Compressive Learning with Privacy Guarantees

Antoine Chatalic, Vincent Schellekens, Florimond Houssiau, Yves-Alexandre de Montjoye, Laurent Jacques, Rémi Gribonval

Compressive Learning with Privacy Guarantees

A. Chatalic[(1)], V. Schellekens[(2)*], F. Houssiau[(3)],
Y.-A. de Montjoye[(3)], L. Jacques[(2)*] and R. Gribonval[(1,4)]
[(1)] Univ. Rennes, Inria, CNRS, IRISA [(2)] ICTEAM/ELEN, UCLouvain
[(3)] Imperial College London [(4)] Univ Lyon, Inria, CNRS, ENS de Lyon, UCB Lyon 1, LIP

January 20, 2021

Abstract This work addresses the problem of learning from large collections of data with privacy guarantees. The compressive learning framework proposes to deal with the large scale of datasets by compressing them into a single vector of generalized random moments, called a sketch vector, from which the learning task is then performed. We provide sharp bounds on the so-called sensitivity of this sketching mechanism. This allows to leverage standard techniques to ensure differential privacy – a well established formalism for defining and quantifying the privacy of a random mechanism – by adding Laplace of Gaussian noise to the sketch. We combine these standard mechanisms with a new feature subsampling mechanism, which reduces the computational cost without damaging privacy. The overall framework is applied to the tasks of Gaussian modeling, k-means clustering and principal component analysis (PCA), for which sharp privacy bounds are derived. Empirically, the quality (for subsequent learning) of the compressed representation produced by our mechanism is strongly related with the induced noise level, for which we give analytical expressions.

1 Introduction

The size and availability of datasets has increased dramatically in the last few decades, leading to tremendous breakthroughs in machine learning and artificial intelligence. However, the large volume and level of detail of these data present two key challenges. Firstly, the sheer size of datasets calls for new machine learning methods able to process them efficiently, both in time and memory. Secondly, the data collected is often of a sensitive nature, and using it to learn publicly released models raises serious privacy concerns, creating a need for algorithms that guarantee the privacy of the dataset contributors.

Compressive learning [27] has been proposed as an answer to the first challenge. In the compressive learning framework, the dataset is compressed into a *sketch*, a vector of generalized random moments [15] – obtained by averaging over the dataset certain random (nonlinear) features of the records [44] – whose size is independent from the number of records. The learning step can then be performed from this sketch only, using greatly reduced computational resources (see Figure 1). Once the sketch is computed, the dataset can be discarded. As opposed to many machine learning algorithms, compressive learning does not need the data to be stored in one place nor to be accessed multiple times; computing the sketch can be done in one pass over the data or from a data stream, has a low memory footprint, and is embarrassingly parallelizable. As the size of the sketch does not depend on the size of the dataset, but rather on the amount of information we want to extract from its underlying distribution, learning from this vector has a computational cost which is *independent* of the initial dataset size.

As compressive learning requires only aggregate information from many individual records, it is intuitively a good candidate to answer the second challenge of privacy preservation. Differential privacy (DP) [19] was proposed by Dwork et al. as a formal privacy definition, that intuitively requires the output of an algorithm to not depend too much on the presence of any record in the dataset. It has many powerful properties, and has been shown to be robust to many attacks, which has made it widely

*V. Schellekens and L. Jacques are funded by the "Fonds de la Recherche Scientifique" (F.R.S. - FNRS). Part of this work was supported by the FNRS Grant T.0136.20 (PDR).

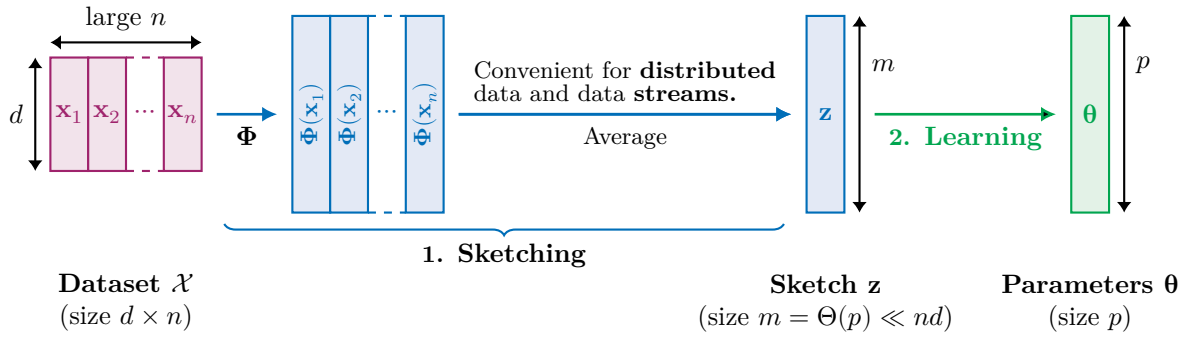

Figure 1: Overview of sketching and parameter learning.

accepted by the scientific community as a standard definition of privacy. It has further received a lot of attention in the industry [50, 23].

A standard approach to ensure differential privacy is to add noise (typically Laplace or Gaussian) to the output of the mechanism one wishes to make private. The privacy level of the resulting randomized mechanism is then known to be determined by the so-called sensitivity of the initial mechanism, which we assume to be deterministic in the following. This is the approach conducted in this paper[1], leading to a generic mechanism relying on noise addition that produces differentially private versions of the sketch, which is applied to the tasks of k-means clustering, Gaussian mixture modeling and principal component analysis (PCA). This differentially private sketch is a private representation of the dataset, which can be used — possibly multiple times and for different purposes — without infringing the privacy of any user in the dataset. By sharply characterizing the sensitivity of the sketching mechanism, we obtain sharp privacy guarantees for the resulting mechanism.

Subsampling the dataset is another common practice to enhance privacy [7]. Although it does not allow to grant privacy alone, it is known to amplify privacy of any existing differentially-private mechanism [4]. We introduce a simple *feature subsampling* mechanism, which differs from more standard data subsampling mechanisms, so that each data sample only contributes to some of the entries of the sketch, in order to reduce the computational cost of sketching. Privacy guarantees are also established for this mechanism through appropriately modified measures of sentitivity. Subsampling allows to reduce drastically the computational complexity, and can be performed in some settings without degrading the quality of the sketch for subsequent learning.

Finally, as privacy naturally has to be traded off for utility, we show empirically for the k-means clustering task that the utility of a noisy sketch is driven – provided the sketch dimension exceeds some task-dependent threshold – by a signal-to-noise ratio, which provides guidelines to parameter tuning of the algorithms. The obtained framework has thus a good balance between computational efficiency, privacy preservation and quality of the learned model.

Summary of contributions The contributions of this paper are as follows:

- We build on existing compressive learning and differential privacy techniques to define a **noisy sketching mechanism** which exploits nonlinear random features.
- We derive **sharp sensitivity estimates** for this mechanism, leading via standard tools to sharp differential privacy guarantees for sketches designed to handle three unsupervised learning tasks: k-means clustering, Gaussian mixture modeling and principal components analysis.
- We extend our framework to **subsampled sketches**, giving the same privacy guarantees for a lower computational cost.
- We show that the utility of a noisy sketch, i.e. its quality for subsequent learning, can be measured by a **signal-to-noise ratio**, and use this quantity for tuning some parameters.

[1] A first and reduced version of this work with privacy upper bounds and without the subsampling mechanism has been previously published [45].

Related Work We focus on the three learning tasks considered in this paper: Gaussian modeling (GMM), PCA and k-means clustering. The two latter have already received a lot of attention in the differential privacy literature, while the former has been less studied.

Addition of noise is the most common way to achieve differential privacy, whether it is on the intermediate steps of an iterative algorithm or directly on the output. Private variants of standard iterative methods include DPLloyd for k-means [8], and variants with improved convergence guarantees [38]. The popular k-means++ seeding method has also been generalized to a private framework [41]. For Gaussian modeling, DP-GMM [59] and DP-EM [42] have been proposed. Note that for iterative algorithms, the privacy budget needs to be split between iterations, de facto limiting the total number of iterates, which becomes a hyper-parameter. Our approach does not suffer from this drawback since the sketch is released at once. Moreover, the same sketch can be used to run the learning algorithm multiple times with e.g. different initializations.

Releasing a private synopsis of the data (similarly to our sketch) rather than directly a noisy solution has already been studied as well. EUGkM [43, 48] suggests for instance to use noisy histograms for clustering (but this method is by nature limited to small dimensions), and private coresets have been investigated by Feldman et al. [24, 25]. For PCA, noise can be added directly on the covariance matrix [22].

The exponential mechanism is another standard noise-additive approach for privacy. A random perturbation is drawn according to a distribution calibrated using a user-defined quality measure, and added to the output. It has been used with success for PCA, perturbing either the covariance [14, 30, 29] or directly the eigenvectors of the covariance [31, 1], and with genetic algorithms for k-means [60]. Such algorithms depend strongly on the quality measure of the output, which must be chosen carefully. Our sketch-based approach is in contrast more generic: the same sketch allows to solve different tasks such as clustering and GMM fitting, and it can easily be extended to new sketches in the future. Alternatively, our mechanism can be seen as a straightforward instantiation of the exponential mechanism, where the output (the sketch) is carefully designed so that is makes sense to simply use the L^1 or L^2 norms as quality measures.

Our sketching mechanism makes use of random projections, which have proven to be very useful to solve efficiently large-scale problems, and induce as well a controlled loss of information which can be leveraged to derive privacy guarantees [9]. Balcan et al. investigated the large-scale high-dimensional clustering setting with an approach based on Johnson-Lindenstrauss dimensionality reduction [3]. Many other embeddings based on random projections have been proposed, see e.g. [32]. Linear compression of the number of samples (rather than reducing the dimension) has been considered [61] but is less scalable. Random algorithms have also been used for PCA and, more generally, for low-rank factorization [28, 53, 2]. Note however that as explained in the next section, the features resulting from the random projection undergo in our setting a *nonlinear transformation*, in the spirit of random features [44], and are averaged; they thus differ a lot from what is done in these works, although they share this common idea.

Private k-means clustering algorithms based on the minimum enclosing ball problem have also been proposed [40, 47]. Yet, it is not clear how such methods compare in practice to the numerous other candidates.

Private empirical risk minimization [13, 56] has emerged as a generic way to design private learning algorithms, but it relies on specific assumptions (e.g. convexity, which does not hold for PCA, GMM modeling and kmeans) on the loss function which defines the learning task, and still relies on multiple passes over the whole dataset.

Closer to our work, Balog et al. [6] recently proposed to release kernel mean embeddings, either as sets of synthetic data points in the input space or using feature maps, similarly to our method. However, to the best of our knowledge, the impact of privacy on the quality of learning in such methods has not been studied in the literature.

Paper outline and reading guide The main existing tools and concepts from compressive learning and differential privacy are respectively recalled in sections 2 and 3. The reader knowledgeable with tools from either of these fields can probably safely skip the corresponding sections except to get familiar with the chosen notations. These tools are combined in Section 4 where we generically characterize the sensitivity of the sketching mechanism and provide new explicit expressions of this sensitivity for particular feature maps. Section 5 is devoted to describing the proposed feature subsampling mechanism

and characterizing its privacy level, and Section 6 gives evidence of the relevance of a noise-to-signal ratio as a proxy for utility, before exploring its use to provide guidelines to tune mechanisms.

2 Statistical Learning using Compressive Methods

Throughout the paper, d always refers to the dimension of the data samples. We denote $\mathcal{D}_n \triangleq E^n$ the set of (ordered) collections of n learning examples in a domain E, and $\mathcal{D} \triangleq \cup_{n\in\mathbb{N}}\mathcal{D}_n$. Unless otherwise specified, we will typically consider $E = \mathbb{R}^d$. The number of elements in a collection $\mathcal{X}$ is denoted $|\mathcal{X}|$. Note that we work with ordered datasets for technical reasons, but this order does not matter from a learning perspective.

Essentially, machine learning aims at inferring the parameters $\boldsymbol{\theta} \in \mathcal{H}$ of a mathematical model from a collection $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ of training samples in $\mathbb{R}^d$ drawn from a probability distribution π_0, i.e. $\mathbf{x}_i \overset{\text{iid}}{\sim} \pi_0$. In statistical learning, a task is defined by a loss function $l : (\mathbf{x}, \boldsymbol{\theta}) \mapsto l(\mathbf{x}, \boldsymbol{\theta}) \in \mathbb{R}$ which measures the relevance of the parameter $\boldsymbol{\theta}$ with respect to $\mathbf{x}$ for the task, and the associated risk function $\mathcal{R}(\pi, \boldsymbol{\theta}) = \mathbf{E}_{\mathbf{x}\sim\pi}\, l(\mathbf{x}, \boldsymbol{\theta})$ which extends this loss to distributions (and thus depends on the chosen loss, although we do not reflect this in the notation for conciseness). Intuitively, $\mathcal{R}(\pi, \boldsymbol{\theta})$ characterizes how $\boldsymbol{\theta}$ is suited to solve the learning task for distribution π. Learning thus amounts to finding $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathcal{H}} \mathcal{R}(\pi_0, \boldsymbol{\theta})$, but since the true distribution π_0 is unknown in practical applications, one typically uses the empirical distribution $\pi_{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}_i}$ associated to a dataset $\mathcal{X}$ and looks for $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\mathcal{H}} \mathcal{R}(\pi_{\mathcal{X}}, \boldsymbol{\theta})$; this is known as empirical risk minimization.

Standard approaches for empirical risk minimization access each data sample multiple times, and hence require them to be stored for the whole runtime of the algorithm. Compressive learning was introduced as a machine learning method that bypasses these needs by learning from a heavily compressed summary of the dataset, called the sketch, instead of the full dataset. In our context, this sketch is defined as the sample average of a feature map $\boldsymbol{\Phi}$, as depicted in Figure 1.

Definition 1 (Sketch)**.** *The sketch $\mathbf{z}_{\mathcal{X}}$ of a dataset $\mathcal{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{D}$ associated with the feature map $\boldsymbol{\Phi}$ is defined as*

$$\mathbf{z}_{\mathcal{X}} \triangleq \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\Phi}(\mathbf{x}_i), \quad \textit{where } \boldsymbol{\Phi} : \mathbb{R}^d \to \mathbb{C}^m \textit{ or } \mathbb{R}^m. \tag{1}$$

The choice of the feature map depends on the learning task to solve, however $\boldsymbol{\Phi}$ will typically be a nonlinear function, in order to capture more information from $\mathcal{X}$ than the first order moments. In this paper, we consider non-linear functions of the form $\boldsymbol{\Phi}(\mathbf{x}) = f(\Omega^T\mathbf{x})$, where $\Omega \in \mathbb{R}^{d\times m}$ is a randomly generated (but fixed) matrix and the nonlinear f is applied pointwise. Hence, the sketch is a collection of generalized random moments of the empirical distribution, i.e. $\mathbf{z}_{\mathcal{X}} = \mathbb{E}_{\mathbf{x}\sim\pi_{\mathcal{X}}}\boldsymbol{\Phi}(\mathbf{x})$. Note that although the feature map is a nonlinear function, sketching is a linear operation w.r.t. distributions. Estimating the desired model parameters from the sketch is done by solving a problem of the form $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}\in\mathcal{H}} \mathcal{S}(\mathbf{z}_{\mathcal{X}}, \boldsymbol{\theta})$, where $\mathcal{S}(\mathbf{z}_{\mathcal{X}}, \cdot)$ is a surrogate for $\mathcal{R}(\pi_{\mathcal{X}}, \cdot)$. This framework bears similarities with compressive sensing [26], which investigates the possibility of recovering signals from a small number of linear measurements, provided that these signals belong to (or can be well approximated by) a low-dimensional model. In this sense, learning from the sketch can be seen as an inverse problem on probability distributions: one will try to recover, in a model adapted to the learning task, a distribution whose sketch (i.e. linear observations) matches the moments computed on the dataset.

In sections 2.1 and 2.2, we formally define the three learning tasks we are interested in − namely clustering, density fitting and PCA −, and explain how they can be solved using a compressive approach.

2.1 Sketching with Fourier Features for Clustering and Density Fitting

We first define the tasks of unsupervised k-means clustering and Gaussian mixture modeling, as these can both be answered with the same feature map.

Definition 2 (k-means clustering task)**.** *Given an integer $k > 0$, k-means clustering consists in finding centroids $C = \{\mathbf{c}_1, \dots, \mathbf{c}_k\} \subseteq \mathbb{R}^d$ minimizing the empirical risk $\mathcal{R}_{\text{KM}}$ associated to the loss function $l_{\text{KM}}(\mathbf{x}, \mathbf{c}) \triangleq \min_{1\le j\le k} \|\mathbf{x} - \mathbf{c}_j\|_2^2$.*

In this specific case, the empirical risk (computed on the empirical distribution $\pi_{\mathcal{X}}$ of the dataset $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$) is also called the sum of squared errors (SSE):

$$\mathcal{R}_{\mathrm{KM}}(\pi_{\mathcal{X}}, C) = \mathrm{SSE}(\mathcal{X}, C) \triangleq \frac{1}{n} \sum_{i=1}^{n} \min_{1 \leq j \leq k} \|\mathbf{x}_i - \mathbf{c}_j\|_2^2. \tag{2}$$

For Gaussian modeling, the empirical risk is the log-likelihood of the mixture of Gaussians that the model fits to the data.

Definition 3 (Gaussian mixture modeling task)**.** *Given an integer $k > 0$, Gaussian mixture modeling consists in finding the parameters (weights $\alpha_1, ..., \alpha_k \in \mathbb{R}_+$ s.t. $\sum_{1 \leq i \leq k} \alpha_i = 1$, locations $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k \in \mathbb{R}^d$ and covariances $\Sigma_1, \dots, \Sigma_k \in \mathbb{R}^{d \times d}$) of a Gaussian mixture M whose p.d.f. $p_M(\mathbf{x}) \triangleq \sum_{i=1}^{k} \alpha_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \Sigma_i)$ maximizes the log-likelihood, i.e. minimizes the empirical risk $\mathcal{R}_{\mathrm{GMM}}$ associated to the loss function $l_{\mathrm{GMM}}(\mathbf{x}, M) \triangleq -\ln p_M(\mathbf{x})$.*

To solve clustering and density modeling tasks in a compressive manner, previous works focused on random Fourier features (RFF), which consist in using the complex exponential as the nonlinear function [44] in the feature map. This has been applied to clustering and fitting parametric mixture models, such as Gaussian mixture models [33] or alpha-stable distributions [35].

Formally, for a given matrix of frequencies $\Omega = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m]$, the random Fourier feature map is defined by $\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}) \triangleq \exp\left(\mathrm{i}\Omega^T \mathbf{x}\right) \in \mathbb{C}^m$ (i.e., we chose the complex exponential nonlinearity $f(\cdot) = \exp(\mathrm{i}\cdot)$). The frequency vectors are typically i.i.d. Gaussians, i.e. $\boldsymbol{\omega}_i \sim \mathcal{N}(0, \sigma^2 I_d)$, where the variance parameter σ^2 must be adapted to the data, e.g. using prior knowledge on the distribution. The choice of the frequency distribution indeed defines implicitly in the sample space a kernel function [27], which can be interpreted as an inner product whose scale must be coherent with the scale of the clusters to identify.

Once the empirical sketch $\mathbf{z}_{\mathcal{X}}$ of a dataset has been computed, k-means clustering (or GMM fitting) can be performed by solving a linear inverse problem: one wants to recover an "ideal" distribution, i.e. belonging to a meaningful mathematical model $\mathcal{P}$, whose moments are as close as possible to the moments $\mathbf{z}_{\mathcal{X}}$ measured on the dataset. An intuitive and sound choice for the model $\mathcal{P}$ is the set of mixtures of k diracs [36] for clustering, or the set of mixtures of k normal distributions [34] for GMM fitting. Writing the elements of $\mathcal{P}$ as mixtures of k simple parametric probability distributions, we then solve

$$\nu^* \in \arg\min_{\nu \in \mathcal{P}} \|\mathbf{E}_{\mathbf{x} \sim \nu} \boldsymbol{\Phi}(\mathbf{x}) - \mathbf{z}_{\mathcal{X}}\|_2, \tag{3}$$

which is a parametric optimization problem acting as a surrogate for risk minimization as explained at the beginning of the section. For clustering, the locations of the k diracs forming ν^* define the estimated centroids, while for GMMs ν^* itself is the estimated mixture density.

