framework to employ the *Gaussian Mixture Model (GMM)* [12, 90].

Finally, we evaluate our approach on three large real-world datasets under three common machine learning applications under the non-interactive setting of DP – clustering, classification, and regression. The non-interactive setting is attractive for these applications since it allows multiple data-analytic algorithms to be run on the released DP-data without requiring additional privacy budget like the interactive setting. We demonstrate that our method can significantly improve the utility performance by up to an order of magnitude for a fixed privacy budget, when compared to four prior methods. More importantly, our method has small loss in utility when compared to the performance of the non-private real data.

We summarize our contribution as follows.

- We exploit the DFM effect for utility enhancement of differential privacy in the non-interactive setting.
- We propose an approach consisting of random orthonormal projection and the Gaussian generative model (*RON-Gauss*) for non-interactive DP data release. We also extend this model to the Gaussian Mixture Model (GMM).
- We present three algorithms to implement RON-Gauss that are suitable for both the *unsupervised* and *supervised* machine learning tasks.
- We rigorously prove that our RON-Gauss algorithms satisfy the strong ϵ-differential privacy.
- We evaluate our method on three real-world datasets on three machine learning applications under the non-interactive DP setting – *clustering*, *classification*, and *regression*. The experimental results show that, when compared to previous methods, our method can considerably enhance the utility performance by up to an order of magnitude for a fixed privacy budget. Finally, compared to the non-private baseline of using real data, RON-Gauss incurs only a small loss in utility across all three machine learning tasks.

Roadmap: We discuss prior works in Section 2, and present the background components of our approach, including details of the DFM effect, in Section 3. Then, we present the proposed RON-Gauss model – along with its theoretical analysis, algorithms for both supervised and unsupervised learning, and the privacy proofs – in Section 4. Finally, we present experimental results showing the strength of RON-Gauss in Section 5, and the discussion in Section 6.

2 Prior Works

Our work focuses on non-interactive differentially-private (DP) data release. Since our method involves dimensionality reduction and a generative model, we discuss the relevant works under these frameworks.

2.1 Generative Models for Differential Privacy

The use of generative models for non-interactive DP data release can be classified into two groups accord-

ing to Bowen and Liu [17]: *non-parametric generative models*, and *parametric generative models*.

2.1.1 Non-Parametric Generative Models

Primarily, these models utilize the differential privacy guarantee of the Exponential mechanism [84], which defines a distribution to synthesize the data based on the input database and the pre-defined quality function. Various methods – both application-specific and application-independent – have been proposed [10, 16, 31, 52, 56, 57, 74, 77, 83, 84, 89, 98, 124, 127]. Our approach contrasts these works in two ways. First, we consider a parametric generative model, and, second, we augment our model with dimensionality reduction to trigger the DFM effect. We will compare our method to this class of model by implementing the non-parametric generative model based on Algorithm 1 in [16].

2.1.2 Parametric Generative Models

Our method of using the *Gaussian generative model*, as well as the *Gaussian Mixture Model*, falls into this category. We aim at building a system that can be applied to various applications and data-types, i.e. *application-independent*. However, many previous works on non-interactive DP data release are application-specific or limited by the data-types they are compatible with. Thus, we discuss these two types separately.

2.1.2.1 Application-Specific

These models are designed for specific applications or data-types. For example, the works by Sala et al. [106] and by Proserpio et al. [98] are for graph analysis, the system by Ororbia et al. [94] is for plaintext statistics, the analysis by Machanavajjhala et al. [81] is for commuting pattern analysis, the Binomial-Beta model by McClure and Reiter [82] and Bayesian-network by Zhang et al. [134] are for binary data, and the LDA model by Jiang et al. [66] is for binary classification. In contrast, in this work, we aim at designing an application-independent generative model.

2.1.2.2 Application-Independent

These generative models are less common, possibly due to the fact that releasing data for general analytics often requires a high level of perturbation that impacts data utility. Bindschaedler et al. [11] design a system for *plausible deniability*, which can be extended to (ϵ, δ)-differential privacy. Acs et al. [3] design a system based on two steps – kernel K-means clustering and generative neural networks – to similarly provide (ϵ, δ)-differential privacy. In contrast, our work aims at providing the strictly stronger ϵ-differential privacy. Another previous method is MODIPS by Liu [80], which applies statistical models based on the concept of sufficient statistics to capture the distribution of the data, and then synthesizes the data from the differentially-private models. This general idea is, in fact, closely related to the Gaussian generative model employed in this work. However, the important distinction is that MODIPS is not accompanied by dimensionality reduction – a step which will be shown to enhance the utility of released data via the DFM effect. For comparison, we implement MODIPS in our experiments and show the improvement achievable by RON-Gauss.

2.2 Dimensionality Reduction and Differential Privacy

Traditionally, data partition and aggregation [28, 31, 57, 58, 60, 61, 75, 93, 98–100, 130, 133, 136] have been applied to enhance data utility in differential privacy. In contrast, our work utilizes the DFM effect for the utility enhancement, of which an important component is *dimensionality reduction (DR)* using the RON projection. We present previous works pertaining to the use of DR in DP here. However, although previous works have explored the use of DR to directly provide DP or to reduce the sensitivity of the query, our work, in contrast, uses DR primarily to trigger the DFM effect for enhancing data utility in the non-interactive setting.

2.2.1 Random Projection

For suitable query functions, random projection has been shown to preserve differential privacy [13, 119, 120]. Alternatively, random projection has also been used to enhance the utility of differential privacy. Multiple types of random projections have been used with the identity query for non-interactive DP data release for both purposes. For example, Blocki et al. [13], Kenthapadi et al. [68], Zhou et al. [137], and Xu et al. [132] use a random matrix whose entries are i.i.d. Gaussian, whereas Li et al. [78] use i.i.d. Bernoulli entries. However, there are three main contrasts to our work. (1) While Blocki et al. [13] use random projection to preserve differential privacy, we use random projection *to enhance utility* via the DFM effect. (2) Instead of i.i.d. Gaussian or Bernoulli entries, we use *random or-*

thonormal (RON) projection to ensure the DFM effect as proved by Meckes [87]. (3) *As opposed to our approach, none of the previous random-projection methods couples DR with a generative model.* We will experimentally compare our work with the method by Li et al. [78], and show that, by exploiting the Gaussian phenomenon via the DFM effect, we achieve significant utility gain.

2.2.2 Other Dimensionality Reduction Methods

Other DR methods have also been used with the identity query to enhance data utility including PCA [66], wavelet transform [128], and lossy Fourier transform [2]. In contrast, our DR is coupled with a generative model, rather than used with the identity query. We experimentally compare our work with the PCA method by Jiang et al. [66] and show that our use of the generative model yields significant improvement.

3 Preliminaries

In this section, we discuss important background concepts related to our work.

3.1 Database Notation

We refer to the database as the *dataset*, which contains n records (samples), each with m real-valued attributes (features) – although, our approach is also compatible with categorical features since they can be converted to real values with encoding techniques [126]. With this setup, the dataset can be represented by the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, whose column vectors $\mathbf{x}_j$ are the samples, with $x_j(i)$ refers to the i^{th} feature. Finally, random variables are denoted by a regular capital letter, e.g. Z, which may refer to a random scalar, vector, or matrix. The reference will be clear from the context.

3.2 Differential Privacy (DP)

Differential privacy (DP) protects against the inference of the participation of a sample in the dataset as follows.

Definition 1 (ϵ-DP)**.** A mechanism $\mathcal{A}$ on a query function $f(\cdot)$ preserves ϵ- differential privacy if for all neighboring pairs $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{m\times n}$ which differ in a single record and for all possible measurable outputs $\mathbf{S} \subseteq \mathcal{R}$,

$$\frac{\Pr[\mathcal{A}(f(\mathbf{X})) \in \mathbf{S}]}{\Pr[\mathcal{A}(f(\mathbf{X}')) \in \mathbf{S}]} \leq \exp(\epsilon).$$

Remark 1. There is also the (ϵ, δ)-differential privacy ((ϵ, δ)-DP) [30, 37], which is a relaxation of this definition. However, this work focuses primarily on the stronger ϵ-DP.

Our approach employs the Laplace mechanism, which uses the notion of L_1-sensitivity. For a general query function whose output can be a $p \times q$ matrix, the L_1-sensitivity is defined as follows.

Definition 2. The L_1-sensitivity of a query function $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ for all neighboring datasets $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{m\times n}$ which differ in a single sample is

$$S(f) = \sup_{\mathbf{X},\mathbf{X}'} \left\| f(\mathbf{X}) - f(\mathbf{X}') \right\|_1 .$$

Remark 2. In DP, the notion of neighboring datasets $\mathbf{X}, \mathbf{X}'$ can be considered in two related ways. The first is the *unbounded* notion when one record is removed or added. The second is the *bounded* notion when values of one record vary. The main difference is that the latter assumes the size of the dataset n is publicly known, while the former assumes it to be private. However, the two concepts are closely related and a mechanism that satisfies one can also satisfy the other with a small cost (cf. [16]). In the following analysis, we adopt the latter notion for clarity and mathematical simplicity.

The main tool for DP guarantee in this work is the Laplace mechanism, which is recited as follows [39, 41].

Theorem 1. *For a query function $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ with the L_1-sensitivity $S(f)$, the following mechanism preserves ϵ-differential privacy:*

$$San(f(\mathbf{X})) = f(\mathbf{X}) + Z,$$

where $Z \in \mathbb{R}^{p\times q}$ with $z_j(i)$ drawn i.i.d. from the Laplace distribution $Lap(S(f)/\epsilon)$.

3.3 Gaussian Generative Model

Gaussian generative model synthesizes the data from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, which is parameterized by the mean $\boldsymbol{\mu} \in \mathbb{R}^m$, and the covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ [90, 121]. Formally, the Gaussian generative model has the following density function:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^m \det(\boldsymbol{\Sigma})}} \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})). \tag{1}$$

Hence, to obtain the Gaussian generative model for our application, we only need to estimate its mean and covariance. This reduces the difficult problem of data

modeling into the much simpler one of statistical estimation. This is particularly important in DP since it has been shown that simple statistics of the database can be privately learned accurately [38, 42]. In addition, this model is supported by the following rationales.

- It is supported by the Diaconis-Freedman-Meckes (DFM) effect [87], which may be viewed as an analog of the Central Limit Theorem (CLT) [54, 101] in the feature space. This effect will be discussed in detail in Section 3.4.
- It is simple to use and well-understood. Sampling from it is straightforward, since there exist multiple available packages, e.g. [22, 49, 111].
- Various methods in data analysis have both directly and indirectly utilized the Gaussian model, e.g. linear/quadratic discriminant analysis [46, 65], PCA [63][90, chapter 12], Gaussian Bayes net [90, chapter 10], Gaussian Markov random field [105], Restricted Boltzmann machine (RBM) network [105], Radial Basis Function (RBF) network [19], factor analysis [90, chapter 12], and SVM [96].

In spite of these advantages, we acknowledge that there is possibly no single parametric model that can universally capture all possible datasets, and generalizing our approach to non-parametric generative models is an interesting future work.

3.4 Diaconis-Freedman-Meckes (DFM)

Intuitively, the Diaconis-Freedman-Meckes (DFM) effect – initially proved by Diaconis and Freedman in 1984 [33] – states that *"under suitable conditions, most projections are approximately Gaussian"*. Later, the precise statement of the conditions and the appropriate projections has been proved by Meckes [86, 87], and the phenomenon has been substantiated both theoretically [55] and empirically [20, 92]. This work considers the theorem by Meckes [87] as follows.