Quantized Sketches Note that the expression of random Fourier features can be rewritten $\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}) = \left(\cos(\Omega^T \mathbf{x}) + \mathrm{i} \cos(\Omega^T \mathbf{x} - \frac{\pi}{2})\right)$. It was shown that the cosine in this expression can be replaced by any other nonlinear, periodic function ρ, while preserving the properties of the sketch. Specifically, if a uniform, random dithering $\mathbf{u} \in [0, 2\pi[^m$ is added before the nonlinearity, i.e. $f(\cdot) = \rho(\cdot + \mathbf{u}) + \mathrm{i}\rho(\cdot + \mathbf{u} - \frac{\pi}{2})$ with $u_j \overset{\mathrm{iid}}{\sim} \mathcal{U}([0, 2\pi[)$, the moment-fitting cost function defined in (3) can easily be adapted so that recovery can still be performed. In particular, a good approximation to the complex exponential moment fitting can be obtained using a quantized sketching variant [46], i.e. quantizing the cosine of both real and imaginary parts with ± 1 (one-bit quantization). Although the average sketch will in the end belong to $\mathbb{C}^m$, the mechanism produces individual sketches $\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}_i)$ in $(\{-1, 1\} + \mathrm{i}\{-1, 1\})^m$, and thus has a reduced memory footprint ($2m$ bits); this is especially convenient if these individual sketches have to be computed on low-power and low-memory devices, or to be sent over a network before being averaged.

To account for all possibilities, we provide a unified definition of the random Fourier feature (RFF) map, covering both quantized and unquantized cases.

Definition 4 (Random Fourier features)**.** *For $\Omega = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m]$, the random Fourier feature map is defined by*

$$\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}) \triangleq \left[\rho(\Omega^T \mathbf{x} + \mathbf{u}) + \mathrm{i}\rho(\Omega^T \mathbf{x} + \mathbf{u} - \frac{\pi}{2})\right] \in \mathbb{C}^m, \tag{4}$$

with the particular cases

$$\begin{cases} \mathbf{u} = \mathbf{0} \text{ and } \rho = \cos \text{ for random Fourier features} \\ \mathbf{u} \in [0, 2\pi[^m \text{ and } \rho = 2^{-1/2}\,\mathrm{sign} \circ \cos \text{ for quantized features.} \end{cases}$$

Note that with the normalization used on ρ, we have for any j that $|\mathbf{\Phi}^{\mathrm{RFF}}(\mathbf{x})_j| = 1$ for both quantized and nonquantized features.

2.2 Sketching with Quadratic Features for Compressive PCA

Principal component analysis consists, for a given $k < d$, in finding a k-dimensional linear subspace that best fits the data. Such a subspace can be parametrized by a matrix $W \in \mathbb{R}^{d\times k}$.

Definition 5 (PCA task)**.** *Principal component analysis aims at finding $W \in \mathbb{R}^{d\times k}$ minimizing the empirical risk $\mathcal{R}_{\mathrm{PCA}}$ associated to the loss function $l_{\mathrm{PCA}}(\mathbf{x}, W) \triangleq \|\mathbf{x} - WW^T\mathbf{x}\|_2^2$.*

The matrix of second moments $C = \frac{1}{n}\sum_{1\leq i\leq n} \mathbf{x}_i\mathbf{x}_i^T$, which can be seen as a sketch computed using the feature map $\mathbf{\Phi}(\mathbf{x}) = \mathrm{vec}(\mathbf{x}\mathbf{x}^T)$, is known to capture all the information needed to solve the PCA problem. In practical applications, only the first k eigenvectors of C are needed, and so the sketch can be further reduced using low-rank matrix recovery techniques. The following feature map is then used [27].

Definition 6 (Random quadratic features)**.** *Let $\Omega = [\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$. Choosing the nonlinearity $f(\cdot) = (\cdot)^2$, the feature function used for PCA is*

$$\mathbf{\Phi}^{\mathrm{RQF}}(\mathbf{x}) \triangleq [(\boldsymbol{\omega}_1^T\mathbf{x})^2, \dots, (\boldsymbol{\omega}_m^T\mathbf{x})^2]^T.$$

We will typically consider two different sampling schemes for Ω:

- **Gaussian:** the $(\boldsymbol{\omega}_i)_{1\leq i\leq m}$ are drawn as $\boldsymbol{\omega}_i \sim \mathcal{N}(0, d^{-1}I_d)$. Note that with this variance, we have $\mathbf{E}_{\boldsymbol{\omega}\sim\mathcal{N}(0,d^{-1}I_d)}\|\boldsymbol{\omega}\|_2^2 = 1$ for coherence with the next sampling scheme.
- **Union of orthonormal bases:** when m/d is an integer, we consider $\Omega = [B_1, ..., B_{m/d}]$ where the $(B_i)_{1\leq i\leq m/d}$ are $d\times d$ blocs whose columns form orthonormal bases of $\mathbb{R}^d$. This setup is useful for two reasons. First it makes it possible to use structured blocs B_i for which the matrix-vector product can be computed efficiently using fast transforms, but it also yields sharp privacy guarantees, as will be discussed in Section 4.1.2.

Note that after averaging this feature map over all data samples, we are left with rank-one measurements of the data covariance matrix, i.e. $\mathbf{z}_{\mathcal{X}} = \left[\boldsymbol{\omega}_i^T C \boldsymbol{\omega}_i\right]_{1\leq i\leq m}^T = \left[\langle \boldsymbol{\omega}_i\boldsymbol{\omega}_i^T, C\rangle\right]_{1\leq i\leq m}^T \triangleq \mathcal{M}(C)$, where $\mathcal{M}$ is a linear operator acting on matrices. Solving the PCA task is here again casted into a linear inverse problem. Indeed, one aims at recovering the first eigenvectors of C, which amounts to finding a low-rank approximation of C. It is well established in the literature that the feature function proposed above is suitable for this task [26, Section 4.6]. The problem thus boils down to finding a low-rank approximation from the sketch [27]:

$$\hat{C} \in \underset{\Sigma\geq 0, \mathrm{rank}(\Sigma)\leq k}{\arg\min} \|\mathcal{M}(\Sigma) - \mathbf{z}_{\mathcal{X}}\|. \tag{5}$$

This is a well studied problem which can be solved using e.g. nuclear norm relaxation [26]. As discussed later in the manuscript, a Burer-Monteiro factorization [11] can also be used, yielding an optimization problem which, despite being non convex, usually displays nice properties [55] and incurs a smaller memory cost than the convex nuclear norm formulation.

3 Differential Privacy

Publishing quantities computed from a collection of people's records – e.g. a machine learning model or aggregate statistics – can compromise the privacy of these users, even when these quantities result from aggregation over millions of data providers [17]. Differential Privacy (DP) was proposed as a strong privacy definition by Dwork et al. [19], and has since been studied and used extensively in research and industry [23, 50]. We here give a brief introduction to DP, and also detail the assumptions made on the attacker (the attack model), which have a direct impact on the kind of guarantees that can be achieved.

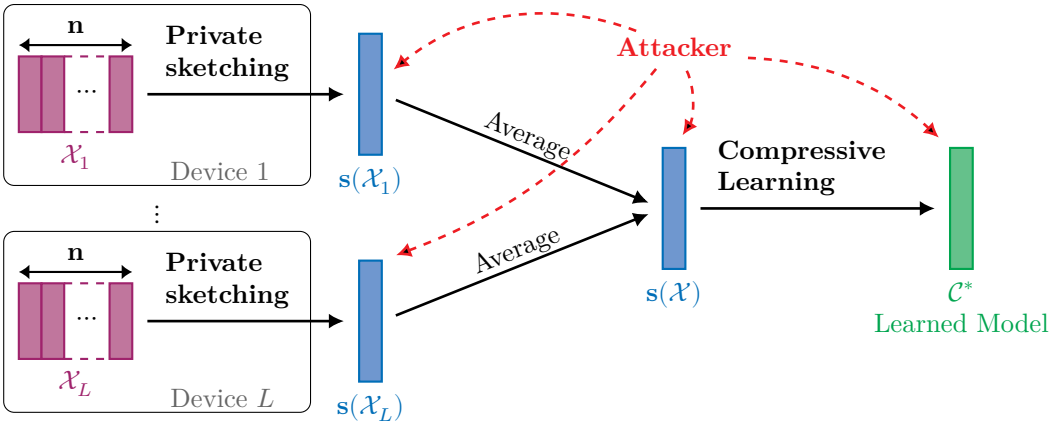

Figure 2: Attack model. The dataset is distributed between L devices, each computing and releasing publicly a subsampled sketch $\mathbf{s}(\mathcal{X}_i)$.

Attack Model We consider a *curator DP* model, where a trusted curator has access to the data, and publishes a noisy sketch of this data. The adversary is non-interactive, in that they have full access to the sketch of the dataset, or to sketches of disjoint subsets of the dataset if the latter is distributed across multiple devices (Figure 2), but cannot query the curator(s) for more data. Whereas there exist some approaches that use random projection matrices as encryption keys [51], we here assume that the feature map $\mathbf{\Phi}$ and the matrix of frequencies Ω are publicly known (similarly to, e.g., [32]). This is essential for analysts, who need to know the feature map in order to learn from the sketch. The model also covers the case where analysts may be adversaries. We assume that each user contributes exactly one record to the total dataset, albeit our results can be extended to allow for multiple records per user. We do not make any assumptions on the background knowledge available to the adversary, nor on the operations that they are able to make. Hence, our privacy guarantees are robust to extreme cases where the adversary knows the entire database save for one user, and has infinite compute power.

3.1 Definition and Properties

Randomness is an old tool for introducing uncertainty ("privacy by plausible deniability") when using sensitive information, e.g. implemented as randomized response surveys [58]. Differential privacy [19] provides a formal definition of the privacy guarantees offered by a *randomized* data release mechanism $R : \mathcal{D} \to \mathcal{Z}$. Intuitively, a mechanism R provides differential privacy if its output does not depend significantly on the presence of any one user in the database, hence hiding this presence from an adversary.

Definition 7 (Differential privacy [19])**.** *The randomized mechanism R achieves ε-differential privacy (noted ε-DP) iff for any measurable set S of the co-domain of R, and any $\mathcal{X}, \mathcal{Y} \in \mathcal{D}$ s.t. $\mathcal{X} \sim \mathcal{Y}$ for some neighboring relation $\sim$ (see below):*

$$\mathbb{P}\left[R(\mathcal{X}) \in S\right] \leq \exp(\varepsilon)\, \mathbb{P}\left[R(\mathcal{Y}) \in S\right] \tag{6}$$

The parameter $\varepsilon > 0$ is called the privacy budget*.*

The smaller ε, the closer the output distributions for two neighboring datasets are, and the stronger the privacy guarantee. Equivalently, differential privacy can be defined through the notion of *privacy loss* of a randomized mechanism. This is particularly useful when proving that a mechanism is differentially private.

Definition 8 (Privacy loss [21])**.** *Let R be a randomized algorithm taking values in $\mathcal{Z}$. If R admits a density $p_{R(\mathcal{X})}$ over $\mathcal{Z}$ for each input $\mathcal{X}$, the privacy loss* function *is defined by*

$$L_R(\mathbf{s}, \mathcal{X}, \mathcal{Y}) \triangleq \log\left(\frac{p_{R(\mathcal{X})}(\mathbf{s})}{p_{R(\mathcal{Y})}(\mathbf{s})}\right).$$

The random mechanism R achieves ε-differential privacy iff $\sup_{\mathbf{s}\in\mathcal{Z}} \sup_{\mathcal{X},\mathcal{Y}\in\mathcal{D}:\,\mathcal{X}\sim\mathcal{Y}} L_R(\mathbf{s},\mathcal{X},\mathcal{Y}) \leq \varepsilon.$

Intuitively, small values of the privacy loss of R for some pair $\mathcal{X},\mathcal{Y}$ characterize regions of the co-domain where output random variables $R(\mathcal{X})$ and $R(\mathcal{Y})$ have "close" distributions.

Neighboring relation The neighboring relation $\sim$ in definition 7 defines the practical guarantees that DP offers. A common definition, called "unbounded" differential privacy (UDP), states that two datasets are neighbors if they differ by the addition or deletion of exactly one sample. From definition 7, this implies that the output of an algorithm that satisfies unbounded DP does not significantly depend on the presence of any one user in the dataset. An alternative is bounded DP (BDP), which defines two datasets as neighbors if and only if they differ by exactly one record by replacement.

We denote $[\![1,n]\!] = \{1,...,n\}$, $\mathcal{S}_n$ the permutation group of $\{1,\dots,n\}$ and $\sigma(\mathcal{X})$ a permuted collection: $\sigma((\mathbf{x}_1,...,\mathbf{x}_n)) = (\mathbf{x}_{\sigma(1)},...,\mathbf{x}_{\sigma(n)})$ for $\sigma \in \mathcal{S}_n$.

Definition 9. *An algorithm provides $\varepsilon-$unbounded DP (UDP) iff it provides $\varepsilon-$DP for the "removal" neighborhood relation $\overset{\mathrm{U}}{\sim}$, defined as*

$$\mathcal{X} \overset{\mathrm{U}}{\sim} \mathcal{Y} \Leftrightarrow \begin{cases} \big||\mathcal{X}| - |\mathcal{Y}|\big| = 1 \text{ (we can assume w.l.o.g. } |\mathcal{X}| = |\mathcal{Y}| + 1 \triangleq n \geq 2) \\ \exists\, \sigma \in \mathcal{S}_{|\mathcal{X}|} \text{ s.t. } \sigma(\mathcal{X}) \overset{\mathrm{U}}{\approx} \mathcal{Y}, \end{cases}$$

where $(\mathbf{x}_1,...,\mathbf{x}_n) \overset{\mathrm{U}}{\approx} (\mathbf{y}_1,...,\mathbf{y}_{n-1}) \Leftrightarrow ((\forall i \in [\![1,n-1]\!],\, \mathbf{x}_i = \mathbf{y}_i)$ and $\mathbf{x}_n$ is arbitrary).

Definition 10. *An algorithm provides ε-bounded DP (BDP) iff it provides $\varepsilon-$DP for the "replacement" neighborhood relation $\overset{\mathrm{B}}{\sim}$:*

$$\mathcal{X} \overset{\mathrm{B}}{\sim} \mathcal{Y} \Leftrightarrow |\mathcal{X}| = |\mathcal{Y}| \text{ and } \exists\, \sigma_1,\sigma_2 \in \mathcal{S}_{|\mathcal{X}|} \text{ s.t. } \sigma_1(\mathcal{X}) \overset{\mathrm{B}}{\approx} \sigma_2(\mathcal{Y}),$$
$$(\mathbf{x}_1,...,\mathbf{x}_n) \overset{\mathrm{B}}{\approx} (\mathbf{y}_1,...,\mathbf{y}_n) \Leftrightarrow \forall i \in [\![1,n-1]\!], \mathbf{x}_i = \mathbf{y}_i, \text{ and } \mathbf{x}_n, \mathbf{y}_n \text{ are arbitrary.}$$

We assume $|\mathcal{X}| = |\mathcal{Y}| + 1$ in the definition for succinctness only, but the relation $\overset{\mathrm{U}}{\sim}$ is symmetric. The key practical difference between the two definitions is that BDP assumes that the size of the dataset is not a sensitive value and can be published freely. Unbounded differential privacy is a stronger definition, as an ε-UDP algorithm is necessarily 2ε-BDP (using the composition lemmas presented below, and because if $\mathcal{X} \overset{\mathrm{B}}{\sim} \mathcal{Y}$, $\mathcal{X}$ can be obtained from $\mathcal{Y}$ by removing an element and adding a new one), while the reverse is not necessarily true. This bound might however not be tight. In the following, we mainly focus on the UDP setting, which is sometimes more tricky. However, most of the results are also adapted for BDP.

Composition An important property of differential privacy is *composition*: using several differentially private algorithms on the same dataset results in similar guarantees, but with a total privacy budget equal to the sum of the budgets of the individual algorithms. Hence, one can design a complex DP algorithm by splitting its privacy budget ε between different simpler routines.

Lemma 1 (Sequential composition [39, Theorem 3])**.** *Let $(R_i)_{1\leq i\leq r}$ be a collection of DP mechanisms on the same domain with respective privacy budgets $(\varepsilon_i)_{1\leq i\leq r}$. Then $R : \mathcal{X} \mapsto (R_1(\mathcal{X}),\dots,R_r(\mathcal{X}))$ provides $(\sum_{i=1}^r \varepsilon_i)-$DP.*

This holds for both bounded and unbounded DP. Parallel composition can also be performed; the following lemma however holds only in the unbounded case.

Lemma 2 (Parallel composition [39, Theorem 4])**.** *Let $(R_i)_{1\leq i\leq r}$ be a collection of independent ε-UDP algorithms on the same domain $\mathcal{D}$, and $\mathcal{D}_i$ be disjoint subsets of $\mathcal{D}$. Then $R : \mathcal{X} \mapsto (R_1(\mathcal{X}\cap\mathcal{D}_1),\dots,R_r(\mathcal{X}\cap\mathcal{D}_r))$ provides $\varepsilon-$UDP, where $(\mathbf{x}_1,...,\mathbf{x}_n)\cap\mathcal{D}_j$ denotes the subtuple with original ordering of the samples $(\mathbf{x}_i)_{1\leq i\leq n}$ that are in $\mathcal{D}_j$.*

These lemmas hold only when the R_i are differentially private according to the same neighboring relation between datasets. Note also that privacy is robust to post-processing: if a mechanism R is ε-DP, then $f(R(\cdot))$ is also ε-DP for any function f. Thus Lemma 2 implies in particular that in a distributed setting, each data holder can compute and release an ε-DP synopsis of its local data (e.g. a noisy sketch), and merging these quantities will lead to a global synopsis which is also ε-DP with respect to the whole dataset.

Alternative privacy definitions Many alternative definitions of privacy have been proposed in the literature [54]. Traditional statistical disclosure control metrics, such as $k-$anonymity [49], define anonymity as a property of the data, e.g. requiring that each user is indistinguishable from $k-1$ others. However, anonymizing large-scale high-dimensional data (such as, e.g., mobility datasets) was shown to be hard, due to the high uniqueness of users in such datasets [16]. Researchers have proposed to make privacy a property of the algorithm, enforcing for instance that the mutual information leakage is bounded [18]. Differential privacy is the most popular of such definitions, as it considers a worst-case adversary, and is hence "future-proof": no future release of auxiliary information can break the privacy guarantees. Connections between differential privacy and other information-theoretic definitions have also been investigated [57].

3.2 The Laplace Mechanism

In this section, we describe the Laplace mechanism [19], a very common and simple mechanism to release privately a function f computed over sensitive values. This mechanism adds Laplace noise to the function's output, whose scale ensures differential privacy. In the following, $\mathcal{L}(b)$ denotes the centered Laplace distribution of parameter b.

Definition 11 (Complex Laplace distribution)**.** *A random variable z follows a centered complex Laplace distribution of parameter b (denoted $z \sim \mathcal{L}^{\mathbb{C}}(b)$) iff its real and imaginary parts follow independently a real Laplace distribution of parameter b. In that case, z admits a density $p_z(z) \propto \exp(-(|\Re z| + |\Im z|)/b)$ and has variance $\sigma_z^2 = \mathbf{E}\left[|z|^2\right] = 4b^2$.*

Definition 12 (Laplace Mechanism)**.** *For any function $f : \mathcal{D} \to \mathbb{R}^m$ (resp. $\mathbb{C}^m$), the Laplace mechanism with parameter $b \in \mathbb{R}$ is the random mechanism $\mathcal{X} \mapsto f(\mathcal{X}) + \boldsymbol{\xi}$ where $(\xi_i)_{1 \leq i \leq m} \overset{\text{iid}}{\sim} \mathcal{L}(b)$ (resp. $\mathcal{L}^{\mathbb{C}}(b)$).*

The Laplace mechanism provides differential privacy if the scale b of the noise is chosen carefully. This scale depends on the notion of sensitivity, which measures the maximum variation of a function between two neighboring datasets.

Definition 13 (L^1-sensitivity)**.** *The L^1-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^m$ for a neighborhood relation $\sim$ is defined as*

$$\Delta_1(f) \triangleq \sup_{\mathcal{X},\mathcal{Y} \in \mathcal{D}:\, \mathcal{X} \sim \mathcal{Y}} \|f(\mathcal{X}) - f(\mathcal{Y})\|_1. \tag{7}$$

This definition extends to complex-valued functions by the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$.

Throughout the paper, we will use superscripts Δ_1^{U} and Δ_1^{B} to denote sensitivities computed respectively w.r.t. the UDP and BDP neighboring relations. Dwork et. al [20] proved that the Laplace mechanism provides ε-differential privacy for the noise level $b = \Delta_1(f)/\varepsilon$. We propose below a straightforward extension of this result for the complex setting. Although only an upper bound on the sensitivity is required in order to prove that a mechanism is differentially private, we will also provide sharp bounds when possible, hence the notion of "sharp privacy level".

Theorem 1. *Let $f : \mathcal{D} \to \mathbb{R}^m$ or $\mathbb{C}^m$. The Laplace mechanism applied on f is differentially private with sharp privacy budget $\varepsilon^* = \Delta_1(f)/b$. For $\varepsilon > 0$, the lowest noise level yielding ε-differential privacy is given by $b^* = \Delta_1(f)/\varepsilon$. This holds for both bounded and unbounded DP, provided that the sensitivities are computed according to the relevant neighborhood relation.*

Proof. Let $\mathcal{X}, \mathcal{Y} \in \mathcal{D}$ be such that $\mathcal{X} \sim \mathcal{Y}$. Let $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ denote the probability densities of the Laplace mechanism applied on f for datasets $\mathcal{X}$ and $\mathcal{Y}$. In the real case, the privacy loss function takes the form

$$L_f(\mathbf{s}, \mathcal{X}, \mathcal{Y}) = \log\left(\frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})}\right) = \frac{1}{b}(\|f(\mathcal{Y}) - \mathbf{s}\|_1 - \|f(\mathcal{X}) - \mathbf{s}\|_1)$$

Hence:
$$\sup_{\substack{\mathbf{s} \in \mathbb{R}^m \\ \mathcal{X},\mathcal{Y} \in \mathcal{D}:\, \mathcal{X} \sim \mathcal{Y}}} L_f(\mathbf{s}, \mathcal{X}, \mathcal{Y}) = \frac{1}{b} \sup_{\mathcal{X},\mathcal{Y} \in \mathcal{D}:\, \mathcal{X} \sim \mathcal{Y}} \sum_{j=1}^{m} \sup_{s_j \in \mathbb{R}} |f(\mathcal{Y})_j - \mathbf{s}_j| - |f(\mathcal{X})_j - \mathbf{s}_j|$$
$$\overset{(*)}{=} \sup_{\mathcal{X},\mathcal{Y} \in \mathcal{D}:\, \mathcal{X} \sim \mathcal{Y}} \frac{\|f(\mathcal{X}) - f(\mathcal{Y})\|_1}{b} = \frac{\Delta_1(f)}{b}.$$

The inequality $\leq$ in (*) follows from the triangle inequality; $s_j = f(\mathcal{Y})_j$ shows the equality. In the complex case, the proof is similar but using the density of a complex Laplace variable (Definition 11), and the definition of L^1-sensitivity in the complex case. □

Note that the function $f : \mathcal{X} \mapsto |\mathcal{X}|$ has UDP/BDP sensitivities $\Delta_1^{\mathrm{U}}(f) = 1$ and $\Delta_1^{\mathrm{B}}(f) = 0$, as all neighboring datasets have the same size for BDP. Releasing n publicly is therefore ε-BDP for any value of ε, but this is not the case with UDP. This confirms the intuition that UDP treats the dataset size as sensitive, while BDP does not.

3.3 Approximate Differential Privacy and the Gaussian Mechanism

Differential privacy is a very strong guarantee, and for many real-world tasks it can lead to severe degradations of the algorithms performance (utility) for small privacy budgets. For this reason, many relaxations of DP have been introduced, the most prominent of which is approximate differential privacy, also commonly called (ε, δ)-DP [20].

Definition 14 (Approximate differential privacy [20])**.** *The randomized mechanism R achieves $(\varepsilon, \delta)-$approximate differential privacy (noted $(\varepsilon, \delta)-DP$) for $\varepsilon > 0$, $\delta \geq 0$ iff for any measurable set S of the co-domain of R, and any $\mathcal{X}, \mathcal{Y} \in \mathcal{D}$ s.t. $\mathcal{X} \sim \mathcal{Y}$ for some neighboring relation:*

$$\mathbb{P}\left[R(\mathcal{X}) \in S\right] \leq \exp(\varepsilon) \cdot \mathbb{P}\left[R(\mathcal{Y}) \in S\right] + \delta. \tag{8}$$

The most common mechanism to achieve $(\varepsilon, \delta)-$DP is the Gaussian mechanism, adding Gaussian noise to the output of a function. As for the Laplace mechanism, we here consider potentially complex-valued outputs, and denote $z \sim \mathcal{N}^{\mathbb{C}}(0, \sigma^2)$ a random variable whose real and imaginary component are independently identically distributed as $\Re z, \Im z \sim \mathcal{N}(0, \sigma^2)$ (note that the variance of z then reads $\sigma_z^2 = 2\sigma^2$).

Definition 15 (Gaussian Mechanism)**.** *For any $f : \mathcal{D} \to \mathbb{R}^m$ (resp. $\mathbb{C}^m$), the Gaussian mechanism with parameter σ is the random mechanism $\mathcal{X} \mapsto f(\mathcal{X}) + \boldsymbol{\xi}$ where $(\xi_j)_{1 \leq j \leq m} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$ (resp. $\mathcal{N}^{\mathbb{C}}(0, \sigma^2)$).*

The advantage of this DP relaxation is that the noise standard deviation needed for $(\varepsilon, \delta)-$DP scales not with the L^1 but with the L^2 sensitivity of f, defined just below, which can be significantly smaller for many functions, including our sketching operator.

Definition 16 (L^2-sensitivity)**.** *The L^2-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^m$ for a neighborhood relation $\sim$ is defined as $\Delta_2(f) \triangleq \sup_{\mathcal{X}, \mathcal{Y} \in \mathcal{D} : \mathcal{X} \sim \mathcal{Y}} \|f(\mathcal{X}) - f(\mathcal{Y})\|_2$. This definition extends to complex-valued functions using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$.*

The "classical" noise calibration for the (real) Gaussian mechanism comes from [21, Appendix A], which shows that, assuming $\varepsilon < 1$, a standard deviation $\sigma > (2\ln(1.25/\delta))^{0.5}\Delta_2(f)/\varepsilon$ is sufficient to guarantee $(\varepsilon, \delta)-$DP. This bound is commonly used but not sharp, especially in the high privacy regime (i.e. small ε), and restricted to $\varepsilon < 1$. The calibration of the required noise parameter σ has recently been carefully tightened by Balle et al. [5], which is the mechanism we will use in this work[2].

Theorem 2 (Analytical Gaussian mechanism [5, Theorem 9])**.** *For each $\varepsilon, \delta > 0$, the lowest noise level σ^* such that the (real) Gaussian mechanism provides $(\varepsilon, \delta)-DP$ is given by $\sigma^* = \eta(\varepsilon, \delta)\frac{\Delta_2(f)}{\sqrt{2\varepsilon}}$, where $\eta(\varepsilon, \delta)$ is described in [5] and can be computed with a numerical algorithmic procedure.*

Note that the term $\eta(\varepsilon, \delta)$ depends on ε, hence it is incorrect to say that σ^* scales in $\varepsilon^{-1/2}$. In particular, when $\varepsilon \to 0$ the noise level converges to a finite constant [5, Section 2.1]. Compared to the standard Gaussian mechanism of Dwork and Roth, Theorem 2 has also the advantage to hold in the low-privacy regime (i.e. when $\varepsilon > 1$).

The result holds for complex-valued feature maps as well using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$, as applying the complex Gaussian mechanism on a complex-valued $\boldsymbol{\Phi}(\cdot)$ is equivalent to applying the real Gaussian mechanism to $[\Re\boldsymbol{\Phi}(\cdot); \Im\boldsymbol{\Phi}(\cdot)]$, and $\|[\Re\boldsymbol{\Phi}(\cdot); \Im\boldsymbol{\Phi}(\cdot)]\|_2 = \|\boldsymbol{\Phi}(\cdot)\|_2$.

We expect from the literature and from the definitions of the sensitivities that using the Gaussian mechanism should help to reduce the noise level required to achieve privacy, and thus increase the overall

[2] An implementation can be found at https://github.com/BorjaBalle/analytic-gaussian-mechanism.

learning performance. We will see in particular in the next section that the L^1 and L^2 sensitivities scale with the sketch size m respectively in m and $m^{1/2}$.

Note that simple composition theorems also exist for approximate differential privacy similarly to Lemma 1. We provide here only the result on sequential composition for succinctness, but results on parallel composition can be found in the literature as well.

Lemma 3 (Sequential composition [21, Theorem 3.16])**.** *Let $(R_i)_{1\leq i\leq r}$ be a collection of $(\varepsilon_i, \delta_i)$-DP mechanisms on the same domain. Then $R : \mathcal{X} \mapsto (R_1(\mathcal{X}), \dots, R_r(\mathcal{X}))$ provides $(\sum_{i=1}^r \varepsilon_i, \sum_{i=1}^r \delta_i)-DP$.*

We now explain how the privacy definitions introduced in this section can be satisfied with the sketching framework.

4 Differentially Private Sketching

Sketching, as proposed in Definition 1, is not sufficient per se to ensure the differential privacy of user contributions, despite the fact that the sketch itself (which is just at most $m \ll nd$ real or complex numbers) cannot contain much information about each of the n samples $\mathbf{x}_i \in \mathbb{R}^d$. In particular, although the vectors $(\boldsymbol{\omega}_j)_{j=1}^m$ are randomly drawn, the sketching mechanism induced by a given set of such vectors is deterministic. We construct a noisy sketch, based on the Laplacian (resp. Gaussian) mechanism, that guarantees ε-differential privacy (resp. (ε, δ)-differential privacy).

The clean sketch $\mathbf{z}_\mathcal{X}$ from (1) can be written $\mathbf{z}_\mathcal{X} = \boldsymbol{\Sigma}(\mathcal{X})/|\mathcal{X}|$, where $\boldsymbol{\Sigma}(\mathcal{X}) \triangleq \sum_{i=1}^n \boldsymbol{\Phi}(\mathbf{x}_i)$ denotes the sum of features and $|\mathcal{X}|$ the number of records. Our mechanism adds noise to the numerator and denominator separately, i.e. releases $(\boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi}, |\mathcal{X}| + \zeta)$ where both $\boldsymbol{\xi}$ and ζ are random. Both quantities are thus made private provided that the noise levels are properly chosen, as discussed in the following subsections. This also means that we can further average sketches after computation in a distributed setting. The sketch $\mathbf{z}_\mathcal{X}$ can then be estimated from these two quantities, e.g. using $\mathbf{s}(\mathcal{X}) \triangleq (\boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi})/(|\mathcal{X}| + \zeta)$, which is private by composition properties of differential privacy. Note that DP is also robust to postprocessing, so one could for instance replace $|\mathcal{X}| + \zeta$ by $\max(|\mathcal{X}| + \zeta, 1)$ to avoid dividing by a null or negative quantity. The noise $\boldsymbol{\xi}$ added to $\boldsymbol{\Sigma}$ can be either Laplacian or Gaussian, and we provide guarantees for both cases respectively in Section 4.1 and Section 4.2, each time for both random Fourier features and PCA. In the following, we use the notations $\boldsymbol{\Sigma}^{\mathrm{RFF}}$ and $\boldsymbol{\Sigma}^{\mathrm{RQF}}$ when $\boldsymbol{\Sigma}$ is computed using respectively $\boldsymbol{\Phi} = \boldsymbol{\Phi}^{\mathrm{RFF}}$ and $\boldsymbol{\Phi} = \boldsymbol{\Phi}^{\mathrm{RQF}}$.

4.1 Private Sketching with the Laplace Mechanism

We introduce formally the noisy sum of features.

Definition 17. *The noisy sum of features $\boldsymbol{\Sigma}_\mathcal{L}$ of a dataset $\mathcal{X} = (\mathbf{x}_i)_{i=1}^n \in \mathcal{D}_n$ with noise parameters b is the random variable*

$$\boldsymbol{\Sigma}_\mathcal{L}(\mathcal{X}) = \boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi},$$

where $\boldsymbol{\Sigma}(\mathcal{X}) \triangleq \sum_{i=1}^n \boldsymbol{\Phi}(\mathbf{x}_i)$ and $\forall j \in [\![1, m]\!]$, $\xi_j \overset{\text{iid}}{\sim} \begin{cases} \mathcal{L}^{\mathbb{C}}(b) & \text{if } \boldsymbol{\Phi} \text{ is complex-valued} \\ \mathcal{L}(b) & \text{if } \boldsymbol{\Phi} \text{ is real-valued} \end{cases}$.

The scale of the noise will depend on the feature map used. Remember that we need an estimate of the sketch, and not just the sum of features. We introduce a generic lemma for this purpose.

Lemma 4. *For any privacy parameter $\varepsilon > 0$ and any choice of $\varepsilon_1, \varepsilon_2 > 0$ such that $\varepsilon_1 + \varepsilon_2 = \varepsilon$, if $\boldsymbol{\Sigma}$ has a finite sensitivity $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma})$, then any mechanism to estimate $\mathbf{z}_\mathcal{X}$ using $\boldsymbol{\Sigma}_\mathcal{L}(\mathcal{X})$ instantiated with a noise level $b = \Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma})/\varepsilon_1$ and $|\mathcal{X}| + \zeta$, where $\zeta \sim \mathcal{L}(\varepsilon_2^{-1})$, is ε-UDP.*

Proof. The Laplace mechanism applied on $\boldsymbol{\Sigma}$ with $b = \Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma})/\varepsilon_1$ is ε_1-UDP according to Theorem 1. Releasing $|\mathcal{X}|$ has sensitivity 1, and thus releasing $|\mathcal{X}| + \zeta$ with $\zeta \sim \mathcal{L}(\varepsilon_2^{-1})$ is ε_2-UDP. The result comes from the sequential composition Lemma 1. □

To prove differential privacy of the sketching mechanism, we thus only need to compute the sensitivity $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma})$ of the sum-of-features function. We will see in Section 4.2 that a similar result can be stated for the Gaussian mechanism using the L^2 sensitivity. We introduce in the following lemma a common

expression to deal with the different cases — Laplacian and Gaussian mechanisms, real- and complex-valued feature maps.

Lemma 5. *Let $\boldsymbol{\Sigma} : (\mathbf{x}_1, ..., \mathbf{x}_n) \mapsto \sum_{1 \leq i \leq |\mathcal{X}|} \boldsymbol{\Phi}(\mathbf{x}_i)$ where $\boldsymbol{\Phi}$ is any feature map taking values in $\mathbb{R}^m$ or $\mathbb{C}^m$. For $p = 1, 2$, the L^p sensitivity of $\boldsymbol{\Phi}$ for datasets on a domain E is*

$$\Delta_p^{\mathrm{U}}(\boldsymbol{\Sigma}) = \sup_{\mathbf{x} \in E} Q_p^{\mathrm{U}}(\mathbf{x})$$

where $Q_p^{\mathrm{U}}(\mathbf{x}) = \|\boldsymbol{\Phi}(\mathbf{x})\|_p$ for real-valued features maps, and extends to complex-valued feature maps using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$.

Note that in particular, $Q_1^{\mathrm{U}}(\mathbf{x}) = \|\mathcal{R}(\boldsymbol{\Phi}(\mathbf{x}))\|_1 + \|\mathcal{I}(\boldsymbol{\Phi}(\mathbf{x}))\|_1$ for a complex-valued $\boldsymbol{\Phi}$.

Proof. For a real-valued feature map $\boldsymbol{\Phi}$, we have by definitions 9, 13 and 16:

$$\Delta_p^{\mathrm{U}}(\boldsymbol{\Sigma}) = \sup_{\mathcal{X}, \mathcal{Y} \in \mathcal{D} : \mathcal{X} \overset{\mathrm{U}}{\sim} \mathcal{Y}} \|\boldsymbol{\Sigma}(\mathcal{X}) - \boldsymbol{\Sigma}(\mathcal{Y})\|_p = \sup_{\substack{\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in \mathcal{D}, \\ \mathcal{Y} = (\mathbf{y}_1, ..., \mathbf{y}_{n-1}) \in \mathcal{D}, \\ \text{such that } \mathcal{X} \overset{\mathrm{U}}{\sim} \mathcal{Y}}} \left\| \sum_{i=1}^{n} \boldsymbol{\Phi}(\mathbf{x}_i) - \sum_{i=1}^{n-1} \boldsymbol{\Phi}(\mathbf{y}_i) \right\|_p = \sup_{\mathbf{x} \in E} \|\boldsymbol{\Phi}(\mathbf{x})\|_p. \quad (9)$$

The result extends to the complex case using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$, with $\|[\mathcal{R}(\boldsymbol{\Phi}(\mathbf{x})); \mathcal{I}(\boldsymbol{\Phi}(\mathbf{x}))]\|_2 = \|\boldsymbol{\Phi}(\mathbf{x})\|_2$ and $\|[\mathcal{R}(\boldsymbol{\Phi}(\mathbf{x})); \mathcal{I}(\boldsymbol{\Phi}(\mathbf{x}))]\|_1 = \|\mathcal{R}(\boldsymbol{\Phi}(\mathbf{x}))\|_1 + \|\mathcal{I}(\boldsymbol{\Phi}(\mathbf{x}))\|_1$. □

In the following, we use the notations $Q_p^{\mathrm{RFF}}, Q_p^{\mathrm{RQF}}$ when the feature maps $\boldsymbol{\Phi}^{\mathrm{RFF}}, \boldsymbol{\Phi}^{\mathrm{RQF}}$ are used. Note however that this Lemma is generic, and could be applied to any new feature map in the future. We compute $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}})$ and $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}})$ using Lemma 5 respectively in Section 4.1.1 and Section 4.1.2. Note that we compute the sensitivities using the expression of the feature maps given in Definitions 4 and 6, but any constant factor $c_{\boldsymbol{\Phi}}$ could be used in these expressions provided that the inverse problem is solved using the same scaling (one could for instance use $c_{\boldsymbol{\Phi}} = 1/\sqrt{m}$ to get normalized features); this would yield similar privacy guarantees, but the sensitivity and thus the noise level b would be multiplied by the same factor. We discuss how to optimally split the privacy budget between ε_1 and ε_2 in Section 6.5.

4.1.1 Random Fourier Features

We compute the L^1-sensitivity of $\boldsymbol{\Sigma}^{\mathrm{RFF}}$ in Lemma 7. We first introduce a lemma on diophantine approximation, that will be needed to prove the sharpness of our bound.

Definition 18. *The scalars $(\omega_j)_{1 \leq j \leq m} \in \mathbb{R}$ are called nonresonant frequencies [52] if they are linearly independent over the rationals. The vectors $(\boldsymbol{\omega}_j)_{1 \leq j \leq m} \in \mathbb{R}^d$ are called nonresonant frequency vectors if there exists a vector $\mathbf{v} \in \mathbb{R}^d$ such that the scalars $(\boldsymbol{\omega}_j^T \mathbf{v})_{1 \leq j \leq m}$ are nonresonant frequencies.*

Lemma 6. *Let $(\varphi_j)_{1 \leq j \leq m}$ be real numbers, $(\boldsymbol{\omega}_j)_{1 \leq j \leq m} \in \mathbb{R}^d$ nonresonant frequencies, and f a 2π-periodic function such that there exists z at which f is continuous and reaches its maximum. Then $\sup_{\mathbf{x} \in \mathbb{R}^d} \inf_{j \in [\![1;m]\!]} f(\boldsymbol{\omega}_j^T \mathbf{x} - \varphi_j) = \sup_{x \in \mathbb{R}} f(x)$.*

The proof is in Appendix A. We can now compute the desired sensitivity.

Lemma 7. *The function $\boldsymbol{\Sigma}^{\mathrm{RFF}}$ built using m frequencies has sensitivity $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) \leq m\sqrt{2}$ for both quantized and unquantized cases. If the frequencies are non resonant then $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = m\sqrt{2}$.*

Proof. We recall that $\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}) = \left(\rho(\Omega^T \mathbf{x} + \mathbf{u}) + \mathrm{i}\rho(\Omega^T \mathbf{x} + \mathbf{u} - \frac{\pi}{2})\right)$ in order to deal with both unquantized ($\rho = \cos$, $\mathbf{u} = 0$) and quantized ($\rho = 2^{-1/2} \operatorname{sign} \circ \cos$, $\mathbf{u} \in [0, 2\pi[^m$) mechanisms with the same formalism. Using the definition of Q_p from Lemma 5 in the Laplace real case, we have

$$Q_1^{\mathrm{RFF}}(\mathbf{x}) \triangleq \|\Re(\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}))\|_1 + \|\Im(\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x}))\|_1 = \sum_{j=1}^{m} |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j)| + |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j - \frac{\pi}{2})|$$

Denoting $f(\cdot) \triangleq \rho(\cdot) + \rho(\cdot - \pi/2)$ we show that $|\rho(\cdot)| + |\rho(\cdot - \pi/2)| = \sup_{\varphi \in \{0,\pi/2,\pi,3\pi/2\}} f(\cdot - \varphi)$. Indeed, both $\rho = \cos$ and $\rho = 2^{-1/2}\,\mathrm{sign} \circ \cos$ satisfy the property $\forall t : \rho(t) = -\rho(t - \pi)$, hence for each $t \in \mathbb{R}$:

$$\begin{aligned} +\rho(t) + \rho(t - \pi/2) &= f(t) \\ +\rho(t) - \rho(t - \pi/2) &= \rho(t) + \rho(t + \pi/2) = f(t + \pi/2) \\ -\rho(t) - \rho(t - \pi/2) &= -f(t) = f(t + \pi) \\ -\rho(t) + \rho(t - \pi/2) &= -f(t + \pi/2) = f(t + 3\pi/2). \end{aligned}$$

As a result, denoting $f_{\boldsymbol{\varphi}}(\mathbf{x}) \triangleq \sum_{j=1}^m f(\boldsymbol{\omega}_j^T \mathbf{x} - \varphi_j)$ for each $\boldsymbol{\varphi} \in \mathbb{R}^m$, we obtain

$$Q_1^{\mathrm{RFF}}(\mathbf{x}) = \sum_{j=1}^m \sup_{\varphi_j \in \{0,\pi/2,\pi,3\pi/2\}} f(\boldsymbol{\omega}_j^T \mathbf{x} + u_j - \varphi_j) = \sup_{\boldsymbol{\varphi} \in \{0,\pi/2,\pi,3\pi/2\}^m} f_{\boldsymbol{\varphi} - \mathbf{u}}(\mathbf{x}). \tag{10}$$

In the complex exponential case $\rho = \cos$ and $f : x \mapsto \sqrt{2}\cos(x - \pi/4)$. In the quantized case as $\rho = 2^{-1/2}\,\mathrm{sign} \circ \cos$, f is a piecewise constant function taking values $0, \sqrt{2}, 0, -\sqrt{2}$. Thus in both cases we have $\sup_{x \in \mathbb{R}} f(x) = \sqrt{2}$. We obtain $\sup_{\mathbf{x} \in \mathbb{R}^d} f_{\boldsymbol{\varphi} - \mathbf{u}}(\mathbf{x}) \leq m\sqrt{2}$ for any $\boldsymbol{\varphi}, \mathbf{u}$ hence, by Lemma 5, we get that $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) \leq m\sqrt{2}$ as claimed.