Theorem 2 (DFM effect [87, Corollary 4])**.** *Let* $\mathbf{W} \in \mathbb{R}^{m\times p}$ *be a random projection matrix with orthonormal columns,* $X \in \mathbb{R}^m$ *be data drawn i.i.d. from an unknown distribution* $\mathcal{X}$*, which satisfies the regularity conditions in Table 1, and let* $\widetilde{X} = \mathbf{W}^T X \in \mathbb{R}^p$ *be the projection of* X *via* $\mathbf{W}$*, which has the distribution* $\widetilde{\mathcal{X}}$*. Then, for* $p \ll m$*, with high probability,*

$$\widetilde{\mathcal{X}} \overset{d_{BL}}{\approx} \sigma\mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ *is the standard multi-variate Gaussian distribution with* p *dimensions,* σ *is the variance of the Gaussian distribution, and* $\overset{d_{BL}}{\approx}$ *is the approximate equality in distribution with respect to the conditional bounded-Lipschitz distance.*

Let the data $X \in \mathbb{R}^m$ be drawn from a distribution $\mathcal{X}$, which satisfies:

$$\mathbb{E}\left[\|X\|^2\right] = \sigma^2 m,$$
$$\sup_{\mathbf{v}\in\mathbb{S}^{m-1}} \mathbb{E}\left\langle \mathbf{v}, X\right\rangle^2 \leq 1,$$
$$\mathbb{E}\left[\left|\|X\|^2 \sigma^{-2} - m\right|\right] \leq c\sqrt{m}.$$

Table 1. The regularity conditions for the DFM effect. σ is the variance defined in Theorem 2, $c > 0$ is a constant, and $\mathbb{S}^{m-1}$ is the topological sphere (cf. [125]).

Remark 3. The conditional bounded-Lipschitz distance d_{BL} is a distance metric, which can be used to measure the similarity between distributions. More detail on d_{BL} can be found in [103]. Here, the notion $\overset{d_{BL}}{\approx}$ is used to indicate that the distance between $\widetilde{\mathcal{X}}$ and $\sigma\mathcal{N}(\mathbf{0}, \mathbf{I})$ is bounded by a small value [87].

Theorem 2 suggests that, under the regularity conditions, most *random orthonormal (RON) projections* of the data are close to Gaussian in distribution. More specifically, Meckes [87] suggests that the Gaussian phenomenon generally occurs for $p < \frac{2\log(m)}{\log(\log(m))}$. For example, if the original dimension of the dataset is $m = 100$, the projected data would approach Gaussian with $p \leq 13$. Intuitively, the regularity conditions assure that the data are well-spread around the mean with a finite second moment. The convergence to the standard Gaussian also implies that the mean of X is zero. However, this is less critical since d_{BL} is a distance metric [103], so mean-shift can be shown to result in a Gaussian with a scaled mean-shift (cf. [86, 103]).

4 RON-Gauss: Exploiting the DFM Effect for Non-Interactive Differential Privacy

Based on the DFM effect discussed in Section 3.4, we present our approach for the non-interactive DP data release: the *RON-Gauss* model. In the subsequent discussion, we first give an overview of the RON-Gauss model. Then, we discuss the approach used in RON-Gauss with corresponding theoretical analyses. Finally, we present algorithms to implement RON-Gauss for both unsuper-

vised and supervised learning tasks, and prove that the data generated from RON-Gauss preserve ϵ-DP.

4.1 Overview

RON-Gauss stands for ***R**andom **O**rtho**N**ormal projection with **GAUSS**ian generative model*. As its name suggests, RON-Gauss has two components - dimensionality reduction (DR) via random orthonormal (RON) projection, and parametric data modeling via the Gaussian generative model. Each component plays an important role in RON-Gauss as follows.

The DR via random orthonormal (RON) projection has two purposes. First, as previous works have shown [13, 66, 68, 78, 128, 137], DR can reduce sensitivity of the data. This is true for many DR techniques. However, in this work, we choose the RON projection due to the second purpose of DR in the RON-Gauss model, i.e. to trigger the DFM effect. This is verified by Theorem 2 as proved by Meckes [87].

The parametric modeling via the Gaussian generative model also has two purposes. First, it allows us to fully exploit the DFM effect since, unlike most practical data-analytic settings, we know the distribution of the data from the effect. Second, it allows us to reduce the difficult problem of non-interactive private data release into the more amenable one of DP statistical estimation. Particularly, it reduces the problem into that of privately estimating the mean and covariance – a problem which has seen success in DP literature (cf. [15, 36, 38, 39, 41–43, 124]).

Combining these two components is crucial for getting high utility from the released data, as we will demonstrate in our experiments, and we highlight the main differences between our approach and previous works in non-interactive DP data release as follows.

- Although prior works have used DR for improving utility of the released data (cf. Section 2.2), these works do not use the Gaussian generative model. Hence, they do not fully exploit the DFM effect.
- Similarly, there have been prior works that use generative models for synthesizing private data (cf. Section 2.1). However, without DR, the sensitivity is generally large for high-dimensional data, and, more importantly, the DFM effect does not apply.
- Unlike the work by Blocki et al. [13], we do not use random projection to provide DP. In RON-Gauss, DP is provided after the projection via the Laplace mechanism *on the Gaussian generative model*.
- Unlike previous works that use i.i.d. Gaussian or Bernoulli random projection [68, 78, 132, 137], we use RON projection, which has been proved to be suitable for the DFM effect (Theorem 2).

Finally, we acknowledge that although RON-Gauss is designed to be application-independent, it may not be suitable for every task. In this work, we focus on popular machine learning tasks including clustering, regression, and classification. As discussed in Section 3.3, many machine learning algorithms implicitly or explicitly utilize the Gaussian model, so RON-Gauss is generally suitable for these applications.

4.2 Approach and Theoretical Analysis

The RON-Gauss model uses the following steps:

1. Pre-processing to satisfy the conditions for the DFM effect (Theorem 2).
   (a) Pre-normalization.
   (b) Data centering.
   (c) Data re-normalization.
2. RON projection.
3. Gaussian generative model estimation.

We provide the detail of each process as follows.

4.2.1 Data Pre-Processing

Given a dataset with n samples and m features, to utilize the DFM effect, we want to ensure that the data satisfy the regularity conditions of Theorem 2. We show that the following *sample-wise normalization* ensures the conditions are satisfied.

Lemma 1 (Sample-wise normalization)**.** *Let $D \in \mathbb{R}^m$ be data drawn i.i.d. from a distribution $\mathcal{D}$. Let $X \in \mathbb{R}^m$ be derived from D by the sample-wise normalization*[1]*:*

$$X = \frac{D}{\|D\|}.$$

Then,

$$\mathbb{E}\left[\|X\|^2\right] = 1,$$
$$\sup_{\mathbf{v}\in\mathbb{S}^{m-1}} \mathbb{E}\left\langle \mathbf{v}, X\right\rangle^2 \leq 1,$$
$$\mathbb{E}\left[\left|\|X\|^2 \sigma^{-2} - m\right|\right] = \left|\sigma^{-2} - m\right|.$$

Proof. The first and third equalities are obvious by observing that $\|X\| = 1$. The second inequality follows

1 Here, it is implicitly assumed that $\|D\|$ is finite, which is typically the case when we are given a training dataset.

from the Cauchy–Schwarz inequality [25, 109] as follows. $\mathbb{E}\langle \mathbf{v}, X\rangle^2 \leq \mathbb{E}\left[\|\mathbf{v}\|^2 \|X\|^2\right] = 1$, since $\mathbf{v}$ is on the surface of the unit sphere. □

From this lemma, we can verify that the sample-wise normalization satisfies the regularity conditions in Theorem 2 simply by considering $\sigma = 1/\sqrt{m}$. Recall from Theorem 2 that the choice of σ indicates what the variance of the projected data will be. In other words, a low value of σ geometrically signifies a narrow bell curve of Gaussian. Hence, in our application, this is the normalization we employ. However, we observe that this normalization has an effect of placing all data samples onto the surface of the sphere, i.e. $\mathbf{x}_i \in \mathbb{S}^{m-1}$. Hence, it is beneficial to center the data before performing this normalization to ensure that the data remain well-spread after the normalization. For this reason, data pre-processing for RON-Gauss consists of three steps – pre-normalization, data centering, and data re-normalization. As we will discuss shortly, the pre-normalization is to aid with the sensitivity derivation of the sample mean used in the centering process, while the re-normalization is to ensure the regularity conditions in Theorem 2 is satisfied before the projection. These three steps are discussed in detail as follows.

4.2.1.1 Pre-Normalization

We start with a given dataset $\mathbf{X} \in \mathbb{R}^{m\times n}$ with n samples and m features, and perform the preliminary sample-wise normalization as follows.

$$\mathbf{x}_i := \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|},$$

for all $\mathbf{x}_i \in \mathbf{X}$. This normalization ensures that $\|\mathbf{x}_i\| = 1$ for every sample, which will be important for the derivation of the L_1-sensitivity in the next step.

4.2.1.2 Data Centering

Data centering is performed before RON projection in order to reduce the bias of the covariance estimation for the Gaussian generative model and to ensure that the data are well-spread. Data centering is achieved simply by subtracting the DP-mean of the dataset. Given the pre-normalized dataset, $\{\mathbf{x}_i \in \mathbb{R}^m; \|\mathbf{x}_i\| = 1\}_{i=1}^n$, the sample mean is $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$, and the L_1-sensitivity of the sample mean can be computed as follows.

Lemma 2. *Given a sample-wise normalized dataset $\mathbf{X} \in \mathbb{R}^{m\times n}$, the L_1-sensitivity of the sample mean is $s(f) = 2\sqrt{m}/n$.*

Proof. For neighboring datasets $\mathbf{X}$, $\mathbf{X}'$,

$$\begin{aligned} s(f) &= \sup_{\mathbf{X},\mathbf{X}'} \left\|f(\mathbf{X}) - f(\mathbf{X}')\right\|_1 \\ &= \sup \frac{1}{n}\left\|\mathbf{x}_i - \mathbf{x}_i'\right\|_1 \leq \sup \frac{\sqrt{m}}{n}\left\|\mathbf{x}_i - \mathbf{x}_i'\right\|_F \\ &\leq \sup \frac{\sqrt{m}}{n}(\|\mathbf{x}_i\|_F + \left\|\mathbf{x}_i'\right\|_F) = \frac{2\sqrt{m}}{n}, \end{aligned}$$

where the first inequality uses the norm relation [62, page 333]. □

With the L_1-sensitivity of the sample mean, we can then derive DP-mean $\boldsymbol{\mu}^{DP}$ via the Laplace mechanism (Theorem 1), and perform data centering by $\bar{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}^{DP}\mathbf{1}^T$, where $\mathbf{1}$ is the vector with all ones. We note that, although the mean is DP-protected, the centered data are not DP-protected, so they cannot be released. RON-Gauss only uses the centered data to estimate the covariance, which is then DP-protected. Hence, the centered data are never published. In addition, as will be important to the DP analysis later, we note that this centering process ensures that any neighboring datasets would be centered by the same mean. Hence, neighboring $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$ would still differ by only one record.