When the frequencies $(\boldsymbol{\omega}_j)_{1 \leq j \leq m}$ are nonresonant, f being 2π periodic and admitting (in both quantized/unquantized cases) a point $z \in \mathbb{R}$ at which it reaches its maximum and is continuous, we apply Lemma 6 and get according to Lemma 5:

$$\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sup_{\mathbf{x} \in \mathbb{R}^d} Q_1^{\mathrm{RFF}}(\mathbf{x}) = \sup_{\boldsymbol{\varphi} \in \{0,\pi/2,\pi,3\pi/2\}^m} \sup_{\mathbf{x} \in \mathbb{R}^d} f_{\boldsymbol{\varphi} - \mathbf{u}}(\mathbf{x}) = m\sqrt{2}, \tag{11}$$

where the supremum is independent of the choice of $\boldsymbol{\varphi}$. □

Note that this holds only for $E = \mathbb{R}^d$. If the domain is restricted to e.g. $E = \mathcal{B}_2 = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq 1\}$ the upper bound may not be reached, even with nonresonant frequencies, so an improved privacy may be possible.

For this result to be applicable, we still need to prove that the frequencies are nonresonant in practice.

Lemma 8. *Frequency vectors drawn i.i.d. according to a distribution which is absolutely continuous w.r.t. the Lebesgue measure are almost surely nonresonant.*

Proof. The set of resonant frequencies has a zero Lebesgue measure. The reader can refer to [52, Corollary 9.3 p. 166] for a proof relying on strong incommensurability. □

4.1.2 Random Quadratic Features

In this section only, we restrict ourselves to datasets whose elements are bounded by 1 in L^2-norm. The domain is thus $E = \mathcal{B}_2 \triangleq \left\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\right\}$, and we still use the notations $\mathcal{D}_n \triangleq E^n$ and $\mathcal{D} \triangleq \cup_{n \in \mathbb{N}} \mathcal{D}_n$.

Lemma 9. *The function $\boldsymbol{\Sigma}^{\mathrm{RQF}}$ built using a matrix of frequencies $\Omega = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m]$, has sensitivity $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = \|\Omega\|_2^2$ where $\|\cdot\|_2$ denotes the spectral norm.*

Proof. Let $\lambda_{\max}$ denote the largest eigenvalue function. We have according to Lemma 5

$$\begin{aligned} \Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) &= \sup_{\mathbf{x} \in \mathcal{B}_2} Q_1^{\mathrm{RQF}}(\mathbf{x}) = \sup_{\mathbf{x} \in \mathcal{B}_2} \|\boldsymbol{\Phi}^{\mathrm{RQF}}(\mathbf{x})\|_1 \\ &= \sup_{\mathbf{x} \in \mathcal{B}_2} \sum_{j=1}^m (\omega_j^T \mathbf{x})^2 = \sup_{\mathbf{x} : \|\mathbf{x}\| \leq 1} \mathbf{x}^T \Big(\sum_{j=1}^m \boldsymbol{\omega}_j \boldsymbol{\omega}_j^T\Big) \mathbf{x} = \lambda_{\max}(\Omega\Omega^T) = \|\Omega\|_2^2. \end{aligned}$$

□

The quantity $\|\Omega\|_2^2$ can be computed numerically for a given Ω. When m is a multiple of d and Ω is a concatenation of m/d orthonormal bases as detailed in Section 2.2, we have $\Omega\Omega^T = m/d\mathbf{I}_d$ and thus $\|\Omega\|_2^2 = m/d$. When Ω has i.i.d. $\mathcal{N}(0, 1/d)$ entries, $\|\Omega\|_2^2$ is of the same order with high probability.

4.2 Approximate Differential Privacy with the Gaussian Mechanism

In practice, in order to increase the utility of private mechanisms relying on additive perturbations, ε-DP is often relaxed to approximate (ε, δ)-DP. In this section we provide an (ε, δ)-DP sketching mechanism based on the Gaussian mechanism.

Definition 19. *The Gaussian noisy sum of features $\boldsymbol{\Sigma}_{\mathcal{G}}(\mathcal{X})$ of a dataset $\mathcal{X} = (\mathbf{x}_i)_{i=1}^n$ with noise parameters σ is the random variable*

$$\boldsymbol{\Sigma}_{\mathcal{G}}(\mathcal{X}) = \boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi}$$

where $\boldsymbol{\Sigma}(\mathcal{X}) \triangleq \sum_{i=1}^n \boldsymbol{\Phi}(\mathbf{x}_i)$ and $\forall j \in [\![1, m]\!]$, $\xi_j \overset{\text{iid}}{\sim} \begin{cases} \mathcal{N}^{\mathbb{C}}(0, \sigma^2) \text{ if } \boldsymbol{\Phi} \text{ is complex-valued} \\ \mathcal{N}(0, \sigma^2) \text{ if } \boldsymbol{\Phi} \text{ is real-valued} \end{cases}$.

The only difference with Definition 17 is that the noise added on the sum of features $\boldsymbol{\Sigma}(\mathcal{X})$ is Gaussian.

We now introduce an equivalent of the composition lemma 4 for the Gaussian case.

Lemma 10. *For any privacy parameter $\varepsilon > 0$ and choice of $\varepsilon_1, \varepsilon_2 > 0$ such that $\varepsilon_1 + \varepsilon_2 = \varepsilon$, if $\boldsymbol{\Sigma}_{\mathcal{G}}$ has finite L^2 sensitivity $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma})$, then any mechanism to estimate $\mathbf{z}_{\mathcal{X}}$ using $\boldsymbol{\Sigma}_{\mathcal{G}}(\mathcal{X})$ with noise level $\sigma = \eta(\varepsilon_1, \delta) \cdot \Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma})/\sqrt{2\varepsilon_1}$ (where η refers to Theorem 2), and $|\mathcal{X}| + \zeta$ where $\zeta \sim \mathcal{L}(\varepsilon_2^{-1})$, is ε-UDP.*

Proof. The Gaussian mechanism applied on $\boldsymbol{\Sigma}$ with $\sigma = \eta(\varepsilon_1, \delta)\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma})/\sqrt{2\varepsilon_1}$ is (ε_1, δ)-UDP according to Theorem 2. As in lemma 4, releasing $|\mathcal{X}| + \zeta$ with $\zeta \sim \mathcal{L}(\varepsilon_2^{-1})$ is $(\varepsilon_2, 0)$-UDP. The result comes from Lemma 3 on sequential composition of approximate differential privacy. □

Note that we add Laplacian noise on the dataset size; if Gaussian noise was added we would have to split not only ε but also δ between the sum of features and the dataset size. As there is no difference between $\Delta_1^{\mathrm{U}}(|\cdot|)$ and $\Delta_2^{\mathrm{U}}(|\cdot|)$, allocating a part of δ to the denominator would not bring any substantial gain compared to putting all the budget on the numerator.

We now compute the sensitivities $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}})$ (Section 4.2.1) and $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}})$ (Section 4.2.2). Here again, in case the feature maps are multiplied by a constant factor, the L^2 sensitivity and thus the noise level σ need to be multiplied by the same factor.

4.2.1 Random Fourier Features

For random Fourier features, computing the L^2 sensitivity is much more straightforward than the L^1 sensitivity, as each component of the feature map has a constant modulus. We get the following result.

Lemma 11. *The function $\boldsymbol{\Sigma}^{\mathrm{RFF}}$ has sensitivity $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sqrt{m}$ for both quantized and unquantized cases.*

Proof. Using the fact that $|\boldsymbol{\Phi}(\mathbf{x})_j| = 1$ for any j and $\mathbf{x}$, we have by Lemma 5

$$\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sup_{\mathbf{x} \in \mathbb{R}^d} Q_2^{\mathrm{RFF}}(\mathbf{x}) = \sup_{\mathbf{x} \in \mathbb{R}^d} \sqrt{\sum_{j=1}^m \left|\boldsymbol{\Phi}_j(\mathbf{x})\right|^2} = \sqrt{m}.$$ □

As expected, the standard deviation of the Gaussian noise is smaller than the standard deviation of the Laplacian noise that one would need to add in order to reach the same privacy level with the Laplace mechanism. Indeed, the L^2 sensitivity only scales with $\sqrt{m}$, where the L^1 sensitivity was scaling linearly with m.

For bounded differential privacy, we have the following result.

Lemma 12. *The function $\boldsymbol{\Sigma}^{\mathrm{RFF}}$ computed with nonresonant features has sensitivity $\Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = 2\sqrt{m}$ for both quantized and unquantized cases.*

The proof of this result can be found in Appendix B.

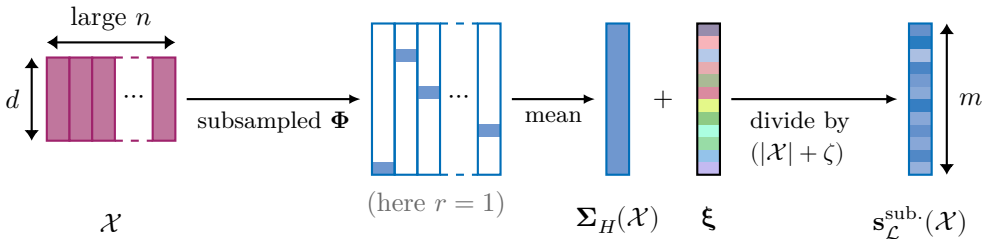

Figure 3: Overview of the sketching mechanism from Definition 21 with subsampling.

4.2.2 Random Quadratic Features

In this subsection, we consider again datasets whose elements are bounded by 1 in L^2-norm, and reuse the notations $E = \mathcal{B}_2, \mathcal{D}_n = E^n, \mathcal{D} \triangleq \cup_{n\in\mathbb{N}}\mathcal{D}_n$.

Lemma 13. *The function $\mathbf{\Sigma}^{\text{RQF}}$ built using a matrix of frequencies $\Omega = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m]$, has L^2 sensitivity $\Delta_2^{\text{U}}(\mathbf{\Sigma}^{\text{RQF}}) = S_4(\Omega)$, where $S_4(\Omega) = \left(\sup_{\mathbf{x}:\|\mathbf{x}\|_2\leq 1} \sum_{j=1}^m (\omega_j^T\mathbf{x})^4\right)^{1/2}$.*

Proof. We have by Lemma 5

$$\Delta_2^{\text{U}}(\mathbf{\Sigma}^{\text{RQF}}) = \sup_{\mathbf{x}\in\mathcal{B}_2} Q_2^{\text{RQF}}(\mathbf{x}) = \sup_{\mathbf{x}\in\mathcal{B}_2} \|\mathbf{\Phi}^{\text{RQF}}(\mathbf{x})\|_2 = \left(\sup_{\mathbf{x}:\|\mathbf{x}\|_2\leq 1} \sum_{j=1}^m (\omega_j^T\mathbf{x})^4\right)^{1/2} = S_4(\Omega). \qquad \square$$

The quantity $S_4(\Omega)$ can be estimated numerically.

5 A faster mechanism with frequency subsampling

We now introduce a sketching mechanism that subsamples *the features* as shown in Figure 3, and then build on top of it a noisy sketch that guarantees ε-differential privacy.

This mechanism differs from the standard approach which consists in subsampling the data samples [4] rather than the features. The following result is for instance well known in the literature.

Lemma 14 ([4, Table 1])**.** *Let M be an ε-UDP mechanism, and denote by $\mathcal{S}$ the Poisson data-subsampling mechanism with parameter α (i.e. each data sample $\mathbf{x}_i$ is kept independently from the others with probability α). Then the mechanism $\mathcal{X} \mapsto M(\mathcal{S}(\mathcal{X}))$ is ε'-UDP with $\varepsilon' = \log(1 + \alpha(\exp(\varepsilon) - 1)) < \varepsilon$.*

We will see in Lemma 15 that this privacy level ε' is sharp when Lemma 14 is applied to our sketching mechanism, but also that the same bound is obtained — and is still sharp — when sampling the *features* with probability α rather than the data samples. Although sampling the data is more generic — this can be used for any application and any mechanism, while sampling the features is more specific —, both techniques are relevant in our setting, and we will see in Section 6.3 that subsampling the features rather than the data can in some cases yield better utility-privacy tradeoffs at constant computational complexity. We will focus mostly on pure differential privacy guarantees for simplicity and conciseness, and give a generic upper bound that applies to approximate differential privacy as well.

The proposed subsampling mechanism, which consists in computing only some of the m entries of $\mathbf{\Phi}(\mathbf{x}_i)$ for each data sample $\mathbf{x}_i$ as shown in Figure 3, is mainly introduced in order to reduce the computational complexity of the sketching operation. This complexity is dominated by the computation of all the $(\boldsymbol{\omega}_j^T\mathbf{x}_i)_{1\leq i\leq n, 1\leq j\leq m}$, i.e. by the matrix product $\Omega^T X$, which costs $\Theta(mdn)$ when using a dense matrix Ω. As shown below in Lemma 16, subsampling (whether it is on the data samples or on the features) does not bring any advantage in enforcing differential privacy, i.e. the noise level required to get privacy is at least the same as without subsampling. Indeed, the privacy "amplification" induced by the subsampling operation is compensated by the fact that the sketch must be properly rescaled afterwards. We will prove however that in some settings, the guarantees obtained with and without subsampling are exactly the same. Moreover, it will be shown in Section 6.4 that feature subsampling

can be performed for large collections without significantly damaging the utility of the sketch, which motivates our approach.

Subsampling does also reduce the amount of information that is released about each sample, although this has no impact on differential privacy guarantees. Indeed, in the extreme case of feature subsampling we will measure only one floating point (or complex) number per data sample. When using a quantized sketch, this is further reduced to one bit (or two) of information per sample. For instance, if we only have the quantized random Fourier measurement of a sample $\mathbf{x}$ associated to the frequency $\boldsymbol{\omega}_j$, we can can only infer that $\mathbf{x}$ belongs to a union of "slices" of the ambient space delimited by affine hyperplanes orthogonal to $\boldsymbol{\omega}_j$. But in practice these features are further averaged over the samples (such individual sketches are computed by the data holder but not released publicly), the subsampling is performed randomly (so that we don't know which entry of the sketch a given sample contributed to) and, in the differential privacy scenario, noise can still be added to the obtained sketch. Although we only focus on differential privacy in this paper, we expect that this variant of the framework would be beneficial when working with alternative privacy definitions that rely on average information-theoretic quantities, such as mutual information [57].

Subsampling schemes We define $\mathcal{H} \triangleq \{0,1\}^m$ the set of binary masks $\mathbf{h}$ and $\mathcal{H}^n$ the set of all possible tuples $(\mathbf{h}_1, \dots, \mathbf{h}_n)$ of n such masks. Pointwise multiplication is denoted $\odot$. In the following, we consider a real number $0 < \alpha \leq 1$ and denote $\mathcal{P}_\alpha$ the set of probability distributions $p_\mathbf{h}$ on $\mathcal{H}$ satisfying $\forall j \in [\![1, m]\!]\, \mathbf{E}_{\mathbf{h}\sim p_\mathbf{h}} \mathbf{h}_j = \alpha$. Particular examples of probability distributions belonging to $\mathcal{P}_\alpha$ include

- **Poisson feature sampling:** the distribution $(\mathrm{Bern}\,(\alpha))^m$, corresponding to masks which m entries are drawn i.i.d. according to a Bernoulli distribution with parameter α;
- **Poisson data sampling:** the masks are $\chi \mathbf{1}$ with $\chi \sim \mathrm{Bern}\,(\alpha)$. This corresponds to subsampling the data rather than the features, which is a well known strategy as discussed above.
- **Uniform feature sampling:** the uniform distribution $\mathcal{U}(\mathcal{H}_r)$ over

$$\mathcal{H}_r \triangleq \left\{ \mathbf{h} \in \mathcal{H} \,|\, \sum_{i=1}^{m} h_i = r \right\}, \tag{12}$$

  where $1 \leq r \leq m$ is an integer, in which case $\mathcal{U}(\mathcal{H}_r) \in \mathcal{P}_\alpha$ with $\alpha \triangleq r/m$;
- **Block-uniform feature sampling:** when m/d is an integer and r is a multiple of d, $\mathcal{U}(\mathcal{H}_r^{\text{struct.}})$ is the uniform distribution over $\mathcal{H}_r^{\text{struct.}}$, the subset of $\mathcal{H}_r$ containing only the vectors which are structured by blocs of size d, i.e. $\mathcal{H}_r^{\text{struct.}} \triangleq \{\mathbf{h} = [h_1, ..., h_m] \in \mathcal{H}_r | \forall i \in [\![1, m/d]\!], h_{(i-1)d+1} = h_{(i-1)d+2} = ... = h_{id}\}$. In this case we also have $\mathcal{U}(\mathcal{H}_r^{\text{struct.}}) \in \mathcal{P}_\alpha$ with $\alpha \triangleq r/m$. This scheme will be useful when Ω is a structured transform, as explained in the next paragraph.

A note on sketching complexity When computing $r = \lceil \alpha m \rceil$ features per input sample rather than computing the whole matrix product $\Omega^T X$, the sketching complexity goes down from $\Theta(mdn)$ to $\Theta(rdn)$. In the high-dimensional setting, previous works [12] suggested to speed such computations by using structured matrices Ω made of $\lceil m/d \rceil$ square $d \times d$ blocks associated to as many fast transforms. In that case, the matrix-vector multiplication for each square block is performed at once using the corresponding fast transform with complexity $\Theta(d \log(d))$. We can thus rely on block-uniform subsampling mechanism introduced above using $r = d$, so that for each data sample $\mathbf{x}_i$ we compute the d measurements associated to a randomly chosen block. The sketching cost is then $\Theta(d \log(d) n)$, while computing the same number $r = d$ of measurements with a dense matrix Ω would have scaled in $\Theta(d^2 n)$.

Sketching with subsampling We first define how features are subsampled using a fixed tuple of masks, and then define the sketching mechanism using random masks.

Definition 20. *The sum of subsampled features of a dataset $\mathcal{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, using a fixed set of binary masks $H = (\mathbf{h}_1, ..., \mathbf{h}_n) \in \mathcal{H}^n$ that has been drawn according to some distribution in $\mathcal{P}_\alpha$ is defined as*

$$\boldsymbol{\Sigma}_H(\mathcal{X}) \triangleq \frac{1}{\alpha} \sum_{i=1}^{n} \boldsymbol{\Phi}(\mathbf{x}_i) \odot \mathbf{h}_i.$$

The constant $1/\alpha$ in Definition 20 is used to ensure that we always have $\mathbf{E}_H|\mathcal{X}|^{-1}\boldsymbol{\Sigma}_H(\mathcal{X}) = \mathbf{z}_{\mathcal{X}}$ when H is drawn according to $p_{\mathbf{h}}^n$ for some $p_{\mathbf{h}} \in \mathcal{P}_\alpha$. We will see below that, although subsampling reduces the noise level allowing to make the (unnormalized) sum of features private (which is sometimes referred to as "privacy amplification by subsampling"), the rescaling factor $1/\alpha$ is required to obtain sketches of comparable utility for a given noise level, and privacy is *not* amplified once taking this factor into account. We now introduce the whole mechanism, where the masks themselves are drawn randomly.