4.2.1.3 Data Re-Normalization

After adjusting the mean, the centered dataset $\bar{\mathbf{X}}$ would likely not remain normalized. Hence, to ensure the regularity conditions in Theorem 2, we re-normalize the data after the centering process using the same sample-wise normalization scheme, i.e.

$$\bar{\mathbf{x}}_i := \frac{\bar{\mathbf{x}}_i}{\|\bar{\mathbf{x}}_i\|}.$$

Hence, we again have $\|\bar{\mathbf{x}}_i\| = 1$ for every sample. In addition, neighboring datasets still differ by only one sample since the normalization factor only depends on the corresponding sample, but not on any other sample.

4.2.1.4 Summary

Data_Preprocessing (Algorithm 1) summarizes these steps for pre-processing the data, which include pre-normalizing, centering, and re-normalizing. The DP mean derivation uses the Laplace mechanism with the sensitivity in Lemma 2. If needed, the DP mean can also be acquired from this algorithm.

4.2.2 RON Projection

As shown in Lemma 1, after the pre-processing steps, $\bar{\mathbf{X}}$ can be shown to be in a form that complies with the

Algo. 1 DATA_PREPROCESSING

Input: dataset $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\epsilon_\mu > 0$.

1. Pre-normalize: $\mathbf{x}_i := \mathbf{x}_i / \|\mathbf{x}_i\|$ for all $\mathbf{x}_i \in \mathbf{X}$.
2. Derive the DP mean: $\boldsymbol{\mu}^{DP} = (\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i) + Z$, where $z_j(i)$ is drawn i.i.d. from $Lap(2\sqrt{m}/n\epsilon_\mu)$.
3. Center the data: $\bar{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}^{DP}\mathbf{1}^T$.
4. Re-normalize: $\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_i / \|\bar{\mathbf{x}}_i\|$ for all $\bar{\mathbf{x}}_i \in \bar{\mathbf{X}}$.

Output: $\bar{\mathbf{X}}$, $\boldsymbol{\mu}^{DP}$.

Algo. 2 RON_PROJECTION

Input: pre-processed dataset $\bar{\mathbf{X}} \in \mathbb{R}^{m \times n}$, and dimension $p < m$.

1. Form a matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ whose elements are drawn i.i.d. from the uniform distribution.
2. Factorize $\mathbf{A}$ via the QR factorization as $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{m \times m} : \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$.
3. Construct a RON projection matrix $\mathbf{W} = [\mathbf{q}_1, \ldots, \mathbf{q}_p] \in \mathbb{R}^{m \times p}$.
4. Project the data: $\widetilde{\mathbf{X}} = \mathbf{W}^T\bar{\mathbf{X}} \in \mathbb{R}^{p \times n}$.

Output: $\widetilde{\mathbf{X}} \in \mathbb{R}^{p \times n}$, $\mathbf{W}$.

regularity conditions in Theorem 2. The next step is to project $\bar{\mathbf{X}}$ onto a low-dimensional space using the random orthonormal (RON) projection matrix $\mathbf{W} \in \mathbb{R}^{m \times p} : \mathbf{W}^T\mathbf{W} = \mathbf{I}$. The RON projection matrix is derived independently of the dataset, so it does not leak privacy, and no privacy budget is needed for its acquisition.

The projection is done via the linear transformation $\widetilde{\mathbf{x}}_i = \mathbf{W}^T\bar{\mathbf{x}}_i \in \mathbb{R}^p$ for each sample, or, equivalently, in the matrix notation $\widetilde{\mathbf{X}} = \mathbf{W}^T\mathbf{X} \in \mathbb{R}^{p \times n}$. Since the projection is done sample-wise, the neighboring datasets would still differ by only one sample after the projection. This property will be important to the DP analysis of the RON projection later. In addition, the other important theoretical aspect of the RON projection step is the bound on the projected data. This is provided by the following lemma.

Lemma 3. *Given a normalized data sample $\bar{\mathbf{x}} \in \mathbb{R}^m$ and a random orthonormal (RON) projection matrix, $\mathbf{W} \in \mathbb{R}^{m \times p} : \mathbf{W}^T\mathbf{W} = \mathbf{I}$, let $\widetilde{\mathbf{x}} = \mathbf{W}^T\bar{\mathbf{x}}$ be the projection via $\mathbf{W}$. Then, $\|\widetilde{\mathbf{x}}\|_F \leq 1$.*

Proof. The proof is provided in Appendix A. □

Lemma 3 indicates that the RON projection of the normalized data does not change their Frobenius norm. This will be critical in the DP analysis of RON-Gauss algorithms in the next step.

RON_PROJECTION (Algorithm 2) summarizes the current step that projects the pre-processed data onto a lower dimension p via RON projection. The RON projection matrix can be derived efficiently via the QR factorization [50, 107], as shown in the algorithm. Specifically, the RON projection matrix is constructed by stacking side-by-side p column vectors of the unitary matrix $\mathbf{Q}$ from the QR factorization. Then, the projection is done via the linear operation $\widetilde{\mathbf{X}} = \mathbf{W}^T\bar{\mathbf{X}} \in \mathbb{R}^{p \times n}$. If needed, the RON projection matrix can also be acquired from the output of the algorithm.

We further note that the projected data $\widetilde{\mathbf{X}}$ are still not DP-protected. This is one key difference between our work and that of Blocki et al. [13], which uses random projection to directly provide DP. In our work, $\widetilde{\mathbf{X}}$ is never released, and we only use it to estimate the covariance of the Gaussian generative model in the next step. The DP protection in our work is provided in this next step on the Gaussian generative model.

4.2.3 Gaussian Generative Model Estimation

This step constructs the Gaussian generative model, which is where the DP protection in RON-Gauss is provided. We emphasize that RON-Gauss is an output-perturbation algorithm, and we employ the standard DP threat model, i.e. the RON-Gauss algorithm is run by a trusted entity and only the output of the algorithm is available to the public. The DP-protected Gaussian generative model is then used to synthesize DP dataset for the non-interactive DP data release setting we consider. Synthesizing DP data from a parametric model, as opposed to releasing the model itself, has two benefits. First, existing machine learning software can readily be used with the DP data as if they were the real data. Second, it presents an additional challenge for an attacker aiming to perform inference attacks, since the attacker would also need to estimate the model parameters from the released data, incurring further errors.

Before delving into the detail of this step, there is an important distinction to be made about the data-analytic problems we consider. Since machine learning is currently the prominent tool in data analysis, we follow the convention in machine learning and consider two classes of problems – *unsupervised learning* and *supervised learning*. The main difference between the two is that, in the latter, in addition to the feature data in $\widetilde{\mathbf{X}}$, the *teacher value* or *training label* $\mathbf{y} \in \mathbb{R}^n$ is also required to guide the data-analytic process. Hence, in the subsequent analysis, we first consider the simpler class of unsupervised learning, and then, show a simple modification to include the teacher value for the supervised

learning. Additionally, we conclude with an extension of RON-Gauss to the Gaussian Mixture Model.

4.2.3.1 Unsupervised Learning

The unsupervised learning problems do not require the training label, so the Gaussian generative model only needs to synthesize DP-protected $\widetilde{\mathbf{X}}$. The main parameter for the Gaussian generative model is the covariance matrix $\mathbf{\Sigma}$, so we need to estimate $\mathbf{\Sigma}$ from $\widetilde{\mathbf{X}}$. We use the following formulation for the sample covariance:

$$\mathbf{\Sigma} = \frac{1}{n}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T = \frac{1}{n}\sum_{i=1}^{n}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T \in \mathbb{R}^{p\times p}. \tag{2}$$

We note that this estimate may be statistically biased since the mean may not necessarily be zero after the re-normalization. However, this formulation would yield significantly lower sensitivity than its unbiased counterpart. This is due to the observation that only one summand can change for neighboring datasets since, as mentioned in the previous step, neighboring projected datasets $\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}'}$ still differ by only one sample. Hence, we are willing to trade the bias for a much lower sensitivity. In Appendix B, we specifically show that the saving in sensitivity by our formulation is in the order of n, compared to the MLE of the covariance. Clearly, for large datasets, this is significant and can be the difference between usable and unusable models.

Next, we derive the sensitivity of the covariance estimate in Eq. (2) as follows.

Lemma 4. *Given a dataset* $\mathbf{X} \in \mathbb{R}^{m\times n}$*, let* $\widetilde{\mathbf{X}}$ *be the pre-processed and RON-projected dataset via* Data_Preprocessing *and* RON_Projection*. Then, the covariance* $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$ *in Eq.* (2) *has the* L_1*-sensitivity of* $2\sqrt{p}/n$*.*

Proof. For neighboring datasets $\mathbf{X}, \mathbf{X}'$,

$$\begin{aligned}
s(f) &= \sup \left\| \frac{1}{n}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T - \frac{1}{n}\widetilde{\mathbf{X}'}\widetilde{\mathbf{X}'}^T \right\|_1 \\
&= \sup \Bigg\| \frac{1}{n}\mathbf{W}^T[\mathbf{X} - \boldsymbol{\mu}^{DP}\mathbf{1}^T][\mathbf{X} - \boldsymbol{\mu}^{DP}\mathbf{1}^T]^T\mathbf{W} \\
&\quad - \frac{1}{n}\mathbf{W}^T[\mathbf{X}' - \boldsymbol{\mu}^{DP}\mathbf{1}^T][\mathbf{X}' - \boldsymbol{\mu}^{DP}\mathbf{1}^T]^T\mathbf{W} \Bigg\|_1, \\
&= \sup \frac{1}{n} \Bigg\| \sum_{i=1}^{n} \mathbf{W}^T[\mathbf{x}_i - \boldsymbol{\mu}^{DP}][\mathbf{x}_i - \boldsymbol{\mu}^{DP}]^T\mathbf{W} \\
&\quad - \sum_{i=1}^{n} \mathbf{W}^T[\mathbf{x}'_i - \boldsymbol{\mu}^{DP}][\mathbf{x}'_i - \boldsymbol{\mu}^{DP}]^T\mathbf{W} \Bigg\|_1,
\end{aligned}$$

where the second equality is simply from the definition of $\widetilde{\mathbf{X}}$ through Data_Preprocessing and RON_Projection. Since all of the summands are the same except for one in the neighboring datasets, we have

$$\begin{aligned}
s(f) = \sup \frac{1}{n} \Big\| &\mathbf{W}^T[\mathbf{x}_i - \boldsymbol{\mu}^{DP}][\mathbf{x}_i - \boldsymbol{\mu}^{DP}]^T\mathbf{W} \\
&- \mathbf{W}^T[\mathbf{x}'_i - \boldsymbol{\mu}^{DP}][\mathbf{x}'_i - \boldsymbol{\mu}^{DP}]^T\mathbf{W} \Big\|_1.
\end{aligned}$$

Then, to simplify the notation and to apply Lemma 3, we note that $\widetilde{\mathbf{x}}_i = \mathbf{W}^T[\mathbf{x}_i - \boldsymbol{\mu}^{DP}]$ by definition. Hence,

$$\begin{aligned}
s(f) &= \sup \frac{1}{n} \left\| \widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T - \widetilde{\mathbf{x}'}_i\widetilde{\mathbf{x}'}_i^T \right\|_1 \\
&\le \sup \frac{\sqrt{p}}{n} \left\| \widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T - \widetilde{\mathbf{x}'}_i\widetilde{\mathbf{x}'}_i^T \right\|_F \\
&\le \sup \frac{2\sqrt{p}}{n} \left\| \widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T \right\|_F \le \sup \frac{2\sqrt{p}}{n} \left\| \widetilde{\mathbf{x}}_i \right\|_F^2 = \frac{2\sqrt{p}}{n},
\end{aligned}$$

where the first inequality uses the norm relation [62, page 333], and the last equality uses Lemma 3. □

Lemma 4 provides an important insight into the RON-Gauss model. As discussed in the overview (Section 4.1), the RON projection step of RON-Gauss serves two purposes – to initiate the DFM effect, and to reduce the sensitivity of the model. The latter purpose can clearly be observed from Lemma 4. Specifically, Lemma 4 indicates that the L_1-sensitivity of the main parameter of the RON-Gauss model, i.e. the covariance, reduces as the dimension p reduces. This is particularly attractive when the original data are very high-dimensional as the noise added by the Laplace mechanism could be greatly reduced. For example, for the original data with 100 dimensions, the RON projection onto a 10-dimensional subspace would reduce the sensitivity by about 3x.

With the L_1-sensitivity derived, the Laplace mechanism in Theorem 1 can be used to derive the DP covariance matrix: $\mathbf{\Sigma}^{DP}$. With $\mathbf{\Sigma}^{DP}$, RON-Gauss then generates the synthetic DP data from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{DP})$. If the mean is needed, we can readily use $\mathbf{W}^T\boldsymbol{\mu}^{DP}$, which already satisfies DP due to the post-processing invariance of DP [36]. This completes the RON-Gauss model for unsupervised learning.

Algorithm 3 summarizes the RON-Gauss model for unsupervised learning. First, the data are pre-processed via Data_Preprocessing (Algorithm 1). The DP mean derivation in Data-Preprocessing spends the privacy budget ϵ_μ. Second, the data are projected onto a lower dimension p via RON_Projection. Third, the algorithm derives the DP covariance using the Laplace mechanism with the sensitivity derived in Lemma 4. Finally, the algorithm synthesizes DP data by drawing samples from the Gaussian generative model parametrized by the DP covariance. We conclude the discussion on RON-Gauss for unsupervised learning with the DP analysis of Algorithm 3.

Algo. 3 RON-Gauss for unsupervised learning

Input: dataset $\mathbf{X} \in \mathbb{R}^{m\times n}$, dimension $p < m$, and $\epsilon_\mu, \epsilon_\Sigma > 0$.

1. Obtain the pre-processed data $\bar{\mathbf{X}} \in \mathbb{R}^{m\times n}$ from DATA_PREPROCESSING with inputs $\mathbf{X}$ and ϵ_μ.
2. Obtain the RON-projected data $\widetilde{\mathbf{X}} \in \mathbb{R}^{p\times n}$ from RON_PROJECTION with inputs $\bar{\mathbf{X}}$ and p.
3. Derive the DP covariance: $\mathbf{\Sigma}^{DP} = (\frac{1}{n}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T) + Z$, where $z_j(i)$ is drawn i.i.d. from $Lap(2\sqrt{p}/n\epsilon_\Sigma)$.
4. Synthesize DP data $\mathbf{x}_i^{DP} \in \mathbb{R}^p$ by drawing samples from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{DP})$.

Output: $\{\mathbf{x}_1^{DP}, \ldots, \mathbf{x}_{n'}^{DP}\}$.

Theorem 3. *Algorithm 3 preserves* $(\epsilon_\mu + \epsilon_\Sigma)$*-differential privacy.*

Proof. The proof follows the following induction. The DP data $\mathbf{x}_i^{DP}$ are derived from only one source, i.e. the Gaussian generative model of RON-Gauss. Based on the post-processing invariance of DP, if the model is DP-protected, then the released data are also similarly DP-protected. The Gaussian generative model is parametrized by $\mathbf{\Sigma}^{DP}$, which is DP-protected. Specifically, the $\mathbf{\Sigma}^{DP}$ computation in step 3(a) spends ϵ_Σ privacy budget with the Laplace mechanism, according to Theorem 1. However, the centering process in step 1(c) also spends ϵ_μ privacy budget on the Laplace mechanism to derive $\boldsymbol{\mu}^{DP}$, which assists in the $\mathbf{\Sigma}^{DP}$ derivation process. Due to the serial composition theorem [36], the two privacy budgets add up. Hence, $\mathbf{\Sigma}^{DP}$ preserves $(\epsilon_\mu+\epsilon_\Sigma)$-differential privacy, and, consequently, the Gaussian generative model preserves $(\epsilon_\mu+\epsilon_\Sigma)$-differential privacy, so do the synthesized data. □

4.2.3.2 Supervised Learning

The unsupervised learning does not involve the guidance from the training label. However, in supervised learning, the training label $\mathbf{y}$ is also required. Hence, the Gaussian generative model needs to be modified to incorporate the training label into the model in order to synthesize both DP-protected $\widetilde{\mathbf{X}}$ and $\mathbf{y}$.

A simple method to incorporate the training label into the Gaussian generative model is to treat it as another feature. However, when RON projection is applied, it should only be applied to the feature data, but *not to the training label*. This is because when the projection is applied, each induced feature is a linear combination of all original features. Therefore, if RON projection is also applied to the training label, the integrity of the training label would be spoiled. In other words, we may not be able to extract the training label from the projected data. Thus, to preserve the integrity of the training label, it should *not* be modified by the RON projection process.

In RON-Gauss, the aforementioned challenge in supervised learning is navigated by augmenting the data matrix with the training label as $\mathbf{X}_a = \begin{bmatrix}\widetilde{\mathbf{X}}\\ \mathbf{y}^T\end{bmatrix} \in \mathbb{R}^{(p+1)\times n}$. Then, the augmented covariance matrix can be written in block form as,

$$\mathbf{\Sigma}_a = \frac{1}{n}\begin{bmatrix}\widetilde{\mathbf{X}}\\ \mathbf{y}^T\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{X}}^T & \mathbf{y}\end{bmatrix} = \frac{1}{n}\begin{bmatrix}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T & \widetilde{\mathbf{X}}\mathbf{y}\\ \mathbf{y}^T\widetilde{\mathbf{X}}^T & \mathbf{y}^T\mathbf{y}\end{bmatrix}. \quad (3)$$

This can then be used in a similar fashion to the covariance matrix in Eq. (2) for unsupervised learning. We note that, again, this may not be an unbiased estimate of the augmented covariance matrix since the mean may not necessarily be zero, but, similar to the unsupervised learning design, it has significantly lower sensitivity than the unbiased counterpart. Therefore, we are willing to trade the bias for achieving small sensitivity. Given the training label with bounded value[2] $\mathbf{y} \in [-a, a]^n$, the sensitivity of the augmented covariance matrix can be derived as follows.

Lemma 5. *The* L_1*-sensitivity of the augmented covariance matrix in Eq.* (3) *is* $\frac{2\sqrt{p}+4a\sqrt{p}+a^2}{n}$.

Proof. For neighboring datasets $\mathbf{X}, \mathbf{X}'$, the sensitivity is

$$\begin{aligned} S(\mathbf{\Sigma}_a) &= \sup\frac{1}{n}\left\|\begin{bmatrix}\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T & \widetilde{\mathbf{X}}\mathbf{y}\\ \mathbf{y}^T\widetilde{\mathbf{X}}^T & \mathbf{y}^T\mathbf{y}\end{bmatrix} - \begin{bmatrix}\widetilde{\mathbf{X}'}\widetilde{\mathbf{X}'}^T & \widetilde{\mathbf{X}'}\mathbf{y}'\\ \mathbf{y}'^T\widetilde{\mathbf{X}'}^T & \mathbf{y}'^T\mathbf{y}'\end{bmatrix}\right\|_1,\\ &= \sup\frac{1}{n}\Big(\left\|\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T - \widetilde{\mathbf{X}'}\widetilde{\mathbf{X}'}^T\right\|_1 + 2\left\|\widetilde{\mathbf{X}}\mathbf{y} - \widetilde{\mathbf{X}'}\mathbf{y}'\right\|_1\\ &\quad + \left\|\mathbf{y}^T\mathbf{y} - \mathbf{y}'^T\mathbf{y}'\right\|_1\Big). \end{aligned}$$

The proof then considers each summand separately. The first summand is the sensitivity of $\mathbf{\Sigma}$ in Eq. (2), so it is $2\sqrt{p}/n$. The last summand can be written as,

$$\begin{aligned}\sup\frac{\left\|\mathbf{y}^T\mathbf{y} - \mathbf{y}'^T\mathbf{y}'\right\|_1}{n} &= \sup\frac{\left\|\sum_{i=1}^n y(i)^2 - \sum_{i=1}^n y'(i)^2\right\|_1}{n}\\ &= \sup\frac{\left\|y(i)^2 - y'(i)^2\right\|_1}{n} = \frac{a^2}{n},\end{aligned}$$

where the second equality is because only one element in $\mathbf{y}$ and $\mathbf{y}'$ differs.

2 As suggested by Liu [80], real-world data are often bounded, and the bounded-valued assumption is often made in DP analysis for multi-dimensional query (cf. [27, 39, 43, 137]).

Algo. 4 RON-Gauss for supervised learning

Input: dataset with training labels $\mathbf{X} \in \mathbb{R}^{m\times n}, \mathbf{y} \in [-a, a]^n$, dimension $p < m$, and $\epsilon_\mu, \epsilon_\Sigma > 0$.

1. Obtain the pre-processed data $\bar{\mathbf{X}} \in \mathbb{R}^{m\times n}$ from DATA_PREPROCESSING with inputs $\mathbf{X}$ and ϵ_μ.
2. Obtain the RON-projected data $\widetilde{\mathbf{X}} \in \mathbb{R}^{p\times n}$ from RON_PROJECTION with inputs $\bar{\mathbf{X}}$ and p.
3. Form the augmented data matrix $\mathbf{X}_a = \begin{bmatrix} \widetilde{\mathbf{X}} \\ \mathbf{y}^T \end{bmatrix} \in \mathbb{R}^{(p+1)\times n}$.
4. Derive the DP augmented covariance: $\mathbf{\Sigma}_a^{DP} = (\frac{1}{n}\mathbf{X}_a\mathbf{X}_a^T) + Z^{(\Sigma)}$, where $z_j^{\Sigma}(i)$ is drawn i.i.d. from $Lap((2\sqrt{p} + 4a\sqrt{p} + a^2)/n\epsilon_\Sigma)$.
5. Synthesize DP augmented data $\begin{bmatrix} \mathbf{x}_i^{DP} \\ y(i)^{DP} \end{bmatrix} \in \mathbb{R}^{p+1}$ by drawing samples from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_a^{DP})$.

Output: $\{\mathbf{x}_1^{DP}, \ldots, \mathbf{x}_{n'}^{DP}\}$ with training label $\mathbf{y}^{DP}$.