Definition 21. *The Laplacian subsampled sum of features $\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ of a dataset $\mathcal{X} \in \mathcal{D}_n$ using a mask distribution $p_{\mathbf{h}} \in \mathcal{P}_\alpha$ and a noise parameter b is the random variable*

$$\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}(\mathcal{X}) = \boldsymbol{\Sigma}_H(\mathcal{X}) + \boldsymbol{\xi}, \tag{13}$$

where $\forall j \in [\![1, m]\!]$, $\xi_j \overset{\text{iid}}{\sim} \begin{cases} \mathcal{L}^{\mathbb{C}}(b) & \text{if } \boldsymbol{\Phi} \text{ is complex-valued} \\ \mathcal{L}(b) & \text{if } \boldsymbol{\Phi} \text{ is real-valued} \end{cases}$, *and* $H = (\mathbf{h}_1, ..., \mathbf{h}_n)$ *with* $\mathbf{h}_i \overset{\text{iid}}{\sim} p_{\mathbf{h}}$.

For a deterministic set of masks H, we denote $\boldsymbol{\Sigma}_{\mathcal{L},H}(\mathcal{X}) = \boldsymbol{\Sigma}_H(\mathcal{X}) + \boldsymbol{\xi}$ the sum that is randomized only on $\boldsymbol{\xi}$. Compared to the deterministic sum of features $\boldsymbol{\Sigma}$, the Laplacian subsampled sum of features $\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}$ thus picks at random some values from each feature vector $\boldsymbol{\Phi}(\mathbf{x}_i)$ according to a random mask $\mathbf{h}_i$ and then adds Laplacian noise $\boldsymbol{\xi}$ on the sum of those contributions. Note that in this mechanism (and by opposition to $\boldsymbol{\Sigma}_{\mathcal{L},H}$), both $\boldsymbol{\xi}$ and H are random quantities.

In order to formulate our results, we define a quantity Q_1^{U} which is similar to the quantity from Lemma 5 but takes into account a mask $\mathbf{h} \in \mathcal{H}$. Although we only consider Q_1^{U} for the moment, we also introduce the quantities $Q_1^{\text{B}}, Q_2^{\text{U}}, Q_2^{\text{B}}$ which will be used in Section 5.3 for generalizing some results to the BDP and/or approximate DP settings. For a real-valued feature map and $p \in \{1, 2\}$, we define

$$Q_p^{\text{U}}(\mathbf{x}, \mathbf{h}) \triangleq \frac{1}{\alpha}\|\boldsymbol{\Phi}(\mathbf{x}) \odot \mathbf{h}\|_p \tag{14}$$

$$Q_p^{\text{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) \triangleq \frac{1}{\alpha}\|(\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\mathbf{y})) \odot \mathbf{h}\|_p. \tag{15}$$

The definition extends to complex-valued feature maps using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$, but using the same mask $\mathbf{h}$ for both real and imaginary parts.

Similarly to Section 4 where the privacy was directly driven by the quantity $\Delta_1^{\text{U}}(\boldsymbol{\Sigma})$, itself equal to $\sup_{\mathbf{x} \in E} Q_1^{\text{U}}(\mathbf{x})$, the following lemma gives a generalization taking the masks into account.

Lemma 15. *The Laplacian subsampled sum $\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ from Definition 21 with noise level b is UDP with sharp privacy parameter ε^*, defined as*

$$\exp(\varepsilon^*) = \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\text{U}}(\mathbf{x}, \mathbf{h})\right). \tag{16}$$

The proof can be found in Appendix C.1. For the standard Poisson data-subsampling mechanism, Equation (16) can be rewritten

$$\begin{aligned} \exp(\varepsilon^*) &= \sup_{\mathbf{x} \in E} \left\{(1 - \alpha) \cdot 1 + \alpha \cdot \exp\left(\frac{1}{b}\frac{1}{\alpha}\|\boldsymbol{\Phi}(\mathbf{x})\|_1\right)\right\} \\ &= 1 + \alpha\left(\exp\left(\frac{1}{\alpha}\frac{\Delta_1^{\text{U}}(\boldsymbol{\Sigma})}{b}\right) - 1\right). \end{aligned} \tag{17}$$

We thus recover the known bound of Lemma 14, however with the additional guarantee in our case that the bound is sharp. Indeed, according to this lemma if M is a random mechanism and $\mathcal{S}$ denotes the Poisson data-subsampling mechanism with parameter α, then the mechanism $\mathcal{X} \mapsto M(\mathcal{S}(\mathcal{X}))$ is ε'-UDP with $\varepsilon' = \log(1 + \alpha(\exp(\varepsilon) - 1))$. Applying this result to the mechanism $M : \mathcal{X} \mapsto \alpha^{-1}\boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi}$ which, by Theorem 1, is ε-UDP with $\varepsilon = \Delta_1^{\text{U}}(\alpha^{-1}\boldsymbol{\Sigma})b^{-1} = \alpha^{-1}\Delta_1^{\text{U}}(\boldsymbol{\Sigma})b^{-1}$ when $\boldsymbol{\xi}$ has iid (complex) Laplace components with parameter b, yields that the mechanism $\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ is ε'-UDP with $\exp(\varepsilon') = 1 + \alpha(\exp(\varepsilon) - 1) = 1 + \alpha\big(\exp\big(\alpha^{-1}\Delta_1^{\text{U}}(\boldsymbol{\Sigma})b^{-1}\big) - 1\big) = \exp(\varepsilon^*)$.

Lemma 15 allows us to show that subsampling cannot improve differential privacy guarantees.

Lemma 16. *If the Laplacian subsampled sum $\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ from Definition 21 is ε-UDP, then the noisy sum $\mathbf{\Sigma}_{\mathcal{L}}(\mathcal{X})$ computed with the same feature map and the same noise parameter (but without subsampling) is ε-UDP as well.*

Before we prove Lemma 16, let us just mention that for specific feature maps discussed later, the Laplacian subsampled sum $\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ is in fact just as differentially private as the one computed without subsampling (i.e. $\mathbf{\Sigma}_{\mathcal{L}}(\mathcal{X})$), while offering flexible tradeoffs between computational complexity and utility.

Furthermore, note that Lemma 16 also applies to standard Poisson data-subsampling. Hence, the idea that privacy might be "amplified" by subsampling should be mitigated. While the required noise level in order to make the sum of features private is indeed smaller when subsampling, the plain subsampled sum of features still needs to be rescaled by α^{-1} to obtain a sketch whose utility is comparable with that of the sketch computed using all samples. Overall, with both subsampling strategies (on the samples or on the features), one can at best obtain the same guarantees as when no subsampling is used.

Proof. Recall the definitions of $Q_1^{\mathrm{U}}(\mathbf{x})$ and $Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})$ given respectively in Lemma 5 and Equation (14). Using Jensen's inequality and the fact that the masks are drawn according to some $p_{\mathbf{h}} \in \mathcal{P}_\alpha$, we have for any $\mathbf{x}$ the lower bound

$$\mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right) \geq \exp\left(\frac{1}{b} \mathbf{E}_{\mathbf{h}} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right) = \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x})\right)$$

According to Lemma 15, taking the supremum on $\mathbf{x}$, we get

$$\begin{aligned}\sup_{\mathcal{X}, \mathcal{Y} \in \mathcal{D} : \mathcal{X} \overset{\mathrm{U}}{\sim} \mathcal{Y}} \sup_{\mathbf{s} \in \mathcal{Z}} \frac{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})}(\mathbf{s})}{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{Y})}(\mathbf{s})} &= \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right) \geq \sup_{\mathbf{x} \in E} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x})\right) \\ &= \exp\left(\frac{1}{b} \sup_{\mathbf{x} \in E} Q_1^{\mathrm{U}}(\mathbf{x})\right) = \exp\left(\frac{1}{b} \Delta_1^{\mathrm{U}}(\mathbf{\Sigma})\right)\end{aligned}$$

where the last equality comes from Lemma 5. If $\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})$ is ε-DP, we thus have

$$\exp(\frac{1}{b} \Delta_1^{\mathrm{U}}(\mathbf{\Sigma})) \leq \sup_{\mathcal{X}, \mathcal{Y} \in \mathcal{D} : \mathcal{X} \overset{\mathrm{U}}{\sim} \mathcal{Y}} \sup_{\mathbf{s} \in \mathcal{Z}} \frac{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})}(\mathbf{s})}{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{Y})}(\mathbf{s})} \leq \exp(\varepsilon)$$

which means $b \geq \Delta_1^{\mathrm{U}}(\mathbf{\Sigma})/\varepsilon$, hence by Theorem 1, $\mathbf{\Sigma}_{\mathcal{L}}(\mathcal{X})$ is ε-DP. □

In the following, we denote $\bar{\mathbf{\Sigma}}_H^{\mathrm{RFF}}$ and $\bar{\mathbf{\Sigma}}_H^{\mathrm{RQF}}$ the sums of subsampled features when using respectively $\mathbf{\Phi} = \mathbf{\Phi}^{\mathrm{RFF}}$ or $\mathbf{\Phi} = \mathbf{\Phi}^{\mathrm{RQF}}$ as a feature map. We now provide specific results for these two feature maps.

5.1 Random Fourier Features

The following lemma generalizes the notion of sensitivity to the subsampled case. We include the BDP case which will be used in Section 5.3.

Lemma 17. *Consider $\mathbf{\Phi}^{\mathrm{RFF}}$ built using nonresonant frequencies, and $r \in [\![1, m]\!]$. For each $\mathbf{h} \in \mathcal{H}_r$ we have*

$$\begin{aligned}\sup_{\mathbf{x} \in \mathbb{R}^d} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h}) &= \sup_{\mathbf{x} \in \mathbb{R}^d} \inf_{\mathbf{h}' \in \mathcal{H}_r} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h}') = \sqrt{2} m. \\ \sup_{\mathbf{x}, \mathbf{y} \in \mathbb{R}^d} Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) &= \sup_{\mathbf{x}, \mathbf{y} \in \mathbb{R}^d} \inf_{\mathbf{h}' \in \mathcal{H}_r} Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}') = 2\sqrt{2} m.\end{aligned}$$

Moreover $Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h}) \leq \sqrt{2} m$ and $Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) \leq 2\sqrt{2} m$ always hold, even for resonant frequencies.

The proof is quite similar to the proof of Lemma 7, and can be found in Appendix C.

We can now state the main result for random Fourier features.

Lemma 18. *Consider $r \in [\![1, m]\!]$, $\alpha = r/m$, and a probability distribution $p_{\mathbf{h}} \in \mathcal{P}_\alpha$ such that $\mathbf{h} \in \mathcal{H}_r$ almost surely. Then for any $\varepsilon > 0$, $\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RFF}}(\mathcal{X})$ from Definition 21 with noise level $b = \sqrt{2} m/\varepsilon$ and mask distribution $p_{\mathbf{h}}$ is ε-UDP. The bound is sharp if $\mathbf{\Phi}^{\mathrm{RFF}}$ is built using nonresonant frequencies.*

Proof. By Lemma 15 and Lemma 17 we have

$$\sup_{\mathcal{X},\mathcal{Y}\in\mathcal{D}:\mathcal{X}\overset{\mathrm{U}}{\approx}\mathcal{Y}} \sup_{\mathbf{s}\in\mathcal{Z}} \frac{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RFF}}(\mathcal{X})}(\mathbf{s})}{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RFF}}(\mathcal{Y})}(\mathbf{s})} = \sup_{\mathbf{x}\in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})\right)$$
$$= \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} \sup_{\mathbf{x}\in E} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})\right) = \exp\left(\frac{1}{b} m\sqrt{2}\right) = \exp(\varepsilon).$$

The second and third equalities are consequences of Lemma 17, and hold because $\mathbf{h}$ belongs to $\mathcal{H}_r$ almost surely. □

5.2 Random Quadratic Features

We recall that for random quadratic features, $E = \mathcal{B}_2 = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$, $\mathcal{D}_n \triangleq (\mathcal{B}_2)^n$, and $\mathcal{D} \triangleq \cup_{n\in\mathbb{N}} \mathcal{D}_n$. We give a generic upper bound in Lemma 20, and below in Lemma 22 a sharp bound when Ω is a union of orthonormal bases. We first provide a simple lemma used in both results. For any mask $\mathbf{h} \in \mathcal{H}_r$ with r non-zero entries at indexes $i_1, ..., i_r$, and any matrix of frequencies $\Omega = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_m] \in \mathbb{R}^{d\times m}$, we denote $\Omega_{\mathbf{h}} = [\boldsymbol{\omega}_{i_1}, ..., \boldsymbol{\omega}_{i_r}]$ the matrix obtained from Ω by keeping only the columns corresponding to nonzero indexes of $\mathbf{h}$.

Lemma 19. *Consider the functions $Q_1^{\mathrm{U}}, Q_1^{\mathrm{B}}$ associated to the feature map $\mathbf{\Phi}^{\mathrm{RQF}}$. For each $\mathbf{h} \in \mathcal{H}$*

$$\sup_{\mathbf{x}\in E} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}) = \sup_{\mathbf{x},\mathbf{y}\in E} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}) = \frac{1}{\alpha}\|\Omega_{\mathbf{h}}\|_2^2.$$

Proof.

$$\sup_{\mathbf{x},\mathbf{y}\in E} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}) = \sup_{\mathbf{x},\mathbf{y}\in E} \frac{1}{\alpha} \sum_{j\in\mathrm{supp}(\mathbf{h})} |(\boldsymbol{\omega}_j^T x)^2 - (\boldsymbol{\omega}_j^T y)^2|$$
$$= \sup_{\mathbf{x}\in E} \frac{1}{\alpha} \sum_{j\in\mathrm{supp}(\mathbf{h})} (\boldsymbol{\omega}_j^T x)^2 = \sup_{\mathbf{x}\in E} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})$$
$$= \sup_{\mathbf{x}\in E} \frac{1}{\alpha} \mathbf{x}^T \left(\sum_{j\in\mathrm{supp}(\mathbf{h})} \boldsymbol{\omega}_j \boldsymbol{\omega}_j^T\right) \mathbf{x} = \frac{1}{\alpha} \lambda_{\max}\left(\sum_{j\in\mathrm{supp}(\mathbf{h})} \boldsymbol{\omega}_j \boldsymbol{\omega}_j^T\right) = \frac{1}{\alpha}\|\Omega_{\mathbf{h}}\|_2^2.$$

□

For any $p_{\mathbf{h}} \in \mathcal{P}_\alpha$, we denote $\mathrm{supp}(p_{\mathbf{h}})$ the support of $p_{\mathbf{h}}$, that is the set of possible outcomes of $\mathbf{h} \sim p_{\mathbf{h}}$.

Lemma 20. *Let $p_{\mathbf{h}} \in \mathcal{P}_\alpha$. For any $\varepsilon > 0$, releasing $\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RQF}}(\mathcal{X})$ from Definition 21 with noise parameter $b = \frac{m}{r\varepsilon} \sup_{\mathbf{h}\in\mathrm{supp}(p_{\mathbf{h}})} \|\Omega_{\mathbf{h}}\|_2^2$ and mask distribution $p_{\mathbf{h}}$ is ε-UDP.*

Proof. By Lemma 15, with $E = \mathcal{B}_2$

$$\sup_{\mathcal{X},\mathcal{Y}\in\mathcal{D}:\mathcal{X}\overset{\mathrm{U}}{\approx}\mathcal{Y}} \sup_{\mathbf{s}\in\mathcal{Z}} \frac{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RQF}}(\mathcal{X})}(\mathbf{s})}{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\mathrm{RQF}}(\mathcal{Y})}(\mathbf{s})} = \sup_{\mathbf{x}\in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})\right) \leq \sup_{\mathbf{x}\in E} \sup_{\mathbf{h}\in\mathrm{supp}(p_{\mathbf{h}})} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})\right)$$
$$= \exp\left(\frac{1}{b}\frac{1}{\alpha} \sup_{\mathbf{h}\in\mathrm{supp}(p_{\mathbf{h}})} \|\Omega_{\mathbf{h}}\|_2^2\right),$$

by Lemma 19, which concludes the proof. □

Whether or when the bound of Lemma 20 is sharp in general is an open question. The finer bound $\mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b}\frac{1}{\alpha}\|\Omega_{\mathbf{h}}\|_2^2\right)$ holds, but does not yield explicit guarantees. A sharp and explicit bound can be achieved in a specific case of interest.

Lemma 21. *Consider m a multiple of d and Ω a concatenation of m/d orthonormal bases as described in Section 2.2. Let r be a multiple of d, and $\mathbf{h} \in \mathcal{H}_r^{\mathrm{struct.}}$. Then for any $\mathbf{x}$ such that $\|\mathbf{x}\|_2 = 1$, we have*

$$Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}) = \sup_{\mathbf{x}\in E} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}) = \sup_{\mathbf{x},\mathbf{y}\in E} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}) = \frac{m}{d}.$$

Proof. Let us rewrite $\Omega = [B_1, ..., B_{m/d}]$ where the $(B_i)_{1\leq i\leq m/d}$ are $d \times d$ blocs corresponding to orthonormal bases. We have $\Omega\Omega^T = \sum_{i=1}^{m/d} B_i B_i^T = m/d\, \mathbf{I}_d$. As $\mathbf{h} \in \mathcal{H}_r^{\text{struct.}}$, we have for the same reason $\Omega_{\mathbf{h}}\Omega_{\mathbf{h}}^T = (r/d)\, \mathbf{I}_d$. As a result, for any $\mathbf{x} \in E$ we have $Q_1^{\text{U}}(\mathbf{x}, \mathbf{h}) = \frac{1}{\alpha}(r/d)\|\mathbf{x}\|_2^2 = (m/d)\|\mathbf{x}\|_2^2$ and the result follows from $E = \mathcal{B}_2$. Given that $\mathbf{\Phi}^{\text{RFF}}$ takes only positive values and vanishes in 0, we have $\sup_{\mathbf{x},\mathbf{y}\in E} Q_1^{\text{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sup_{\mathbf{x},\mathbf{y}\in E} \|\mathbf{\Phi}^{\text{RFF}}(\mathbf{x}) - \mathbf{\Phi}^{\text{RFF}}(\mathbf{y})\|_1 = \sup_{\mathbf{x}\in E} \|\mathbf{\Phi}^{\text{RFF}}(\mathbf{x})\|_1 = Q_1^{\text{U}}(\mathbf{x}, \mathbf{h})$. □

Lemma 22. *Consider m a multiple of d and Ω a concatenation of m/d orthonormal bases as described in Section 2.2. Let r be a multiple of d, and $p_{\mathbf{h}} = \mathcal{U}(\mathcal{H}_r^{\text{struct.}})$ be the block-uniform distribution. For any $\varepsilon > 0$, releasing $\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\text{RQF}}(\mathcal{X})$ with mask distribution $p_{\mathbf{h}}$ and noise parameter $b = m/(d\varepsilon)$ is ε-UDP, and the bound is sharp.*

Proof. By Lemma 15 and Lemma 21, it follows that

$$\sup_{\mathcal{X},\mathcal{Y}\in\mathcal{D}:\mathcal{X}\overset{\text{U}}{\sim}\mathcal{Y}} \sup_{\mathbf{s}\in\mathcal{Z}} \frac{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\text{RQF}}(\mathcal{X})}(\mathbf{s})}{p_{\bar{\mathbf{\Sigma}}_{\mathcal{L}}^{\text{RQF}}(\mathcal{Y})}(\mathbf{s})} = \sup_{\mathbf{x}\in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\text{U}}(\mathbf{x}, \mathbf{h})\right) = \sup_{\mathbf{x}\in E} \exp\left(\frac{1}{b}\frac{m}{d}\|\mathbf{x}\|_2^2\right) = \exp\left(\frac{m}{db}\right) = \exp(\varepsilon),$$

where the second equality comes from Lemma 21 as any $\mathbf{x}$ for which $\|\mathbf{x}\|^2 = 2$ reaches the supremum for all $\mathbf{h} \in \mathcal{H}_r^{\text{struct.}}$ simultaneously. □

Note that the noise level required to get differential privacy when Ω is a union of orthonormal bases is independent of r and is the same as when $r = m$, i.e. without subsampling.

5.3 An Upper Bound for Approximate and Bounded Differential Privacy

Similarly to Definition 21, we define the Gaussian subsampled sum of features.

Definition 22. *The Gaussian subsampled sum of features $\bar{\mathbf{\Sigma}}_{\mathcal{G}}(\mathcal{X})$ of a dataset $\mathcal{X} \in \mathcal{D}_n$ using a mask distribution $p_{\mathbf{h}} \in \mathcal{P}_\alpha$ and a noise parameter σ is the random variable*

$$\bar{\mathbf{\Sigma}}_{\mathcal{G}}(\mathcal{X}) = \mathbf{\Sigma}_H(\mathcal{X}) + \boldsymbol{\xi}, \tag{18}$$

where $\forall j \in [\![1, m]\!]$, $\xi_j \overset{\text{iid}}{\sim} \begin{cases} \mathcal{N}^{\mathbb{C}}(0, \sigma^2) \text{ if } \mathbf{\Phi} \text{ is complex-valued} \\ \mathcal{N}(0, \sigma^2) \text{ if } \mathbf{\Phi} \text{ is real-valued} \end{cases}$, and $H = (\mathbf{h}_1, ..., \mathbf{h}_n)$ with $\mathbf{h}_i \overset{\text{iid}}{\sim} p_{\mathbf{h}}$.