For the second summand, we have

$$\begin{aligned}
\sup \frac{2\left\|\widetilde{\mathbf{X}}\mathbf{y} - \widetilde{\mathbf{X}'}\mathbf{y}'\right\|_1}{n} &= \sup \frac{2\left\|\sum_i \widetilde{\mathbf{x}}_i y(i) - \sum_i \widetilde{\mathbf{x}}'_i y'(i)\right\|_1}{n} \\
&= \sup \frac{2\left\|\widetilde{\mathbf{x}}_i y(i) - \widetilde{\mathbf{x}}'_i y'(i)\right\|_1}{n} \\
&\leq \sup \frac{2(\|\widetilde{\mathbf{x}}_i y(i)\|_1 + \left\|\widetilde{\mathbf{x}}'_i y'(i)\right\|_1)}{n} \\
&\leq \sup \frac{2\sqrt{p}(\|\widetilde{\mathbf{x}}_i y(i)\|_F + \left\|\widetilde{\mathbf{x}}'_i y'(i)\right\|_F)}{n} \\
&\leq \frac{2\sqrt{p}(2a)}{n} = \frac{4a\sqrt{p}}{n},
\end{aligned}$$

where the second line is from the fact that the other $n-1$ terms are similar for neighboring datasets. By combing the three summands, we have completed the proof. □

As the L_1-sensitivity of $\mathbf{\Sigma}_a$ has been derived, we can use the Laplace mechanism in Theorem 1 to acquire the DP augmented covariance matrix: $\mathbf{\Sigma}_a^{DP}$. Then, similar to the unsupervised learning, RON-Gauss generates the synthetic DP data – which include both the feature data and the training label – from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_a^{DP})$. Notice that the only difference between the RON-Gauss model for unsupervised learning and supervised learning is the use of $\mathbf{\Sigma}^{DP}$ (Eq. (2)) and $\mathbf{\Sigma}_a^{DP}$ (Eq. (3)), respectively.

Algorithm 4 presents the RON-Gauss model for supervised learning. The algorithm is similar to Algorithm 3. The only difference is the use of the augmented covariance matrix in step 4 with the sensitivity from Lemma 5 to incorporate the training label. As a result, Algorithm 4 can synthesize both the DP feature data $\mathbf{x}_i^{DP}$ and the DP training label $\mathbf{y}^{DP}$. Finally, we present the privacy guarantee of Algorithm 4 as follows.

Algo. 5 RON-Gauss' extension to GMM

Input: dataset $\mathbf{X} \in \mathbb{R}^{m\times n}, \mathbf{y} \in \{c_1, c_2, \ldots, c_L\}^n$, dimension $p < m$, and $\epsilon_{\boldsymbol{\mu}}, \epsilon_{\mathbf{\Sigma}} > 0$.

1. **for** c in $\{c_1, \ldots, c_L\}$ **do:**
   (a) Form $\mathbf{X}_c$, whose n_c column vectors are all samples in class c.
   (b) Obtain the pre-processed data $\bar{\mathbf{X}}_c \in \mathbb{R}^{m\times n_c}$ and the DP class-mean $\boldsymbol{\mu}_c^{DP}$ from DATA_PREPROCESSING with inputs $(\mathbf{X}_c, \epsilon_\mu)$.
   (c) Obtain the RON-projected data $\widetilde{\mathbf{X}}_c \in \mathbb{R}^{p\times n_c}$ and the projection matrix $\mathbf{W}$ from RON_PROJECTION with inputs $\bar{\mathbf{X}}_c$ and p.
   (d) Derive the DP class-covariance: $\mathbf{\Sigma}_c^{DP} = (\frac{1}{n_c}\widetilde{\mathbf{X}}_c\widetilde{\mathbf{X}}_c^T) + Z$, where $z_j(i)$ is drawn i.i.d. from $Lap(2\sqrt{p}/n_c\epsilon_\Sigma)$.
   (e) Synthesize DP class-c data $\mathbf{X}_c^{DP}$ by drawing samples from $\mathcal{N}(\mathbf{W}^T\boldsymbol{\mu}_c^{DP}, \mathbf{\Sigma}_c^{DP})$, and assign $\mathbf{y}_c^{DP} = c$ for all samples in $\mathbf{X}_c^{DP}$.
2. Let $\mathbf{X}^{DP} = [\mathbf{X}_{c_1}^{DP}, \ldots, \mathbf{X}_{c_L}^{DP}]$.
3. Let $\mathbf{y}^{DP} = [\mathbf{y}_{c_1}^{DP^T}, \ldots, \mathbf{y}_{c_L}^{DP^T}]^T$.

Output: $\{\mathbf{X}^{DP}, \mathbf{y}^{DP}\}$.

Theorem 4. *Algorithm 4 preserves* $(\epsilon_\mu + \epsilon_\Sigma)$*-differential privacy.*

Proof. The proof mirrors that of Theorem 3 but uses the sensitivity of the augmented covariance in Lemma 5 instead. □

4.2.3.3 Extension to Gaussian Mixture Model

Algorithm 4 for supervised learning uses the unimodal Gaussian generative model. The labels synthesized from this algorithm are numerical. In many applications, e.g. regression, this may already be effective. However, in some applications, e.g. *classification*, it is desirable to synthesize the labels that are discrete or categorical. To this end, we extend RON-Gauss to a multi-modal Gaussian generative model using the *Gaussian Mixture Model* (GMM) [12, 90]. Conceptually, each mode of GMM can be used to capture the distribution of the data in each class. Thus, the entire dataset is modeled by the mixture of these modes. In fact, many classifiers such as Linear Discriminant Analysis (LDA) [46], Bayes Net [90], and mixture of Gaussians [90] also utilize this type of generative model, so this GMM extension has historically been shown to be effective for classification.

In classification, the training label is categorical, i.e. $y \in \{c_1, \ldots, c_L\}$ for L-class classification. Algorithm 5 presents an extension of RON-Gauss to GMM. The algorithm iterates through the data samples in each class. It derives DP samples for each class in a similar procedure to Algorithm 3 with one difference. For GMM, the data

in each class are generated from the Gaussian generative model with the mean equal to the RON-projected DP class-mean, i.e. $\mathbf{W}^T\boldsymbol{\mu}_c^{DP}$. This is to capture the multi-modal nature of GMM. Since every DP sample drawn from each iteration of step 1 belongs to the same class, the same training label is assigned for every synthesized sample. Finally, after iterating through all classes, the algorithm stacks the DP samples and training labels together before releasing the synthesized data.

We note that this algorithm assumes that each data sample belongs to one class only, so each mode of Gaussian is derived from a disjoint set of data. This is the common setting in supervised learning applications (cf. [12, 71, 90, 122]). In addition, to comply with the bounded DP notion we adopt throughout (cf. Remark 2), the algorithm assumes that the number of samples in each class n_c is public information. Finally, we present the DP analysis of Algorithm 5 as follows.

Theorem 5. *Algorithm 5 preserves* $(\epsilon_{\boldsymbol{\mu}} + \epsilon_{\boldsymbol{\Sigma}})$*-differential privacy.*

Proof. Since the data partition is disjoint, and each class has the same domain, the privacy budget used for each class does not add up from the parallel composition [38, 85]. The proof then follows from that of Theorem 3. □

5 Experiments

We demonstrate that RON-Gauss is effective across a range of datasets and machine learning tasks via three experiments. For the datasets, we use a facial expression dataset [32], a sensor displacement dataset [6, 7], and a social media dataset [67]. For the machine learning tasks, we use the clustering, classification, and regression applications. In non-interactive DP, the aim is to release DP data such that the utility of the DP data closely resembles that of the original data. Hence, to evaluate the quality of the non-interactive DP algorithms based on the utility measure commonly used for the respective task (cf. Section 5.2). DP data with high quality, therefore, should provide the values of the utility measure close to that obtained from the non-private data, and our experiments show that RON-Gauss can achieve this objective. We note that we choose task-centric utility measures for our experiments since we want to evaluate the approach based on how much insight can be gained from the synthesized data with respect to each task. However, in other settings, task-independent evaluation metrics such as reconstruction error or mutual information could also be appropriate. We also compare our work to four previous approaches that *solely* relied on either DR, or generative models. Table 2 summarizes the experimental setups, and we discuss them in detail as follows.

5.1 Datasets

5.1.1 Grammatical Facial Expression (GFE)

This dataset is based on facial expression analysis from video images under Libras (a Brazilian sign language), and has 27,936 samples and 301 features [32]. There are multiple clusters based on different grammatical expressions. The image features are designed to be informative of the facial expressions. However, the same features may be used to infer the individuals whose images are included in the dataset. Hence, it is desirable to release a DP-protected dataset. We use this dataset for the privacy-preserving clustering study on Algorithm 3.

5.1.2 Realistic Sensor Displacement (Realdisp)

This is a mobile-sensing dataset used for activity recognition [6, 7]. The features include readings of various motion sensors, and the goal is to identify the activity being performed. However, the same features can possibly be used to identify the individuals whose data are in the dataset. Therefore, it is desirable to release a DP-protected dataset. The dataset consists of 216,752 training samples, and 1,290 testing samples with 117 features. In our experiments, we use this dataset for the privacy-preserving classification study with Algorithm 5. Specifically, we formulate it as a binary classification – identifying whether the subject is performing an action that causes a location displacement or not, e.g. walking, running, cycling, etc.

5.1.3 Buzz in Social Media (Twitter)

This dataset extracts 77 features from Twitter posts, which are used to predict the popularity level of the topic represented as a real value in $[-1, 1]$ [67]. However, these features may also be used to infer the owner of each tweet; thus it is desirable to instead release the DP-protected dataset. The dataset is divided into the training set of 573,820 samples, and the testing set of 4,715 samples. We use this dataset for privacy-preserving regression, and adopt Algorithm 4 for the experiments.

5.2 Setups

Since RON-Gauss algorithms require ϵ_μ and ϵ_Σ for the mean and the covariance, respectively, given a fixed

Exp.	Dataset	Training Size	Feature Size	Metric	ML Alg.	DP Alg.
Clustering	GFE [32]	27,936	301	S.C. (↑better)	K-Means	Alg. 3
Classification	Realdisp [6, 7]	216,752	117	Accuracy (↑better)	SVM	Alg. 5
Regression	Twitter [67]	573,820	77	RMSE (↓better)	KRR	Alg. 4

Table 2. Summary of the experimental setups of the three experiments.

total privacy budget of ϵ, we allocate the budget as: $\epsilon_\mu = 0.3\epsilon$ and $\epsilon_\Sigma = 0.7\epsilon$. The rationale is that the covariance is the more critical parameter in our algorithms, and usually has higher complexity than the mean ($\mathbb{R}^{p\times p}$ vs $\mathbb{R}^m$). For all experiments, we perform 100 trials and report the average with the 95% confidence interval.

5.2.1 Clustering Setup

Clustering is unsupervised learning, so we apply Algorithm 3. We use K-means [71] as the clustering method for its simplicity and efficiency, even for large datasets, and use the *Silhouette Coefficient (S.C.)* [104] as the metric for evaluation. The number of clusters in K-means is set using the Silhouette analysis method [104, 110]. S.C. is defined as follows. For the sample $\mathbf{x}_i$ assigned to class $y(i)$,

- let $a(i)$ be the average distance between $\mathbf{x}_i$ and all other samples assigned to the same class $y(i)$;
- let $b(i)$ be the average distance between $\mathbf{x}_i$ and all points assigned to the next nearest class.

Let $sc(i) = \frac{b(i)-a(i)}{\max\{b(i),a(i)\}}$, and S.C. is defined as:

$$S.C. = \frac{1}{n}\sum_{i=1}^{n} sc(i).$$

Intuitively, S.C. measures the average distance between the sample and its class mean, normalized by the distance to the next nearest class mean. Its range is $[-1, 1]$, where higher value indicates better the performance.

We pick this metric for two reasons. First, as opposed to other metrics including ACC [131], ARI [64], or V-measure [102], S.C. does not require the knowledge of the ground truth. This is vital both for our evaluation and for real-world applications, respectively because the ground truth is not available for the synthetic data in our evaluation, and it is often not available in practice, too. Second, as suggested by Rousseeuw [104], S.C. depends primarily on the distribution of the data, but less on the clustering algorithm used, so it is fitting for the evaluation of non-interactive private data release.

5.2.2 Classification Setup

For classification, we employ the GMM according to Algorithm 5, and use the support vector machine (SVM) [29, 95] as the classifier in all experiments. SVM is chosen since it has been shown to perform well on binary classification [23, 69, 108], and it has been proven – both empirically and theoretically – to generalize well [108, 122]. The evaluation metric is the traditional classification accuracy.

Since we consider the original training data as sensitive, we apply RON-Gauss to generate DP data that are used to train machine learning models. However, we test the machine learning models on the real test data in order to evaluate the ability of the DP training data to capture the classification pattern of the real data.

5.2.3 Regression Setup

We use Algorithm 4 for regression, and use kernel ridge regression (KRR) [71, 95] as the regressor due to its large hypothesis class with proven theoretical error bound [108, 135]. The evaluation metric is the root-mean-square error (RMSE) [12, 71]. Finally, for comparison, we also provide a random-guess baseline of which the prediction is drawn i.i.d. from a uniform distribution. Finally, we manage the train/test split in a similar fashion to the above classification setup (Section 5.2.2).

5.2.4 Comparison to Other Methods

To provide context to the experimental results, we compare our approach to four previous works and a non-private baseline method as follows.

1. Real data: the non-private baseline approach, where the result is obtained from the original data without any modification.
2. Li et al. [78]: the method based on dimensionality reduction via Bernoulli random projection on the identity query.
3. Jiang et al. [66]: the method based on PCA on the identity query.
4. Blum et al. [16]: exponential mechanism for non-interactive setting.
5. Liu [80]: parametric generative model without DR.

We compare RON-Gauss to these five methods for the following reasons. The first comparison is to show the real-world usability of RON-Gauss. The second and

Method	Model	DR	ϵ	S.C.	ΔS.C.
Real data	–	–	–	.286	.00
Li et al. [78]	Identity	Bern. Rand.	1.	.123±.000	.16
Jiang et al. [66]	Identity	PCA	1.	.123±.000	.16
Blum et al. [16]	Exp. Mech.	–	1.	.026±.017	.26
Liu [80]	Gaussian	–	1.	.092±.001	.19
RON-Gauss	Gaussian	RON	1.	.274±.015	.01

Table 3. Clustering results (GFE dataset). ΔS.C. indicates the error relative to the performance by real data.

Method	Model	DR	ϵ	Accuracy (%)
Real data	-	-	-	89.61
Li et al. [78]	Identity	Bern. Rand.	1.	65.04 ± 0.90
Jiang et al. [66]	Identity	PCA	1.	54.51 ± 1.65
Blum et al. [16]	Exp. Mech.	-	1.	51.24 ± 1.83
Liu [80]	GMM	-	1.	61.31 ± 0.65
RON-Gauss	GMM	RON	1.	87.16 ± 0.27

Table 4. Classification results (Realdisp dataset).

third comparisons are to motivate the use of the Gaussian generative model over the identity query, and the remaining comparisons are to motivate DR via the RON projection. For all previous methods, we use the parameters suggested by the respective authors, and we vary the hyper-parameter before reporting the best result.

5.3 Experimental Results

For methods with DR, the results reported are the best results among varied dimensions.[3]

5.3.1 Privacy-Preserving Clustering

Table 3 summarizes the results for clustering, and the following are main observations.

- Compared to the non-private baseline (real data), RON-Gauss has almost identical performance with only 0.01 additional error (4% error) while preserving strong privacy ($\epsilon = 1.0$).
- Compared to Li et al. [78] and Jiang et al. [66], who use the identity query as opposed to the Gaussian generative model, RON-Gauss has over 2x better utility with the same privacy budget.
- Compared to Blum et al. [16] and Liu [80], who do not use DR, RON-Gauss has over 10x and 3x better utility with the same privacy budget.

For RON-Gauss, the optimal number of clusters based on the Silhouette analysis is four. It is interesting to note that RON-Gauss achieves good results despite using the unimodal Gaussian model. This can partially be explained by the curse of dimensionality [9, 34, 48, 70].

3 As discussed by Chaudhuri et al. [26], in the real-world deployment, the parameter tuning process must be private as well.

One consequence of the curse of dimensionality is the concentration of the data mass near the surface of the hypercube encapsulating the data domain space. With respect to our results, this leads to the observation that despite using the unimodal Gaussian model the data generated by RON-Gauss can form different clusters around different parts of the hypercube surface. Thus, if this unimodal distribution can represent the original data well according to the DFM effect, it can provide clustering performance close to that of the original data.

5.3.2 Privacy-Preserving Classification

Table 4 summarizes the classification results. The following are main observations.

- Compared to the non-private baseline (real data), RON-Gauss has almost identical performance with 2.45% additional error, while preserving strong privacy ($\epsilon = 1.0$).
- Compared to Li et al. [78] and Jiang et al. [66], who use the identity query as opposed to GMM, RON-Gauss has over 20% and 30% better utility, respectively, with the same privacy budget.
- Compared to Blum et al. [16] and Liu [80], who do not use DR, RON-Gauss has over 35% and 25% better utility, respectively, with the same privacy.

5.3.3 Privacy-Preserving Regression

Table 5 summarizes the results for regression. The following are main observations.

- Compared to the non-private baseline (real data), RON-Gauss actually performs statistically equally well, while preserving strong privacy ($\epsilon = 1.0$).
- Compared to Li et al. [78] and Jiang et al. [66], who use the identity query as opposed to the Gaussian generative model, RON-Gauss has over 3x better utility with the same privacy budget.

Method	Model	DR	ϵ	RMSE ($\times 10^{-2}$)
Real data	-	-	-	0.21
Li et al. [78]	Identity	Bern. Rand.	1.	0.68 ± 0.01
Jiang et al. [66]	Identity	PCA	1.	0.68 ± 0.00
Blum et al. [16]	Exp. Mech.	-	1.	0.62 ± 0.07
Liu [80]	Gaussian	-	1.	1.00 ± 0.12
RON-Gauss	Gaussian	RON	1.	0.21 ± 0.01

Table 5. Regression results (Twitter dataset). RMSE is an error metric, so lower values indicate better utility. (note: RMSE of random guess is $\sim 57.20 \times 10^{-2}$).

– Compared to Blum et al. [16] and Liu [80], who do not use DR, RON-Gauss has over 3x and 5x better utility with the same privacy budget.

5.3.4 Summary of Experimental Results

RON-Gauss outperforms all four other methods in terms of utility across all three learning tasks. RON-Gauss also performs comparably well relative to the maximum utility achieved by the non-private baseline in all tasks. The main results are concluded as follows.

– RON-Gauss provides performance close to that attainable from the non-private real data.
– Using the Gaussian generative model over the identity query has been shown to provide the utility gain of up to 2x, 30%, and 3x for clustering, classification, and regression, respectively.
– Using RON to reduce dimension of the data has been shown to provide the utility gain of up to 10x, 35%, and 5x, for clustering, classification, and regression, respectively.

6 Discussion

6.1 Effect of Dimension on the Utility

In Section 4, we discuss how RON projection can reduce the level of noise required for DP. This effect can be observed experimentally as illustrated by Figure 1, which shows the relationship between the dimension the data are reduced to and the utility performance. Noticeably, there is a gain in utility as the dimension is reduced. Specifically, the peak performance is achieved at 4 dimensions in this case. This general trend is consistent across different privacy budgets. Seeking the optimal dimension a priori is an interesting topic for future research on the RON-projection-based methods.

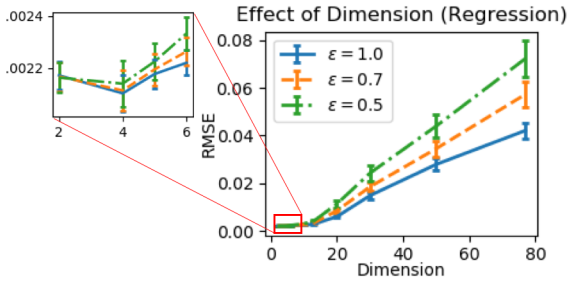

Fig. 1. Effects of dimension on the regression performance on Twitter dataset.

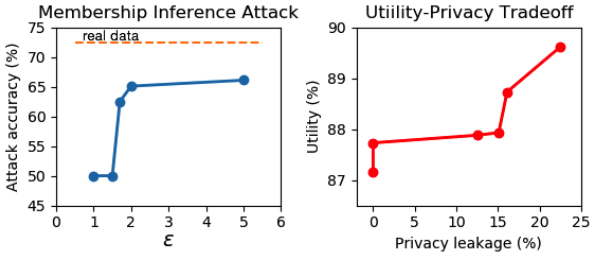

Fig. 2. Membership inference attack on RON-Gauss using Realdisp dataset. (Left) The attack accuracy against different values of ϵ. The dashed line shows the attack accuracy on the real data for comparison. (Right) The tradeoff between the utility (classification accuracy) and the privacy leakage (the difference between the membership inference accuracy and random guess at 50%).

6.2 RON-Gauss Against Membership Inference Attack

Recent works have suggested using inference attacks to measure the susceptibility of the released data and identify the appropriate values of ϵ for non-interactive differential privacy, e.g. [5, 114]. To evaluate RON-Gauss against inference attacks, we implement the membership inference attack proposed by Shokri et al. [114] using their published software [113]. This attack trains shadow machine learning models and an attack model to identify whether a given sample is in the dataset. Since their attack is designed for a classification task, we evaluate it on our classification experimental setup using Realdisp data. For the attack, we use ten shadow models and use neural network for the attack model with 0.01 learning rate trained on 50 epochs. These are the default parameter values of the software used [113].

The results are shown in Figure 2. The test set is chosen such that a random guess on the membership inference attack would yield an accuracy of 50%. Figure

2 (Left) suggests that ϵ values of 1.5 or less are appropriate for this setting since the performance of the membership inference attack is close to a random guess. Figure 2 (Right) illustrates the utility-privacy tradeoff based on this attack. The privacy leakage is defined as the attack accuracy above the random guess level. In other words, it measures how much the attack performs better than a random guess. The utility is measured by the classification accuracy similar to our classification experiments in Section 5. The plot allows the practitioners to choose an ϵ value that meets their utility-privacy tradeoff. For example, if we require the privacy leakage to be less than 10%, the curve in Figure 2 (Right) suggests that we can achieve almost 88% utility.

We note that the membership inference attack of Shokri et al. [114] may not be the optimal inference attack against RON-Gauss, since it is a general attack method not specifically tailored for our approach. We leave the analysis of more advanced attacks that specifically utilize knowledge of the RON-Gauss mechanism to future work, e.g. using hypothesis testing [5].

6.3 RON-Gauss as a Generative Model

Our work uses a parametric generative model to capture the essence of the unknown data distribution. Since RON-Gauss involves DR as an important step, the RON-Gauss model is inevitably lossy, i.e. there is information loss due to the use of the model itself. However, this loss is mitigated partly by the DFM effect, which ensures that the data are close to Gaussian after the RON projection. To illustrate the effectiveness of this effect and of RON-Gauss as a parametric generative model, we test RON-Gauss purely for its quality as a generative model, i.e. without the DP component, on the MNIST dataset [72, 73]. Since MNIST is typically used for classification, we use Algorithm 5 for RON-Gauss and set $\epsilon \to \infty$ to leave out the effect of DP noise. We project the data onto 392 dimensions – half of the original dimensions of 784 – and synthesize the samples, which are then reconstructed into the synthesized images. Examples of the synthesized images are shown in Figure 3. These images show good digit visibility, which indicates the potential of RON-Gauss as a generative model.