Although we do not have an equivalent of Lemma 15 for approximate DP, we provide in Lemma 23 a generic upper bound, which holds for both pure and approximate DP, bounded and unbounded DP. In order to do so, we introduce the following definitions for $p \in \{1, 2\}$ and $\mathbf{h} \in \mathcal{H}$

$$Q_p^{\text{U}}(\mathbf{h}) \triangleq \sup_{\mathbf{x}\in E} Q_p^{\text{U}}(\mathbf{x}, \mathbf{h})$$
$$Q_p^{\text{B}}(\mathbf{h}) \triangleq \sup_{\mathbf{x},\mathbf{y}\in E} Q_p^{\text{U}}(\mathbf{x}, \mathbf{y}, \mathbf{h}).$$

Lemma 23. *Let $p_{\mathbf{h}} \in \mathcal{P}_\alpha$ be a mask distribution.*

- *For any $\varepsilon > 0$, the mechanism $\bar{\mathbf{\Sigma}}_{\mathcal{L}}$ from Definition 21 with mask distribution $p_{\mathbf{h}}$ and noise level $b \geq \max_{\mathbf{h}\in\text{supp}(p_{\mathbf{h}})} Q_1(\mathbf{h})/\varepsilon$ is ε-DP.*
- *For any $\varepsilon, \delta > 0$, the mechanism $\bar{\mathbf{\Sigma}}_{\mathcal{G}}$ from Definition 22 with mask distribution $p_{\mathbf{h}}$ and noise level $\sigma \geq \eta(\varepsilon, \delta) \max_{\mathbf{h}\in\text{supp}(p_{\mathbf{h}})} Q_2(\mathbf{h})/(2\varepsilon)^{1/2}$ (where $\eta(\varepsilon, \delta)$ refers to Theorem 2) is (ε, δ)-DP.*

These hold for both BDP and UDP, with $Q_p(\mathbf{h})$ defined accordingly as $Q_p^{\text{B}}(\mathbf{h})$ or $Q_p^{\text{U}}(\mathbf{h})$.

Proof. Let $\varepsilon > 0$, $R \in \{\bar{\mathbf{\Sigma}}_{\mathcal{L}}, \bar{\mathbf{\Sigma}}_{\mathcal{G}}\}$ be one of the two random mechanisms, and R_H for any H be the associated mechanism that uses the fixed masks H but is randomized on $\boldsymbol{\xi}$. Let $\sim \in \{\overset{\text{U}}{\sim}, \overset{\text{B}}{\sim}\}$ denote the considered neighborhood relation, and δ be such that $\delta = 0$ for pure DP, $\delta > 0$ for approximate DP. We need to show that

$$\forall \mathcal{X} \sim \mathcal{Y} \in \mathcal{D}, \mathbf{s} \in \mathcal{Z} : p_{R(\mathcal{X})}(\mathbf{s}) \leq \exp(\varepsilon) p_{R(\mathcal{Y})}(\mathbf{s}) + \delta$$

Fix $n > 0$ and an arbitrary set of masks $H = (\mathbf{h}_1, ..., \mathbf{h}_n) \in \mathcal{H}^n$, and consider the mechanism $\mathbf{\Sigma}_H$ on $\mathcal{D}' \triangleq \mathcal{D}_n$ (BDP case) or $\mathcal{D}' \triangleq \mathcal{D}_n \cup \mathcal{D}_{n-1}$ (UDP case; note that the expression of $\mathbf{\Sigma}_H(\mathcal{X})$ does not involve the last mask $\mathbf{h}_n$ when $|\mathcal{X}| = n - 1$ in this case) given in Definition 20. For a neighboring relation $\approx$, let $\Delta_{p,\approx}$ denote the L^p sensitivity computed according to $\approx$. For any ordered neighboring relation $\approx \in \{\overset{\text{U}}{\approx}, \overset{\text{B}}{\approx}\}$, according to Theorem 1 for pure DP and Theorem 2 for ADP applied on $\mathcal{D}'$ and w.r.t. $\approx$, if the noise level of $\boldsymbol{\xi}$ in R_H is chosen as $b \geq b^*_H \triangleq \Delta_{1,\approx}(\mathbf{\Sigma}_H)/\varepsilon$ or $\sigma \geq \sigma^*_H \triangleq \eta(\varepsilon, \delta)\Delta_{2,\approx}(\mathbf{\Sigma}_H)/(2\varepsilon)^{1/2}$, then we have for any $\mathcal{X}, \mathcal{Y} \in \mathcal{D}'$ such that $\mathcal{X} \approx \mathcal{Y}$

$$\forall \mathbf{s} \in \mathcal{Z} : p_{R_H(\mathcal{X})}(\mathbf{s}) \leq \exp(\varepsilon) p_{R_H(\mathcal{Y})}(\mathbf{s}) + \delta, \tag{19}$$

Note that the sensitivities depend on the neighboring relation used (UDP/BDP), but are always computed for an ordered relation, thus for $p \in \{1, 2\}$, we have $\Delta_{p,\approx}(\mathbf{\Sigma}_H) = Q_p(\mathbf{h}_n)$ if $H = (\mathbf{h}_1, ..., \mathbf{h}_n)$. The result follows by taking the expectation of these inequalities, which hold simultaneously for all H provided that $b \geq \max_{H \in \text{supp}(p_H)} b^*_H = \max_{\mathbf{h} \in \text{supp}(p_\mathbf{h})} Q_1(\mathbf{h})/\varepsilon$ (resp. that $\sigma \geq \max_{H \in \text{supp}(p_H)} \sigma^*_H = \eta(\varepsilon, \delta) \max_{\mathbf{h} \in \text{supp}(p_\mathbf{h})} Q_2(\mathbf{h})/(2\varepsilon)^{1/2}$).

The masks are drawn i.i.d. according to $p_\mathbf{h}$, and for any pair $(\mathcal{X}, H)$ we have $\mathbf{\Sigma}_H(\sigma(\mathcal{X})) = \mathbf{\Sigma}_{\sigma^{-1}(H)}(\mathcal{X})$, thus for any dataset $\mathcal{X}$ and permutation $\sigma \in \mathcal{S}_{|\mathcal{X}|}$ we have

$$p_{R(\mathcal{X})}(\mathbf{s}) = \mathbf{E}_H\Big[p_{R_H(\mathcal{X})}(\mathbf{s})\Big] = \mathbf{E}_H\Big[p_{R_{\sigma^{-1}(H)}(\mathcal{X})}(\mathbf{s})\Big] = \mathbf{E}_H\Big[p_{R_H(\sigma(\mathcal{X}))}(\mathbf{s})\Big]$$

If $\mathcal{X}$ and $\mathcal{Y}$ are two datasets such that $\mathcal{X} \sim \mathcal{Y}$ (we assume for now $|\mathcal{X}| \geq |\mathcal{Y}|$), then there are two permutations $\sigma_1 \in \mathcal{S}_{|\mathcal{X}|}, \sigma_2 \in \mathcal{S}_{|\mathcal{Y}|}$ such that $\sigma_1(\mathcal{X}) \approx \sigma_2(\mathcal{Y})$ for the related ordered relation (it follows from the definition for BDP, and one can take one permutation to be the identity for UDP).

Thus using the appropriate noise level according to Equation (19) we have

$$\begin{aligned}
&\forall \mathcal{X} \sim \mathcal{Y}, \mathbf{s} \in \mathcal{Z} : \mathbf{E}_H[p_{R_H(\sigma_1(\mathcal{X}))}(\mathbf{s})] \leq \exp(\varepsilon)\mathbf{E}_H[p_{R_H(\sigma_2(\mathcal{Y}))}(\mathbf{s})] + \delta \\
\text{i.e. } &\forall \mathcal{X} \sim \mathcal{Y}, \mathbf{s} \in \mathcal{Z} : p_{R(\sigma_1(\mathcal{X}))}(\mathbf{s}) \leq \exp(\varepsilon) p_{R(\sigma_2(\mathcal{Y}))}(\mathbf{s}) + \delta \\
\text{i.e. } &\forall \mathcal{X} \sim \mathcal{Y}, \mathbf{s} \in \mathcal{Z} : p_{R(\mathcal{X})}(\mathbf{s}) \leq \exp(\varepsilon) p_{R(\mathcal{Y})}(\mathbf{s}) + \delta,
\end{aligned}$$

which is the desired result.

Note that $\overset{\text{U}}{\approx}$ is not a symmetric relation, but in the UDP case with $|\mathcal{Y}| = |\mathcal{X}|+1$, we can still find σ_1, σ_2 such that $\sigma_1(\mathcal{Y}) \overset{\text{U}}{\approx} \sigma_2(\mathcal{X})$. We hence obtain the desired result by deriving an equivalent of Equation (19) for the relation $\overset{\text{U}}{\approx}_s$, defined as $\mathcal{X} \overset{\text{U}}{\approx}_s \mathcal{Y} \Leftrightarrow \mathcal{Y} \overset{\text{U}}{\approx} \mathcal{X}$. As for any H, we have $\Delta_{p,\overset{\text{U}}{\approx}}(\mathbf{\Sigma}_H) = \Delta_{p,\overset{\text{U}}{\approx}_s}(\mathbf{\Sigma}_H)$ on $\mathcal{D}'$, we get the same result. □

Whether or nor the bounds from Lemma 23 are sharp for certain scenarios is a question left open for future work.

From Lemma 23, one can get guarantees for (ε, δ)-DP with the two simple following results.

Lemma 24. *Let $\mathbf{h} \in \mathcal{H}_r$. Then for RFF we have $\sup_\mathbf{x} Q^{\text{U}}_2(\mathbf{x}, \mathbf{h}) = \frac{m}{\sqrt{r}}$.*

Proof.

$$\sup_\mathbf{x} Q^{\text{U}}_2(\mathbf{x}, \mathbf{h}) = \sup_{\mathbf{x} \in E} \frac{m}{r} \|\mathbf{\Phi}^{\text{RFF}}(\mathbf{x})\|_2 = \frac{m}{r}\sqrt{r} = \frac{m}{\sqrt{r}}.$$ □

Lemma 25. *Let $\mathbf{h} \in \mathcal{H}_r$. Then for RQF we have $\sup_{\mathbf{x},\mathbf{y} \in E} Q^{\text{B}}_2(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sup_{\mathbf{x} \in E} Q^{\text{U}}_2(\mathbf{x}, \mathbf{h}) = \frac{m}{r} S_4(\Omega_\mathbf{h})$.*

Proof.

$$\sup_\mathbf{x} Q^{\text{U}}_2(\mathbf{x}, \mathbf{h}) = \sup_{\mathbf{x} \in E} \frac{m}{r} \|\mathbf{\Phi}^{\text{RQF}}(\mathbf{x})\|_2 = \sup_{\mathbf{x} \in E} \frac{m}{r} \Big(\sum_{i \in \text{supp}(\mathbf{h})} (\boldsymbol{\omega}_i^T \mathbf{x})^4 \Big)^{1/2} = \frac{m}{r} S_4(\Omega_\mathbf{h})$$

As $\mathbf{\Phi}^{\text{RQF}}$ takes positive values and vanishes in $\mathbf{0}$, which belongs to E, the same bound holds for BDP. □

Note that in these two cases, subsampling increases the bounds and might have a negative impact on the utility (for subsequent learning) of the mechanism.

Summary We summarize the results obtained in this paper in the following tables, where $\eta = \eta(\varepsilon, \delta)$ refers to Theorem 2.

	Pure ε-DP		**Approximate (ε, δ)-DP**	
	$\boldsymbol{\Sigma}_{\mathcal{L}}(\mathcal{X}) = \boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi}$ with $\xi_j \sim \mathcal{L}(b)$, $b \geq b^*$		$\boldsymbol{\Sigma}_{\mathcal{G}}(\mathcal{X}) = \boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi}$ with $\xi_j \sim \mathcal{N}(0, \sigma^2)$, $\sigma \geq \sigma^*$	
	UDP	**BDP**	**UDP**	**BDP**
Generic	Theorem 1: $b^* = \Delta_1(\boldsymbol{\Sigma})/\varepsilon$		Theorem 2: $\sigma^* = \frac{\eta}{\sqrt{2\varepsilon}}\Delta_2(\boldsymbol{\Sigma})$	
	$\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}) = \sup_{\mathbf{x}} \lVert\boldsymbol{\Phi}(\mathbf{x})\rVert_1$	$\Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}) = \sup_{\mathbf{x},\mathbf{y}} \lVert\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\mathbf{y})\rVert_1$	$\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}) = \sup_{\mathbf{x}} \lVert\boldsymbol{\Phi}(\mathbf{x})\rVert_2$	$\Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}) = \sup_{\mathbf{x},\mathbf{y}} \lVert\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\mathbf{y})\rVert_2$
RFF	Lemma 7: $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) \leq \sqrt{2m}$	Lemma 17:[1] $\Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) \leq 2\sqrt{2m}$	Lemma 11: $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sqrt{m}$	Lemma 11:[2] $\Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) \leq 2\sqrt{m}$
+ Ω nonresonant	$\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sqrt{2m}$	$\Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = 2\sqrt{2m}$	$\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sqrt{m}$	Lemma 12: $\Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = 2\sqrt{m}$
RQF	Lemma 9: $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = \Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = \lVert\Omega\rVert_2^2$		Lemma 13: $\Delta_2^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = \Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = S_4(\Omega)$	
+ Ω union of orthogonal bases.	Lemma 9: $\Delta_1^{\mathrm{U}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = \Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RQF}}) = m/d$		No particular closed form.	

Table 1: Summary of privacy results **without** subsampling (Section 4) and for the sum of features only. For each type of privacy guarantee (column) and for each sketch feature function (row), we provide a potentially loose ($\leq$) or sharp ($=$) bound on the relevant sensitivity Δ, which can be plugged into the associated privacy-preserving sum of features mechanism (top row). We use the notation $\eta = \eta(\varepsilon, \delta)$, which refers to Theorem 2. [1] With $h = \mathbf{1}$, i.e. $\Delta_1(\boldsymbol{\Sigma}^{\mathrm{RFF}}) = \sup_{\mathbf{x}} Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{1})$ where Q_1^{B} is computed with $r = m$. [2] Using a simple triangle inequality.

	Pure ε-DP		**Approximate (ε, δ)-DP**	
	$\bar{\boldsymbol{\Sigma}}_{\mathcal{L}}(\mathcal{X}) = \boldsymbol{\Sigma}_H(\mathcal{X}) + \boldsymbol{\xi}$ with $\xi_j \sim \mathcal{L}(b), H \sim p_{\mathbf{h}}^n$, $b \geq b^*$		$\bar{\boldsymbol{\Sigma}}_{\mathcal{G}}(\mathcal{X}) = \boldsymbol{\Sigma}_H(\mathcal{X}) + \boldsymbol{\xi}$ with $\xi_j \sim \mathcal{N}(0, \sigma^2), H \sim p_{\mathbf{h}}^n$, $\sigma \geq \sigma^*$	
	UDP	**BDP**	**UDP**	**BDP**
Generic	Lemma 15: $e^{\varepsilon^*} = \sup_{\mathbf{x}} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b^*} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right)$	Lemma 23: $b^* \leq \sup_H \Delta_1^{\mathrm{B}}(\boldsymbol{\Sigma}_H)/\epsilon$	Lemma 23: $\sigma^* \leq \frac{\eta}{\sqrt{2\epsilon}} \sup_{\mathbf{h}} Q_2^{\mathrm{U}}(\mathbf{h})$	Lemma 23: $\sigma^* \leq \frac{\eta}{\sqrt{2\epsilon}} \sup_{\mathbf{h}} Q_2^{\mathrm{B}}(\mathbf{h})$
RFF	Lemma 18: $b^* \leq \sqrt{2}m/\varepsilon$	Lemmas 17 and 23: $b^* \leq 2\sqrt{2}m/\epsilon$	Lemmas 23 and 24: $\sigma^* \leq \frac{\eta}{\sqrt{2\varepsilon}} \frac{m}{\sqrt{r}}$	Lemmas 23 and 24:[1] $\sigma^* \leq \frac{\eta}{\sqrt{2\varepsilon}} \frac{2m}{\sqrt{r}}$
+ Ω nonresonant	$b^* = \sqrt{2}m/\varepsilon$ (same lemma)	not covered	not covered	not covered
RQF	Lemma 20: $b^* \leq \frac{1}{\alpha}\frac{1}{\epsilon} \sup_{\mathbf{h}} \lVert\Omega_{\mathbf{h}}\rVert_2^2$	Lemmas 19 and 23: $b^* \leq \frac{1}{\alpha}\frac{1}{\epsilon} \sup_{\mathbf{h}} \lVert\Omega_{\mathbf{h}}\rVert_2^2$	Lemmas 23 and 25: $\sigma^* \leq \frac{\eta}{\sqrt{2\varepsilon}} \frac{m}{r} \sup_{\mathbf{h}} S_4(\Omega_{\mathbf{h}})$	
+ Ω union of orthogonal bases and $\mathbf{h} \sim \mathcal{U}(\mathcal{H}_r^{\text{struct.}})$.	Lemma 22: $b^* = m/(d\epsilon)$	Lemmas 21 and 23: $b^* \leq m/(d\epsilon)$	No particular closed form	

Table 2: Summary of privacy results **with** subsampling (Section 5). For each type of privacy guarantee (column) and for each sketch feature function (row), we provide a potentially loose ($\leq$) or sharp ($=$) bound on the required additive noise levels (b^* or σ^*). We denote $\eta = \eta(\varepsilon, \delta)$, which refers to Theorem 2, and α the subsampling parameter (and $r \triangleq \alpha m$ when relevant). [1] Using a simple triangle inequality.

6 Utility Guarantees under Differential Privacy

Having established the differential privacy properties of noisy sketching mechanisms, we conclude this paper by investigating the impact of different aspects of those mechanisms on their utility for subsequent learning (i.e. the quality of the models learned from noisy sketches, as measured by the metrics introduced in Definitions 2, 3 and 5).

More precisely, we derive a principled approach to tune various parameters of our mechanism (e.g. the subsampling strategy, the split of the privacy budget, the sketch size) *a priori*. Given a fixed target privacy level, several choices of parameters are indeed possible, that can each yield a different utility value. Our goal is to pick the best choice of parameters (or at least a promising one), without accessing the data, which would require to allocate a significant part of the privacy budget for parameter tuning.

We first establish, both from theoretical sketched learning guarantees as well as from numerical simulations, that a proxy comprising of a noise-to-signal ratio (NSR) and the sketch size m can qualitatively predict the utility (Section 6.1). The NSR is then computed analytically (Section 6.2) and used to tune some of the parameters of our method: the subsampling strategy (Sections 6.3 and 6.4), the splitting of the privacy budget (Section 6.5), and the choice of the sketch size (Section 6.6, where we exploit the combined influence of the NSR and m on the utility).

6.1 Noise to signal ratio as a proxy for utility

We recall (see the beginning of Section 2) that a learning task is defined by a risk function $\mathcal{R}$ and a domain $\mathcal{H}$ (also known as the hypothesis class), and the parameters one would like to learn are $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathcal{H}} \mathcal{R}(\pi_0, \boldsymbol{\theta})$, where π_0 is the true (unknown) distribution of the data. The goal is to estimate – in our case, from the noisy sketch only – a set of parameters $\hat{\boldsymbol{\theta}}$ such that the quantity $\mathcal{R}(\pi_0, \hat{\boldsymbol{\theta}}) - \mathcal{R}(\pi_0, \boldsymbol{\theta}^*)$, called an excess risk, can be controlled. Previous works [27, 10] showed that such a control can be achieved using proof techniques that leverage the analogy between sketching and compressive sensing. Indeed, although the feature map $\boldsymbol{\Phi}$ is non-linear, sketching is a linear operation w.r.t. distributions, and we denote $\mathcal{A}$ the associated operator defined as $\mathcal{A}(\pi) = \mathbf{E}_{x\sim\pi}\Phi(x)$. With this notation, the clean "true" sketch would be $\mathbf{z} = \mathcal{A}(\pi_0)$, and the clean empirical sketch can be denoted $\mathbf{z}_{\mathcal{X}} = \mathcal{A}(\pi_{\mathcal{X}})$. In practice, we observe a noisy version $\mathbf{s}(\mathcal{X})$ of the empirical sketch, which can for example be computed as the ratio $\mathbf{s}(\mathcal{X}) = (\boldsymbol{\Sigma}(\mathcal{X}) + \boldsymbol{\xi})/(|\mathcal{X}| + \zeta)$ with $\boldsymbol{\xi}$ being either Laplacian or Gaussian according to Definitions 17 and 19. As shown in [27, 10], for k-means clustering, Gaussian mixture modeling and PCA, learning from the noisy sketch $\mathbf{s}(\mathcal{X})$ can be expressed as solving a linear inverse problem on a certain parametric set of probability distributions. Under some assumptions on the sketching function $\boldsymbol{\Phi}$ and the learning task, the excess risk can be bounded by a quantity that involves a measure of noise level $\|\mathbf{e}\|_2$, with $\mathbf{e} \triangleq \mathbf{s}(\mathcal{X}) - \mathbf{z}$. As a proxy for the utility of a noisy sketch, we thus propose the noise-to-signal ratio (NSR), defined as

$$\mathrm{NSR} \triangleq \|\mathbf{s}(\mathcal{X}) - \mathbf{s}\|_2^2 / \|\mathbf{s}\|^2$$

w.r.t. some reference sketch $\mathbf{s}$, that will typically be the clean empirical sketch of $\mathcal{X}$, $\mathbf{z}_{\mathcal{X}}$, or the "true" sketch $\mathbf{z}$ of the assumed underlying distribution π_0.