However, admittedly, the visibility of the digits subsides gradually as we project the data onto lower dimensions. Particularly, we observe that, at dimensions lower than 100, the digits are not very visible anymore. This depicts that, despite its promise, RON-Gauss may not yet be the universal model for every situation since there remains the need to balance the information loss due to DR. However, if sufficient information is retained, RON-Gauss has shown the potential to be a quality model by utilizing the DFM effect,as demonstrated by our experiments in Section 5.

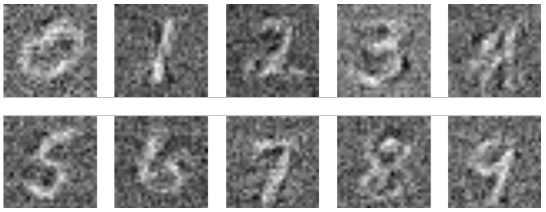

Fig. 3. Synthesized MNIST images from the RON-Gauss model without the DP component using half of the full dimensions.

6.4 The Design of ϵ_μ and ϵ_Σ for RON-Gauss Algorithms

RON-Gauss Algorithms take as inputs two privacy parameters: ϵ_μ and ϵ_Σ. The algorithms are then shown to preserve $(\epsilon_\mu + \epsilon_\Sigma)$-differential privacy. This means that, for a fixed total privacy budget of $\epsilon = \epsilon_\mu + \epsilon_\Sigma$, we can choose how much to allocate to ϵ_μ and ϵ_Σ in order to maximize the utility of the synthesized data. In our experiments, we fix the ratio between the two based on the observation about the sensitivity of the mean and the covariance. However, the allocation can possibly be designed better by formulating it as an optimization problem that aims at maximizing the utility of the synthetic data. Then, the optimal solution can be obtained using grid search, random search, or Bayesian optimization [115]. We leave this as a possible future direction.

7 Conclusion

In this work, we combine two previously non-intersecting techniques – random orthonormal projection and Gaussian generative model – to provide a solution to non-interactive private data release. We propose the RON-Gauss model that exploits the Diaconis-Freedman-Meckes effect, and present three algorithms for both unsupervised and supervised learning. We prove that our RON-Gauss model preserves ϵ-differential privacy. Finally, our experiments on three real-world datasets under clustering, classification, and regression applications show the strength of the method. RON-Gauss provides significant performance improvement over previous approaches, and yields the utility performance close to the non-private baseline, while preserving differential privacy with $\epsilon = 1$.

Acknowledgement

The authors would like to thank Sébastien Gambs for shepherding the paper, the anonymous reviewers for their valuable feedback, and Mert Al, Daniel Cullina, and Alex Dytso for insightful discussions. This work is supported in part by the National Science Foundation (NSF) under the grant CNS-1553437 and CCF-1617286, an Army Research Office YIP Award, and faculty research awards from Google, Cisco, Intel, and IBM.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318. ACM, 2016.

[2] Gergely Acs, Claude Castelluccia, and Rui Chen. Differentially private histogram publishing through lossy compression. In *ICDM*, pages 1–10. IEEE, 2012.

[3] Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *arXiv preprint arXiv:1709.04514*, 2017.

[4] Michael Backes, Pascal Berrang, Anne Hecksteden, Mathias Humbert, Andreas Keller, and Tim Meyer. Privacy in epigenetics: Temporal linkability of microrna expression profiles. In *Proceedings of the 25th USENIX Security Symposium*, 2016.

[5] Raghavendran Balu, Teddy Furon, and Sébastien Gambs. Challenging differential privacy: the case of non-interactive mechanisms. In *European Symposium on Research in Computer Security*, pages 146–164. Springer, 2014.

[6] Oresti Banos, Miguel Damas, Hector Pomares, Ignacio Rojas, Mate Attila Toth, and Oliver Amft. A benchmark dataset to evaluate sensor displacement in activity recognition. In *UBICOMP*, pages 1026–1035. ACM, 2012.

[7] Oresti Banos, Claudia Villalonga, Rafael Garcia, Alejandro Saez, Miguel Damas, Juan A. Holgado-Terriza, Sungyong Lee, Hector Pomares, and Ignacio Rojas. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online*, 14(2):1, 2015.

[8] Michael Barbaro and Tom Zeller Jr. A face is exposed for aol searcher no. 4417749. http://www.nytimes.com/2006/08/09/technology/09aol.html, Aug 9, 2006 2006.

[9] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[10] Raffael Bild, Klaus A. Kuhn, and Fabian Prasser. Safepub: A truthful data anonymization algorithm with strong privacy guarantees. *Proceedings on Privacy Enhancing Technologies*, 1:67–87, 2018.

[11] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. Plausible deniability for privacy-preserving data synthesis. *PVLDB*, 10(5), 2017.

[12] Christopher M. Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[13] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *FOCS*, pages 410–419. IEEE, 2012.

[14] Jeremiah Blocki, Anupam Datta, and Joseph Bonneau. Differentially private password frequency lists. In *NDSS*, 2016.

[15] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005.

[16] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *JACM*, 60(2):12, 2013.

[17] Claire McKay Bowen and Fang Liu. Differentially private data synthesis methods. *arXiv preprint arXiv:1602.01063*, 2016.

[18] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[19] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[20] Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive high-dimensional data visualization. *Journal of computational and graphical statistics*, 5(1):78–99, 1996.

[21] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 1–10. ACM, 2014.

[22] John Burkardt. Normal_dataset: Generate multivariate normal random datasets. https://people.sc.fsu.edu/~jburkardt/cpp_src/normal_dataset/normal_dataset.html, 12/9/2009 2009.

[23] Hyeran Byun and Seong-Whan Lee. Applications of support vector machines for pattern recognition: A survey. In *Pattern recognition with support vector machines*, pages 213–236. Springer, 2002.

[24] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like:" privacy risks of collaborative filtering. In *S&P*, pages 231–246. IEEE, 2011.

[25] Augustin-Louis Cauchy. Sur les formules qui resultent de l'emploie du signe et sur> ou<, et sur les moyennes entre plusieurs quantites. *Cours d'Analyse, 1er Partie: Analyse algebrique*, pages 373–377, 1821.

[26] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *JMLR*, 12(Mar):1069–1109, 2011.

[27] Kamalika Chaudhuri, Anand Sarwate, and Kaushik Sinha. Near-optimal differentially private principal components. In *NIPS*, pages 989–997, 2012.

[28] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial

decompositions. In *ICDE*, pages 20–31. IEEE, 2012.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[30] Paul Cuff and Lanqing Yu. Differential privacy as a mutual information constraint. In *CCS*, pages 43–54. ACM, 2016.

[31] Wei-Yen Day and Ninghui Li. Differentially private publishing of high-dimensional data using sensitivity control. In *CCS*, pages 451–462. ACM, 2015.

[32] Fernando de Almeida Freitas, Sarajane Marques Peres, Clodoaldo Aparecido de Moraes Lima, and Felipe Venancio Barbosa. Grammatical facial expressions recognition with machine learning. In *FLAIRS Conference*, 2014.

[33] Persi Diaconis and David Freedman. Asymptotics of graphical projection pursuit. *The annals of statistics*, pages 793–815, 1984.

[34] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[35] Cynthia Dwork. *Differential privacy*, pages 1–12. Automata, languages and programming. Springer, 2006.

[36] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[37] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[38] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380. ACM, 2009.

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

[40] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390. ACM, 2009.

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

[42] Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2):2, 2010.

[43] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *STOC*, pages 11–20. ACM, 2014.

[44] Ulfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS*, pages 1054–1067. ACM, 2014.

[45] Giulia Fanti, Vasyl Pihur, and Ulfar Erlingsson. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3):41–61, 2016.

[46] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[47] Ronald Aylmer Fisher. Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 22, pages 700–725. Cambridge University Press, 1925.

[48] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[49] Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheip, Bjoern Bornkamp, Martin Maechler, and Torsten Hothorn. Package mvtnorm, 02/02/2016 2016.

[50] Jorgen Pedersen Gram. Uber die entwicklung reeller funktionen in reihen mittels der methode der kleinsten quadrate. *Journal fur reihe und angewandte Mathematik*, 94:41–73, 1883.

[51] Saikat Guha, Mudit Jain, and Venkata N. Padmanabhan. Koi: A location-privacy platform for smartphone apps. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 14–14. USENIX Association, 2012.

[52] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. *Theory of Cryptography*, pages 339–356, 2012.

[53] Andreas Haeberlen, Benjamin C. Pierce, and Arjun Narayan. Differential privacy under fire. In *USENIX Security Symposium*, 2011.

[54] Marjorie G. Hahn and Michael J. Klass. The multidimensional central limit theorem for arrays normed by affine transformations. *The Annals of Probability*, pages 611–623, 1981.

[55] Peter Hall and Ker-Chau Li. On almost linearity of low dimensional projections from high dimensional data. *The annals of Statistics*, pages 867–889, 1993.

[56] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.

[57] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *NIPS*, pages 2339–2347, 2012.

[58] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70. IEEE, 2010.

[59] Moritz Hardt, Guy N. Rothblum, and Rocco A. Servedio. Private data release via learning thresholds. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 168–187. Society for Industrial and Applied Mathematics, 2012.

[60] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. Principled evaluation of differentially private algorithms using dpbench. In *ICMD*, pages 139–154. ACM, 2016.

[61] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3(1-2):1021–1032, 2010.

[62] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[63] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[64] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[65] William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379, 1961.

[66] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado. Differential-private data publishing through component analysis. *Transactions on data privacy*, 6(1):19–34, Apr 2013.

[67] Francois Kawala, Ahlame Douzal-Chouakria, Eric Gaussier, and Eustache Dimert. Predictions d'activite dans les reseaux sociaux en ligne. In *4ieme Conference sur les Modeles et l'Analyse des Reseaux: Approches Mathamatiques et Informatiques*, page 16, 2013.

[68] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):2, 2013.

[69] Ross D. King, Cao Feng, and Alistair Sutherland. Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, 9(3):289–333, 1995.

[70] Mario Köppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, volume 1, pages 4–8, 2000.

[71] S. Y. Kung. *Kernel Methods and Machine Learning*. Cambridge University Press, Cambridge, UK, 2014.

[72] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[73] Yann Lecun, Corinna Cortes, and Christopher J.C Burges. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 1998.

[74] David Leoni. Non-interactive differential privacy: a survey. In *Proceedings of the First International Workshop on Open Data*, pages 40–52. ACM, 2012.

[75] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. A data-and workload-aware algorithm for range queries under differential privacy. *Proceedings of the VLDB Endowment*, 7(5):341–352, 2014.

[76] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB Journal*, 24(6):757–781, 2015.

[77] H. Li, L. Xiong, and X. Jiang. Differentially private synthesization of multi-dimensional data using copula functions. *Advances in database technology : proceedings.International Conference on Extending Database Technology*, 2014:475–486, 2014.

[78] Yang D. Li, Zhenjie Zhang, Marianne Winslett, and Yin Yang. Compressive mechanism: Utilizing sparse representation in differential privacy. In *WPES*, pages 177–182. ACM, 2011.

[79] Changchang Liu and Prateek Mittal. Linkmirage: Enabling privacy-preserving analytics on social relationships. In *23nd Annual Network and Distributed System Security Symposium, NDSS*, pages 21–24, 2016.

[80] Fang Liu. Model-based differential private data synthesis. *arXiv preprint arXiv:1606.08052*, 2016.

[81] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 277–286. IEEE, 2008.

[82] David McClure and Jerome P. Reiter. Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data. *Trans.Data Privacy*, 5(3):535–552, 2012.

[83] Frank McSherry and Ilya Mironov. Differentially private recommender systems: building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636. ACM, 2009.

[84] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103. IEEE, 2007.

[85] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[86] Elizabeth Meckes. Approximation of projections of random vectors. *Journal of Theoretical Probability*, 25(2):333–352, 2012.

[87] Elizabeth Meckes. *Projections of probability distributions: A measure-theoretic Dvoretzky theorem*, pages 317–326. Geometric Aspects of Functional Analysis. Springer, 2012.

[88] Carl D. Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[89] Noman Mohammed, Rui Chen, Benjamin Fung, and Philip S. Yu. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–501. ACM, 2011.

[90] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, One Rogers Street Cambridge MA 02142-1209, 2012.

[91] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *S&P*, pages 300–314. IEEE, 2012.

[92] David CL Ngo, Andrew BJ Teoh, and Alwyn Goh. Biometric hash: high-confidence face recognition. *IEEE transactions on circuits and systems for video technology*, 16(6):771–775, 2006.

[93] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007.

[94] II Ororbia, G. Alexander, Fridolin Linder, and Joshua Snoke. Privacy protection for natural language records: Neural generative models for releasing synthetic twitter data. *arXiv preprint arXiv:1606.01151*, 2016.

[95] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *JMLR*, 12(Oct):2825–2830, 2011.

[96] Nicholas G. Polson and Steven L. Scott. Data augmentation for support vector machines. *Bayesian Analysis*,

6(1):1–23, 2011.

[97] Netflix Prize. http://www.netflixprize.com/.

[98] Davide Proserpio, Sharon Goldberg, and Frank McSherry. Calibrating data to sensitivity in private data analysis: a platform for differentially-private analysis of weighted datasets. *PVLDB*, 7(8):637–648, 2014.

[99] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Differentially private grids for geospatial data. In *ICDE*, pages 757–768. IEEE, 2013.

[100] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Understanding hierarchical methods for differentially private histograms. *Proceedings of the VLDB Endowment*, 6(14):1954–1965, 2013.

[101] Raul Rojas. Why the normal distribution. *Freis Universitat Berlin lecture notes*, 2010.

[102] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420, 2007.

[103] Gunter Rote. A new metric between polygons. In Werner Kuich, editor, *ICALP: International Colloquium on Automata, Languages, and Programming*, pages 404–415. Springer, Berlin, Heidelberg, July 1992.

[104] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[105] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

[106] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98. ACM, 2011.

[107] Erhard Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.

[108] Bernhard Scholkopf and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[109] Hermann Amandus Schwarz. *Uber ein die Flachen kleinsten Flacheninhalts betreffendes Problem der Variationsrechnung*, pages 223–269. Gesammelte Mathematische Abhandlungen. Springer, 1890.

[110] Scikit-learn. Selecting the number of clusters with silhouette analysis on kmeans clustering. http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py.

[111] SciPy.org. scipy.stats.multivariate_normal. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.multivariate_normal.html, 5/11/2014 2014.

[112] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *CCS*, pages 1310–1321. ACM, 2015.

[113] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. https://github.com/csong27/membership-inference.

[114] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017.

[115] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[116] Open SNP. http://opensnp.org/.

[117] Florian Tramer, Zhicong Huang, Jean-Pierre Hubaux, and Erman Ayday. Differential privacy with bounded priors: reconciling utility and privacy in genome-wide association studies. In *CCS*, pages 1286–1297. ACM, 2015.

[118] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pages 400–416. Springer, 2011.

[119] Jalaj Upadhyay. Circulant matrices and differential privacy. *analysis*, 16:47, 2014.

[120] Jalaj Upadhyay. Randomness efficient fast-johnson-lindenstrauss transform with applications in differential privacy and compressed sensing. *arXiv preprint arXiv:1410.2470*, 2014.

[121] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[122] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[123] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[124] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

[125] Eric W. Weisstein. Sphere. http://mathworld.wolfram.com/Sphere.html.

[126] Wikipedia. Categorical variable. https://en.wikipedia.org/wiki/Categorical_variable#cite_ref-yates_1-0, 2017.

[127] Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. In *NIPS*, pages 2451–2459, 2010.

[128] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, 2011.

[129] Yonghui Xiao and Li Xiong. Protecting locations with differential privacy under temporal correlations. In *CCS*, pages 1298–1309. ACM, 2015.

[130] Yonghui Xiao, Li Xiong, Liyue Fan, and Slawomir Goryczka. Dpcube: differentially private histogram release through multidimensional partitioning. *arXiv preprint arXiv:1202.5358*, 2012.

[131] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[132] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 2017.

[133] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22(6):797–822, 2013.

[134] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1423–1434. ACM, 2014.

[135] Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

[136] Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, and Yingtao Xie. Towards accurate histogram publication under differential privacy. In *SDM*, pages 587–595. SIAM, 2014.

[137] Shuheng Zhou, Katrina Ligett, and Larry Wasserman. Differential privacy with compression. In *ISIT*, pages 2718–2722. IEEE, 2009.

A Proof of Lemma 3

Proof. The proof uses the property of orthogonal projection in a vector space. First, notice that $\left\|\mathbf{W}^T\mathbf{x}\right\|_F = \left\|\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_F$, which can be verified as follows.

$$\begin{aligned}\left\|\mathbf{W}^T\mathbf{x}\right\|_F &= \sqrt{\mathrm{tr}(\mathbf{x}^T\mathbf{W}\mathbf{W}^T\mathbf{x})}\\ &= \sqrt{\mathrm{tr}(\mathbf{x}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T\mathbf{x})}\\ &= \left\|\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_F,\end{aligned}$$

where the second equality is from the fact that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$. Then, notice that $\mathbf{P} = \mathbf{W}\mathbf{W}^T$ is a projection matrix with p orthonormal basis as the columns of $\mathbf{W}$ (cf. [88, Chapter 5]). Therefore, the idempotent property of $\mathbf{P}$ can be used as,

$$\begin{aligned}\left\|\mathbf{W}^T\mathbf{x}\right\|_F &= \left\|\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_F = \left\|\mathbf{P}\mathbf{x}\right\|_F\\ &= \langle\mathbf{P}\mathbf{x},\mathbf{P}\mathbf{x}\rangle_F = \langle\mathbf{P}\mathbf{x},\mathbf{x}\rangle_F.\end{aligned}$$

The last equality can be verified as follows. Let $\mathbf{P}\mathbf{x} \in \mathcal{P}$, and let $\mathbf{x}^\perp = \mathbf{x} - \mathbf{P}\mathbf{x} \in \mathcal{P}^\perp$, then $\langle\mathbf{P}\mathbf{x},\mathbf{x}\rangle_F = \left\langle\mathbf{P}\mathbf{x},\mathbf{P}\mathbf{x}+\mathbf{x}^\perp\right\rangle_F = \langle\mathbf{P}\mathbf{x},\mathbf{P}\mathbf{x}\rangle_F + \left\langle\mathbf{P}\mathbf{x},\mathbf{x}^\perp\right\rangle_F = \langle\mathbf{P}\mathbf{x},\mathbf{P}\mathbf{x}\rangle_F$ from the additivity of the inner product and the fact that $\left\langle\mathbf{P}\mathbf{x},\mathbf{x}^\perp\right\rangle_F = 0$ by construction. Then, using the Cauchy-Schwarz inequality, $\|\mathbf{P}\mathbf{x}\|_F^2 = \langle\mathbf{P}\mathbf{x},\mathbf{x}\rangle_F^2 \le \|\mathbf{P}\mathbf{x}\|_F\|\mathbf{x}\|_F$, and, hence, $\|\mathbf{P}\mathbf{x}\|_F = \left\|\mathbf{W}\mathbf{W}^T\mathbf{x}\right\|_F = \left\|\mathbf{W}^T\mathbf{x}\right\|_F \le \|\mathbf{x}\|_F$. □

B L_1-Sensitivity of the MLE of the Covariance

Consider the maximum likelihood estimate (MLE) [47] for the covariance matrix, which is an unbiased estimate (cf. [123]):

$$\boldsymbol{\Sigma}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(\widetilde{\mathbf{x}}_i - \boldsymbol{\mu})(\widetilde{\mathbf{x}}_i - \boldsymbol{\mu})^T,$$

where $\boldsymbol{\mu}$ is the sample mean specific to the instance of the dataset. Hence, the neighboring datasets may have different means. Then, the sensitivity can be derived as follows.

Lemma 6. *The L_1-sensitivity of the MLE of the covariance matrix is $(2\sqrt{p}+2n\sqrt{p})/n$.*

Proof. For neighboring datasets $\mathbf{X}, \mathbf{X}'$,

$$\begin{aligned}s(f) =& \sup\frac{1}{n}\left\|\sum_{i=1}^{n}(\widetilde{\mathbf{x}}_i-\boldsymbol{\mu})(\widetilde{\mathbf{x}}_i-\boldsymbol{\mu})^T\right.\\ &\left.-\sum_{i=1}^{n}(\widetilde{\mathbf{x}}'_i-\boldsymbol{\mu}')(\widetilde{\mathbf{x}}'_i-\boldsymbol{\mu}')^T\right\|_1\\ =& \sup\frac{1}{n}\left\|(\sum_{i=1}^{n}\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T - n\boldsymbol{\mu}\boldsymbol{\mu}^T)\right.\\ &\left.-(\sum_{i=1}^{n}\widetilde{\mathbf{x}}'_i\widetilde{\mathbf{x}}_i'^T - n\boldsymbol{\mu}'\boldsymbol{\mu}'^T)\right\|_1\\ =& \sup\frac{1}{n}\left\|(\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T - \widetilde{\mathbf{x}}'_i\widetilde{\mathbf{x}}_i'^T) + n(\boldsymbol{\mu}'\boldsymbol{\mu}'^T - \boldsymbol{\mu}\boldsymbol{\mu}^T)\right\|_1\\ \le& \sup\frac{1}{n}(\left\|\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T\right\|_1 + \left\|\widetilde{\mathbf{x}}'_i\widetilde{\mathbf{x}}_i'^T\right\|_1)\\ &+\left\|\boldsymbol{\mu}'\boldsymbol{\mu}'^T\right\|_1 + \left\|\boldsymbol{\mu}\boldsymbol{\mu}^T\right\|_1\\ \le& \sup\frac{2\sqrt{p}}{n}\left\|\widetilde{\mathbf{x}}_i\widetilde{\mathbf{x}}_i^T\right\|_F + 2\sqrt{p}\left\|\boldsymbol{\mu}'\boldsymbol{\mu}'^T\right\|_F\\ \le& \frac{2\sqrt{p}}{n} + 2\sqrt{p}.\end{aligned}$$

The last inequality is due to the following observation: $\|\boldsymbol{\mu}\|_F = \frac{1}{n}\|\sum\widetilde{\mathbf{x}}_i\|_F \le \frac{1}{n}\sum\|\widetilde{\mathbf{x}}_i\|_F = 1$. □