The NSR was indeed shown empirically [45] to be a good proxy to estimate the utility of a sketching mechanism for the task of clustering where performance is measured with the SSE (sum of squared errors) defined in (2). Figures 4 and 5 give an overview of this correlation. On Figure 4, we plot the relative SSE (RSSE, i.e. the ratio between the SSE obtained with centroids determined from a sketch and the SSE obtained with centroids computed using Lloyd's algorithm) for data generated according to Gaussian mixtures with parameters $k = d = 10$, $m = 10kd$. The desired NSR is obtained by adding isotropic noise of controlled magnitude on the clean sketch computed without subsampling. In Figure 5, we plot the RSSE for $n = 10^4$ using different sketch sizes and NSRs, again obtained with isotropic noise and without subsampling. The red dashed line corresponds to $m = 2kd$, and as expected [36] the reconstruction fails below this line. From this plot, we derive that when $m \geq 2kd$, one can consider that the reconstruction is successful provided that $\mathrm{NSR} \leq m/(10^3 kd) \triangleq \mathrm{NSR}_{\max}(m)$ (yellow area). We thus propose to use NSR-minimization as a criterion to tune the parameters of our method, assuming the sketch size m is fixed (we discuss how to select this size in Section 6.6).

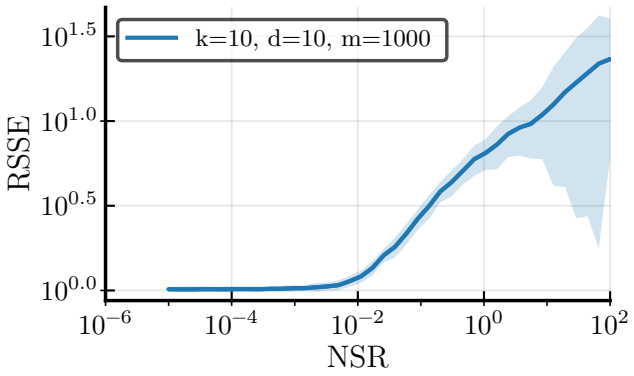

Figure 4: Correlation between relative (w.r.t. k-means with 3 trials) SSE and NSR. Medians of 100 trials and variance.

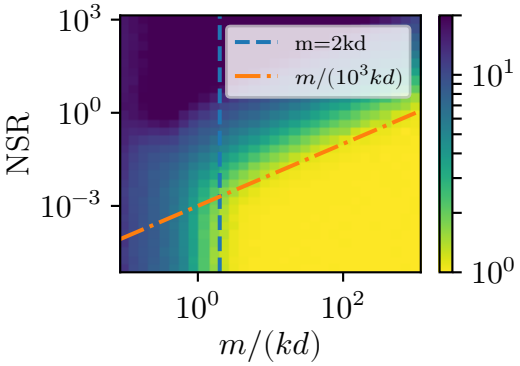

Figure 5: RSSE as a function of m/kd and NSR, using $n = 10^4$, $k = d = 10$, medians of 100 trials.

6.2 Analytical estimation of the noise level

We now compute in this section the expected noise level (and NSR) induced by the mechanisms introduced in the previous sections and possibly combined with a hybrid dataset subsampling mechanism (i.e., we consider subsampling at the same time features and data samples as explained below).

Let $\mathcal{X}$ be a fixed dataset. The noise level can be measured with respect to the "true" sketch $\mathbf{z}$ of the assumed underlying distribution π_0, or with respect to the clean empirical sketch $\mathbf{z}_\mathcal{X}$. In the first case, which is relevant to take into account the statistical significance due to the size n of the dataset, we define $\mathbf{e} \triangleq \mathbf{s}(\mathcal{X}) - \mathbf{z}$, and the noise level as $\mathbf{E}\|\mathbf{e}\|_2^2$, where the expectation is taken on both the randomness of the mechanism and on the draw of $\mathcal{X}$. We define the noise-to-signal ratio (NSR) as the noise level normalized by the signal energy, i.e. $\text{NSR} = \mathbf{E}\|\mathbf{e}\|_2^2 / \|\mathbf{z}\|_2^2$. When $\mathbf{z}_\mathcal{X}$ is chosen as the reference signal rather than $\mathbf{z}$, we have $\text{NSR} = \mathbf{E}\|\mathbf{s}(\mathcal{X}) - \mathbf{z}_\mathcal{X}\|_2^2 / \|\mathbf{z}_\mathcal{X}\|_2^2$, and the expectation is taken w.r.t. the randomness of the mechanism only.

Subsampling the dataset. Although we were mainly interested in subsampling the individual features $\mathbf{\Phi}(\mathbf{x}_i)$ when introducing our sampling mechanism in Section 5, we have seen that another straightforward way to reduce the computational complexity is to simply subsample the dataset. The sum of features combining both subsampling strategies is given as

$$\widetilde{\mathbf{\Sigma}}(\mathcal{X}) \triangleq \frac{1}{\beta}\left(\frac{1}{\alpha}\sum_{i=1}^{n} g_i \cdot \mathbf{\Phi}(\mathbf{x}_i) \odot \mathbf{h}_i + \boldsymbol{\xi}\right), \tag{20}$$

where the scalars $(g_i)_{1\leq i\leq n}$ are in $\{0,1\}$ and randomly drawn (i.i.d. Bernoulli with parameter β, or n' among n without replacement, in which case we define $\beta \triangleq n'/n$), the masks $(\mathbf{h}_i)_{1\leq i\leq n}$ are drawn as previously i.i.d. according to a distribution $p_\mathbf{h} \in \mathcal{P}_\alpha$, and the additive noise $\boldsymbol{\xi}$ is Laplacian or Gaussian. Note that, when the $(g_i)_{1\leq i\leq n}$ are drawn i.i.d. according to a Bernoulli distribution, then (20) can be seen as a special case of Definition 21 with $\mathbf{h}'_i = g_i\mathbf{h}_i$ and $\alpha' = \alpha\beta$. We here clearly dissociate the two sampling strategies, which allows us to consider sampling the data without replacement, but will also make it easier to separate the contributions to the NSR coming from the two sampling strategies. From now on, we consider an estimator $\mathbf{s}(\mathcal{X})$ of $\mathbf{z}_\mathcal{X}$ as a function of the sum of features $\widetilde{\mathbf{\Sigma}}(\mathcal{X})$ introduced in (20), which by an adequate choice of the parameters encompasses all the mechanisms previously defined.

Noise-to-signal ratio when n is public. When the dataset size n is assumed to be public (e.g. in a BDP setting), we can use the estimator $\mathbf{s}(\mathcal{X}) \triangleq \widetilde{\mathbf{\Sigma}}(\mathcal{X})/n$. The following result is proved in Appendix D.

Lemma 26. *The noise-to-signal ratio of the mechanism $\mathbf{s}(\mathcal{X}) = \widetilde{\mathbf{\Sigma}}(\mathcal{X})/n$ with additive noise of variance σ_ξ^2, features subsampling with parameter $\alpha \triangleq r/m$ and i.i.d. Poisson subsampling of the dataset samples*

with parameter $0 \leq \beta \leq 1$ *is*

$$\text{w.r.t. } \mathbf{z}\text{:} \quad \mathrm{NSR}_{\mathbf{z}} = \frac{1}{n}\left(\frac{1}{\alpha\beta}\frac{\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|^2}{\|\mathbf{z}\|^2} - 1\right) + \frac{m}{n^2\beta^2}\frac{\sigma_\xi^2}{\|\mathbf{z}\|^2}$$

$$\text{w.r.t. } \mathbf{z}_{\mathcal{X}}\text{:} \quad \mathrm{NSR}_{\mathbf{z}_{\mathcal{X}}} = \frac{1}{n}\left(\frac{1}{\alpha\beta} - 1\right)\left(\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{\Phi}(\mathbf{x}_i)\|^2\right)\frac{1}{\|\mathbf{z}_{\mathcal{X}}\|^2} + \frac{m}{n^2\beta^2}\frac{\sigma_\xi^2}{\|\mathbf{z}_{\mathcal{X}}\|^2}.$$

The expressions for sampling without replacement differ slightly and are given in the proof in Appendix D.

Noise-to-signal ratio when n is sensitive. When the dataset size is considered sensitive, noise ζ must be added on n for privacy as discussed earlier. Our estimator of the sketch can then be written $\mathbf{s}(\mathcal{X}) = \widetilde{\mathbf{\Sigma}}(\mathcal{X})f(|\mathcal{X}| + \zeta)$, where $f(|\mathcal{X}| + \zeta)$ is an estimator of $1/|\mathcal{X}|$. The noise-to-signal ratio is now defined as $\mathrm{NSR} \triangleq \mathbf{E}\|\widetilde{\mathbf{\Sigma}}(\mathcal{X})f(|\mathcal{X}| + \zeta) - \mathbf{s}\|_2^2/\|\mathbf{s}\|_2^2$, where $\mathbf{s}$ stands for the reference signal, which can again be either $\mathbf{z}$ or $\mathbf{z}_{\mathcal{X}}$. An analytic expression of this NSR is given in Appendix D which involves the bias and variance of the estimator of $1/|\mathcal{X}|$ defined by f. Considering an unbiased estimator, a Cramer-Rao lower bound leads to the following result which is proved in Appendix D.

Lemma 27. *When using an estimator f of $1/n$ computed from the quantity $n+\zeta$, where $\zeta \sim \mathcal{L}(0, \sigma_\zeta/\sqrt{2})$, a Cramer-Rao bound on the noise-to-signal ratio of the sketching mechanism is*

$$\mathrm{NSR}_\zeta \geq \left(1 + \frac{\sigma_\zeta^2}{2n^2}\right)(\mathrm{NSR} + 1) - 1,$$

where NSR *refers to the ratio obtained without f (i.e. when $\zeta = 0$) as computed in Lemma 26, and can be computed with respect to either $\mathbf{z}$ or $\mathbf{z}_{\mathcal{X}}$.*

6.3 Comparison of the two subsampling strategies

For a given dataset size n, the sketching cost scales in $\Theta(n\alpha\beta)$ when subsampling the sketches with $r = \alpha m$ observations (with $\alpha \leq 1$) and subsampling the dataset by using only $n' = \beta n$ samples. Hence for a given n, a constant product $\alpha\beta$ means a constant computational complexity. We now use Lemma 26 to show that, for an equivalent computational complexity and privacy, subsampling the sketches leads to a better NSR, and hence likely a better utility, than subsampling the dataset.

For Poisson data subsampling, the only term of the NSR (given in Lemma 26) that varies with (α, β) at constant complexity $\alpha\beta$ is the term coming from the additive noise:

$$\mathrm{NSR}_\xi \triangleq \frac{m}{n^2\beta^2}\frac{\sigma_\xi^2}{\|\mathbf{z}\|^2}. \tag{21}$$

Note that this holds as well when working with the Cramer-Rao bound from Lemma 27, as the term σ_ζ does not depend on α, β at all. To investigate how this varies we need to take into account that for a fixed target privacy ε, the variance σ_ξ^2 also depends on β.

Let us consider the ε-DP setting with random Fourier features as an illustration (a similar reasoning holds for random quadratic features). When the $(g_i)_{1\leq i\leq n}$ are i.i.d. Bernoulli random variables with parameter β, then the distribution of the $(g_i\mathbf{h}_i)_{1\leq i\leq n}$ is in $\mathcal{P}_{\alpha\beta}$ and thus according to Lemma 15, releasing $\widetilde{\mathbf{\Sigma}}$ as defined in (20) with noise $\boldsymbol{\xi}$ of parameter b_ξ (i.e. with total noise level b_ξ/β given that $\boldsymbol{\xi}$ is normalized by β^{-1} in (20)) is ε-DP with

$$\begin{aligned}
\exp(\varepsilon) &= \sup_{\mathbf{x}\in E} \mathbf{E}_g\mathbf{E}_{\mathbf{h}} \exp\left(\frac{\beta}{b_\xi}\frac{1}{\alpha\beta}\|\mathbf{\Phi}(\mathbf{x}) \odot (g\mathbf{h})\|_1\right) \\
&= \sup_{\mathbf{x}\in E}\left((1-\beta)\cdot 1 + \beta\cdot\mathbf{E}_{\mathbf{h}}\exp\left(\frac{1}{b_\xi}\frac{1}{\alpha}\|\mathbf{\Phi}(\mathbf{x}) \odot \mathbf{h}\|_1\right)\right) \\
&\overset{\text{(i)}}{=} (1-\beta) + \beta\cdot\exp(\varepsilon'), \text{ with } \varepsilon' = \frac{\sqrt{2}m}{b_\xi},
\end{aligned}$$

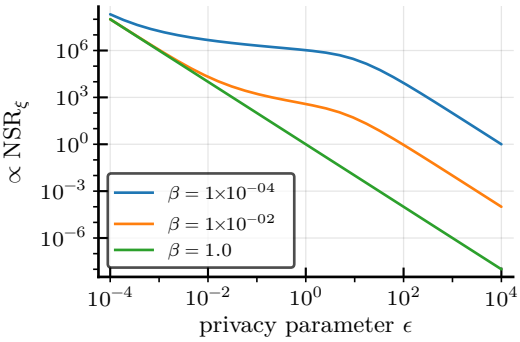

Figure 6: Variation of NSR_ξ as a function of ε for different values of the data subsampling parameter β. This quantity is the only variable term of the NSR (cf Lemma 26) at constant computational complexity, i.e. when the product $\alpha\beta$ is constant. Displayed using (23) with the convention $\frac{m^3}{n^2\|\mathbf{z}\|^2} = 1$ to fix a vertical scale.

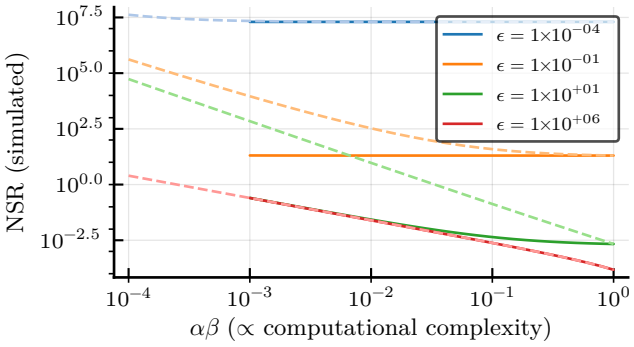

Figure 7: Total NSR *vs* total subsampling factor $\alpha\beta$. (plain) pure feature subsampling ($\alpha \in [1/m, 1], \beta = 1$); (dashed) pure data subsampling ($\alpha = 1, \beta \in [1/n, 1]$, Poisson sampling). All curves computed with analytic expressions of NSR, for $m = 10^3, n = 10^4$.

where (i) follows from Lemma 15 (applied on the mechanism which subsamples only the features and adds Laplacian noise at level b_ξ) and Lemma 18 (assuming non-resonant frequencies and distribution $p_{\mathbf{h}}$ such that $\mathbf{h} \in \mathcal{H}_r$ almost surely, where $r = \alpha m$). This is equivalent to saying that (20) is ε-DP provided

$$b_\xi = \frac{\sqrt{2}m}{\varepsilon'}, \text{ with } \varepsilon' = \log(1 + (\exp(\varepsilon) - 1)/\beta) \tag{22}$$

and that this noise level is sharp. The same bound could have been obtained by applying Lemma 14 on our mechanism which subsamples only the features, however here again Lemma 15 additionally proves that the bound is sharp.

Given that $\sigma_\xi^2 \propto b_\xi^2$, the resulting NSR term is then according to (21) and (22) of the order of

$$\mathrm{NSR}_\xi \propto \frac{m^3}{n^2\|\mathbf{z}\|^2}\frac{1}{\beta^2(\varepsilon')^2} = \frac{m^3}{n^2\|\mathbf{z}\|^2}\frac{1}{\beta^2\log^2(1 + (\exp(\varepsilon) - 1)/\beta)}. \tag{23}$$

In particular we have $\mathrm{NSR}_\xi \propto \frac{m^3}{n^2\|\mathbf{z}\|^2}\frac{1}{\varepsilon^2}$ when $\beta = 1$.

The behavior of this quantity as a function of the Poisson data subsampling rate $0 < \beta < 1$ (recall that we consider a constant overall subsampling rate $\alpha\beta$) depends on the considered privacy regime. As illustrated on Figure 6, for each of the three curves: a) when $\varepsilon \ll \beta$, we have $\varepsilon' \approx \varepsilon/\beta$ and $1/(\beta^2(\varepsilon')^2) \approx 1/\varepsilon^2$, thus NSR_ξ is of the same order for $\beta \ll 1$ and $\beta = 1$, hence the difference between the two subsampling schemes is negligible; b) if $\varepsilon \gg 1$ then $\varepsilon' \approx \varepsilon$ and NSR_ξ is thus increased by a factor $1/\beta^2$ when subsampling on n (i.e. when $\beta \ll 1$, and in comparison with the setting $\beta = 1$), which might be damageable in terms of utility. Put differently, in light of the expression (23), for each privacy parameter ε, the minimum value of $1/(\beta\varepsilon')^2$ – the quantity which drives NSR_ξ – is achieved at $\beta = 1$. The effect on the total NSR for the two sampling scenarios is shown in Figure 7 using the analytic expressions of Lemma 26 and Equation (23). This confirms that subsampling the features rather than the samples yields substantial NSR gains for moderate ε (i.e. neither too large nor too small). For large ε, the noise level is very low anyway, and no difference appears between the two scenarios. Additional experiments (not shown here) show that when sampling the dataset without replacement rather than with Poisson sampling, and measuring the NSR with $\mathbf{z}_\mathcal{X}$ as a reference signal, subsampling the features can become slightly disadvantaging for large values of ε, but the difference is very small.

6.4 Regimes combining privacy and utility

In this section, we try to highlight the regimes in which the sketches produced by our mechanism are still useful from a learning perspective. We do so by comparing the different contributions to the NSR.

In light of the results of Section 6.3, we focus on subsampling the features only (i.e. $\beta = 1$). In this setting, and when working with random Fourier features, since $\|\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x})\|^2 = m$ for every $\mathbf{x}$, the different contributions to the NSR (computed with $\mathbf{z}$ as reference) are of the order of

$$\mathrm{NSR}_{\mathcal{X}} \approx \frac{1}{n}(C_0 - 1), \qquad \mathrm{NSR}_{\xi} \approx \frac{C_0 m^2}{n^2 \varepsilon^2}, \qquad \mathrm{NSR}_{H} \approx \frac{C_0}{n}\left(\frac{1}{\alpha} - 1\right),$$

where $C_0 \triangleq m/\|\mathbf{z}\|^2$. Using the interpretation of $\|\mathbf{z}\|^2/m$ as an expected value [27] the quantity C_0 is essentially independent of m and satisfies $C_0 > 1$. In practice, empirical simulations on very different datasets suggest that one can safely assume $C_0 < 10$.

Acceptable noise level without subsampling. The total noise is acceptable when the sum of these contributions to the NSR is smaller than some threshold $\mathrm{NSR}_{\max}$, which depends on the sketch size m as seen on Figure 4. Necessary conditions read:

$$\mathrm{NSR}_{\mathcal{X}} \leq \mathrm{NSR}_{\max} \Leftrightarrow n \gtrsim \frac{1}{\mathrm{NSR}_{\max}} \qquad \mathrm{NSR}_{\xi} \leq \mathrm{NSR}_{\max} \Leftrightarrow n \gtrsim \frac{m}{\varepsilon\sqrt{\mathrm{NSR}_{\max}}} \tag{24}$$

Thus utility is preserved (and privacy achieved) when

$$n \gtrsim \max\left(\frac{1}{\mathrm{NSR}_{\max}}, \frac{m}{\varepsilon\sqrt{\mathrm{NSR}_{\max}}}\right). \tag{25}$$

Acceptable noise level with subsampling. The noise induced by feature subsampling is acceptable when $\mathrm{NSR}_H \lesssim \mathrm{NSR}_{\max}$, i.e., for subsampling with $\alpha = 1/m$, when $n \gtrsim m/\mathrm{NSR}_{\max}$. Combining this with the two conditions from the previous paragraph, we conclude that

$$n \gtrsim m \max\left(\frac{1}{\mathrm{NSR}_{\max}}, \frac{1}{\varepsilon\sqrt{\mathrm{NSR}_{\max}}}\right) \tag{26}$$

allows drastic feature subsampling while preserving utility.

Regime where feature subsampling adds insignificant noise. When

$$\mathrm{NSR}_H \ll \max(\mathrm{NSR}_{\mathcal{X}}, \mathrm{NSR}_{\xi}) \quad \Leftrightarrow \quad \frac{1}{\alpha} \ll 1 + \max(1, \frac{m^2}{n\varepsilon^2}), \tag{27}$$

feature subsampling adds insignificant noise compared to the other noises. When subsampling with parameter $\alpha = 1/m$, this is equivalent to $n \ll \frac{m}{\varepsilon^2}$. In light of (24), one can check that the regime where subsampling noise is insignificant while the total noise is acceptable corresponds to

$$\frac{m}{\varepsilon\sqrt{\mathrm{NSR}_{\max}}} \lesssim n \ll \frac{m}{\varepsilon^2}, \tag{28}$$

which is only feasible when the target privacy satisfies $\varepsilon \ll \sqrt{\mathrm{NSR}_{\max}}$.

Example (compressive clustering with random Fourier Features) When performing compressive clustering using random Fourier Features, we observed empirically on Figure 5 that $\mathrm{NSR}_{\max} \approx 10^{-3}m/(kd)$. Thus the condition (26) for having an acceptable noise level when subsampling can be rewritten

$$n \gtrsim \max\left(10^3 kd, \frac{\sqrt{10^3 kdm}}{\varepsilon}\right). \tag{29}$$

Similarly, subsampling with $\alpha = 1/m$ will induce an insignificant noise level only when $\varepsilon \ll \sqrt{10^{-3}m/(kd)}$ according to Equation (28). This confirms that sketching is compatible with drastic features subsampling for private compressive clustering when working with large collections, but also that subsampling can be perfomed without any impact on the NSR for high privacy levels.

6.5 A Heuristic for Privacy Budget Splitting (Laplacian Noise)

When using unbounded differential privacy, one needs to split the total privacy budget ε between a budget $\varepsilon_\xi \triangleq \gamma\varepsilon$ (where $\gamma \in]0,1[$) used for releasing the sum of sketches $\widetilde{\boldsymbol{\Sigma}}$, and a budget $\varepsilon_\zeta \triangleq (1-\gamma)\varepsilon$ used for releasing the dataset size. This is only needed in the UDP setting since for BDP there is no need to split the privacy budget, given that the dataset size is not considered as sensitive.

We build a heuristic for the ε-DP setting which consists in choosing $\gamma^* \in]0,1[$ minimizing the NSR. In light of Section 6.3, we consider for simplicity $\beta = 1$, i.e. subsampling is only performed on the features but not on the samples. We further focus on random Fourier features, where $\|\boldsymbol{\Phi}^{\mathrm{RFF}}(\mathbf{x})\|_2^2 = m$ does not depend on $\mathbf{x}$, leading to a simplified expression of the Cramer-Rao bound on the NSR from Lemma 27:

$$\mathrm{NSR}_*^{\mathrm{RFF}} = \left(1 + \frac{\sigma_\zeta^2}{2n^2}\right)\left(1 - \frac{1}{n} + \frac{m}{n\|\mathbf{z}\|^2}\left(\frac{1}{\alpha} + \frac{1}{n}\sigma_\xi^2\right)\right) - 1,$$

with $\mathbf{z}$ as the reference signal. By injecting for σ_ζ^2 and σ_ξ^2 the values obtained previously for the UDP Laplacian setting, see Table 1, we get an expression of $\mathrm{NSR}_*^{\mathrm{RFF}}$ as a function of γ, which can be minimized w.r.t. the parameter γ.

Lemma 28. *For random Fourier features, an expression of the parameter γ^* minimizing $\mathrm{NSR}_*^{\mathrm{RFF}}$ is given in Appendix E as a function of ε and n. The following approximations can be derived*

$$\begin{aligned} &\text{when } n \ll 1/\varepsilon, \gamma^*(n,\varepsilon) \approx 1/2 \\ &\text{when } n \gg 1/\varepsilon, \gamma^*(n,\varepsilon) \approx 1 - (n\varepsilon)^{-2/3}. \end{aligned}$$

In practice, it is important to choose γ independently of n in order for the whole mechanism to stay private. Given that the NSR only decreases with n, we have for any $\varepsilon > 0$ and any n_0 that $\arg\min_\gamma \max_{n \geq n_0} \mathrm{NSR}(\gamma, n) = \gamma^*(n_0, \varepsilon)$, yielding a simple rule to choose γ. In light of Section 6.4, in the regime of acceptable noise levels we have

$$n \gtrsim n_0(m,\varepsilon) \triangleq m \max\left(\frac{1}{\mathrm{NSR}_{\max}}, \frac{1}{\varepsilon\sqrt{\mathrm{NSR}_{\max}}}\right) \gg 1/\varepsilon$$

hence a possible heuristic is to choose

$$\gamma(m,\varepsilon) \triangleq \gamma^\star(n_0(m,\varepsilon),\varepsilon) \approx 1 - (n_0\varepsilon)^{-2/3}.$$

Note that this is only a heuristic, allowing to choose γ independently of n but optimized for the worst-case scenario with acceptable utility; even if $n < n_0$ the mechanism will be guaranteed to be private (although with limited utility).

6.6 Choice of the Sketch Size

Because the noise level depends on the sketch size m, the design of a sketching procedure becomes delicate since overestimating m decreases the performance, unlike in the non-private case where increasing m usually only helps. As an illustration of this fact, consider the numerical experiment represented Figure 8 (top row), where we estimate the relative SSE (RSSE) achieved by compressive k-means (CKM) from the ε-DP sketch as a function of its size m. Relative SSE is defined as the ratio between the method SSE, as defined in Equation (2), and the SSE of Lloyd's standard kmeans algorithm, which is not private nor compressive. As expected, in the non-private setting the SSE decreases monotonically with m. However, when $\varepsilon < \infty$ and n is moderate, increasing m (and thus the noise, which is proportional to m according to Lemma 7) results in a worse SSE at some point. This phenomenon is more pronounced when the privacy constraints are higher, i.e. a smaller ε induces a smaller range of "optimal" values for the sketch size. There is thus a trade-off to make between revealing enough information for CKM to succeed (m large enough) and not revealing too much information, such that the noise needed to ensure the privacy guarantee is not too penalizing, this trade-off being more difficult in the high privacy regime.

This behavior can be explained by the observations of Section 6.4 (paragraph "acceptable noise level") relative to the NSR. We consider for conciseness here that no subsampling is used (i.e. $\mathrm{NSR}_H = 0$) and

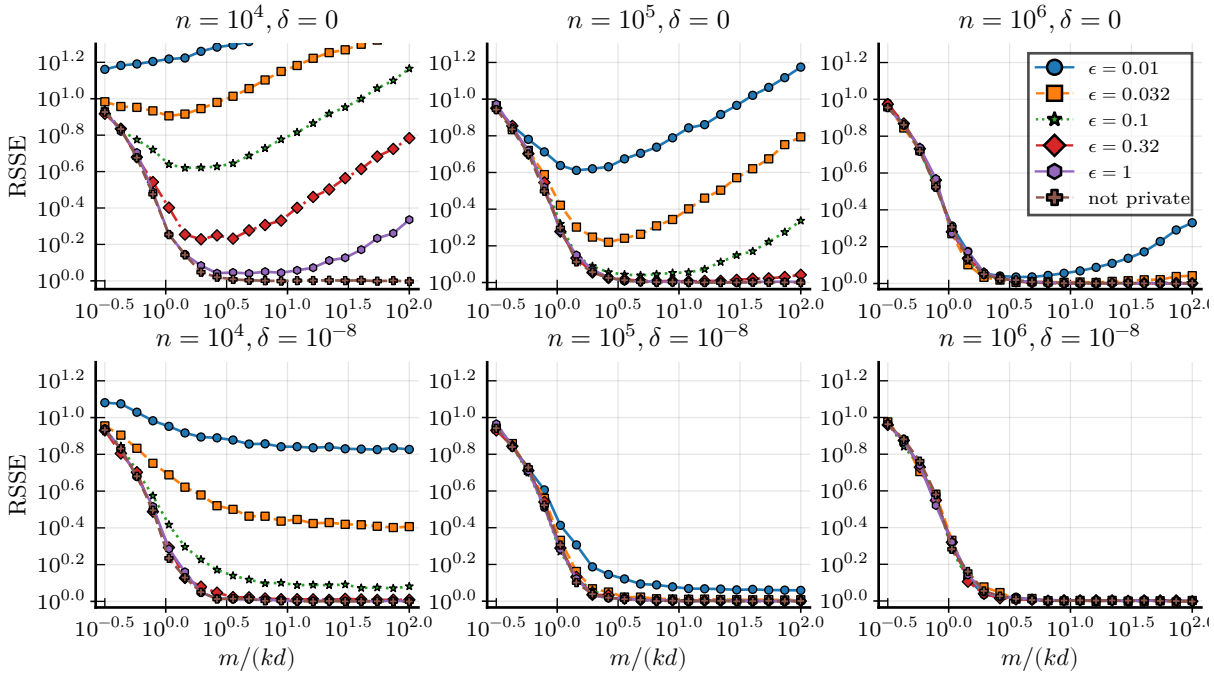

Figure 8: Performance of differentially private compressive k-means as a function of m for $\delta = 0$ (top) and $\delta = 10^{-8}$ (bottom), $n = 10^4, 10^5, 10^6$ and different values of ε. Medians over 200 trials. Synthetic data, $k = 4, d = 8$.

ε-DP ($\delta = 0$). Given that utility is measured w.r.t. the RSSE (which is relative to the optimal error for the given dataset, but agnostic to the true data distribution), we take $\mathbf{z}_{\mathcal{X}}$ as the reference signal to compute the NSR, i.e. we have $\mathrm{NSR}_{\mathcal{X}} = 0$. Utility is then preserved provided that $\mathrm{NSR}_{\xi} \leq \mathrm{NSR}_{\max}$, which according to (29) translates to the condition $n \geq \sqrt{1000kdm}/\varepsilon$. Recall that we also need $m \geq 2kd$ as shown in Section 6.1. These conditions can be rewritten $2 \leq m/(kd) \leq n^2\varepsilon^2/(10^3(kd)^2)$, which is possible only when $n \geq \sqrt{2 \times 10^3 kd}/\varepsilon$. In Figure 8 we have $k = 4, d = 8$, thus this requirement translates respectively for $n = 10^4, 10^5, 10^6$ to the conditions $\varepsilon \geq 0.14, \varepsilon \geq 0.014, \varepsilon \geq 0.0014$, which correspond quite well to what is observed (top row).

As shown on Figure 8 (bottom), relaxing the privacy constraint to allow $\delta > 0$ mitigates the impact of m on the noise to add (recall from theorem 11 that the noise is then proportional to $\sqrt{m}$ instead of m), and that even for smaller values of n. This relaxation has the clear advantage of improving the utility for similar values of n and ε even for small δ, and also facilitates the choice of m, as good utilities can be reached on a wider range of sketch sizes.

7 Discussion and Perspectives

We proposed a framework to learn from potentially massive datasets using limited computational resources, while ensuring the differential privacy of the data providers. Beside being promising as privacy-inducing mechanism in terms of privacy, our framework has several key interesting features compared to other methods from the literature, that are discussed here together with main limitations and perspectives.

Efficient and Distributed Learning Firstly, the computational advantages of non-private sketching [33, 36] remain valid after our addition of a privacy layer. In particular, learning from the sketch can be done with time and space complexities which do not depend on the number of samples n in the dataset. Moreover, the sketching process is embarassingly parallel due to the averaging operation. Sketches coming from several separate data holders can thus be aggregated again after sketching, providing distributed differential privacy for free, without any need for a trusted central party.

Versatility Another advantage is that the sketch, acting as a surrogate for the whole dataset, contains more information than just the output of one specialized algorithm, and can thus be used multiple times. This can be leveraged to solve different learning tasks from a same sketch without breaking privacy, assuming that those tasks can be solved using the same sketching operator [27]. This is what we already observed for random Fourier features, which can be used for both k-means clustering and fitting a Gaussian mixture model, two different but related estimation problems.

This potential versatility of the sketch also allows to run the learning algorithm with different initializations and parameters, producing multiple solutions; the distance to the empirical sketch can be used as a metric to pick the best of these solutions. This is in contrast with usual (e.g. iterative) differentially private methods that can be highly sensitive to the choice of such parameters (which have to be selected *a priori*, as accessing the data for parameter tuning breaks the privacy guarantee). Of course the devil is in the details, and further studies are needed to investigate to what extent it is possible to choose parameters such as the sketch dimension or the "scale parameter" of random Fourier features (see below) so as to combine privacy, utility and versatility. Preliminary investigations indicate that one can find sketch sizes enabling to achieve good utility for both compressive gaussian mixture modeling and compressive k-means with $\delta = 10^{-8}$ and ε above or equal to $10^{-1.5}$

Open challenges Although the sketch serves as a general-purpose synopsis of the dataset, at least *some* a priori knowledge about the data distribution and/or the target task is required when designing the sketch feature map $\mathbf{\Phi} : \mathbf{x} \mapsto f(\Omega^T \mathbf{x})$. We discussed how the nonlinearity f must be selected according to the desired task, and explained in Section 6.6 that the choice of the sketch size m could be seen as a trade-off between performance and privacy. Going for approximate DP mitigates this difficulty. Another crucial point is the choice of the frequencies distribution for Fourier features (Ω is drawn i.i.d. Gaussian in the PCA setting, and this concern does not apply in that case). Even when the general shape of the frequency distribution is selected and only a single scale parameter σ has to be pinned down (σ essentially controls the scale at which we can detect individual clusters), estimating an appropriate value for it is not straightforward. This might be a limitation to using sketching in practice but, on the other side, any heuristic that could be developed in the future to estimate σ should be easy to make private as it releases a single scalar value.

Perspectives Finally, we expect that compressive learning will be extended to more learning tasks in future works. The private sketching framework presented here would be directly transferable to those new algorithms, although the sketch sensitivity would have to be re-computed for novel feature functions. The true potential of private sketching will depend on how well the general field of compressive learning will be able to answer this challenge in the coming years.

References

[1] Amin, K., Dick, T., Kulesza, A., Munoz, A. & Vassilvitskii, S. (2019) Differentially Private Covariance Estimation. in *Advances in Neural Information Processing Systems*, pp. 14190–14199.

[2] Arora, R., braverman, V. & Upadhyay, J. (2018) Differentially Private Robust Low-Rank Approximation. in *Advances in Neural Information Processing Systems 31*, ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett, pp. 4137–4145. Curran Associates, Inc.

[3] Balcan, M.-F., Dick, T., Liang, Y., Mou, W. & Zhang, H. (2017) Differentially private clustering in high-dimensional Euclidean spaces. in *International Conference on Machine Learning*, pp. 322–331.

[4] Balle, B., Barthe, G. & Gaboardi, M. (2018) Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. .

[5] Balle, B. & Wang, Y.-X. (2018) Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. .

[6] Balog, M., Tolstikhin, I. & Schölkopf, B. (2017) Differentially Private Database Release via Kernel Mean Embeddings. *arXiv:1710.01641 [stat]*.

[7] Bassily, R., Smith, A. & Thakurta, A. (2014) Private empirical risk minimization: Efficient algorithms and tight error bounds. in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE.

[8] Blum, A., Dwork, C., McSherry, F. & Nissim, K. (2005) Practical privacy: the SuLQ framework. in *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 128–138. ACM.

[9] Blum, A., Ligett, K. & Roth, A. (2013) A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, **60**(2), 1–25.

[10] Bourrier, A., Davies, M. E., Peleg, T., Pérez, P. & Gribonval, R. (2014) Fundamental Performance Limits for Ideal Decoders in High-Dimensional Linear Inverse Problems. **60**(12), 7928–7946.

[11] Burer, S. & Monteiro, R. D. (2003) A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. **95**(2), 329–357.

[12] Chatalic, A., Gribonval, R. & Keriven, N. (2018) Large-Scale High-Dimensional Clustering with Fast Sketching. in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[13] Chaudhuri, K., Monteleoni, C. & Sarwate, A. D. (2011) Differentially private empirical risk minimization. **12**, 1069–1109.

[14] Chaudhuri, K., Sarwate, A. & Sinha, K. (2012) Near-optimal differentially private principal components. in *Advances in Neural Information Processing Systems*, pp. 989–997.

[15] Cormode, G., Garofalakis, M., Haas, P. J., Jermaine, C. et al. (2011) Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, **4**(1–3), 1–294.

[16] De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M. & Blondel, V. D. (2013) Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, **3**, 1376.

[17] Dinur, I. & Nissim, K. (2003) Revealing information while preserving privacy. in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 202–210. ACM.

[18] Duchi, J. C., Jordan, M. I. & Wainwright, M. J. (2014) Privacy aware learning. *Journal of the ACM (JACM)*, **61**(6), 38.

[19] Dwork, C. (2008) Differential privacy: A survey of results. in *International Conference on Theory and Applications of Models of Computation*, pp. 1–19. Springer.

[20] Dwork, C., McSherry, F., Nissim, K. & Smith, A. (2006) Calibrating Noise to Sensitivity in Private Data Analysis. in *Theory of cryptography conference*, p. 20. Springer.

[21] Dwork, C. & Roth, A. (2014) The Algorithmic Foundations of Differential Privacy. **9**(3), 211–407.

[22] Dwork, C., Talwar, K., Thakurta, A. & Zhang, L. (2014) Analyze gauss: optimal bounds for privacy-preserving principal component analysis. in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing - STOC '14*, pp. 11–20. ACM Press.

[23] Erlingsson, Ú., Pihur, V. & Korolova, A. (2014) Rappor: Randomized aggregatable privacy-preserving ordinal response. in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067. ACM.

[24] Feldman, D., Fiat, A., Kaplan, H. & Nissim, K. (2009) Private Coresets. in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 361–370. ACM.

[25] Feldman, D., Xiang, C., Zhu, R. & Rus, D. (2017) Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. in *Information Processing in Sensor Networks (IPSN), 2017 16th ACM/IEEE International Conference on*, pp. 3–16. IEEE.

[26] Foucart, S. & Rauhut, H. (2013) *A mathematical introduction to compressive sensing*, vol. 1. Birkhäuser Basel.

[27] Gribonval, R., Blanchard, G., Keriven, N. & Traonmilin, Y. (2017) Compressive statistical learning with random feature moments. *arXiv preprint arXiv:1706.07180*.

[28] Hardt, M. & Roth, A. (2012) Beating Randomized Response on Incoherent Matrices. p. 1255.

[29] Imtiaz, H. & Sarwate, A. D. (2016) Symmetric matrix perturbation for differentially-private principal component analysis. in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2339–2343.

[30] Jiang, W., Xie, C. & Zhang, Z. (2016) Wishart mechanism for differentially private principal components analysis. in *Thirtieth AAAI Conference on Artificial Intelligence*.

[31] Kapralov, M. & Talwar, K. (2013) On differentially private low rank approximation. in *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1395–1414. SIAM.

[32] Kenthapadi, K., Korolova, A., Mironov, I. & Mishra, N. (2013) Privacy via the Johnson-Lindenstrauss Transform. *Journal of Privacy and Confidentiality*, **5**(1).

[33] Keriven, N., Bourrier, A., Gribonval, R. & Pérez, P. (2017) Sketching for large-scale learning of mixture models. *Information and Inference: A Journal of the IMA*, **7**(3), 447–508.

[34] Keriven, N., Bourrier, A., Gribonval, R. & Pérez, P. (2017) Sketching for large-scale learning of mixture models. **7**(3), 447–508.

[35] Keriven, N., Deleforge, A. & Liutkus, A. (2018) Blind Source Separation Using Mixtures of Alpha-Stable Distributions. in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 771–775. IEEE.

[36] Keriven, N., Tremblay, N., Traonmilin, Y. & Gribonval, R. (2017) Compressive K-means. in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6369–6373. IEEE.

[37] Kronecker, L. (1884) *Näherungsweise ganzzahlige auflösung linearer gleichungen*.

[38] Lu, Z. & Shen, H. (2019) A Convergent Differentially Private k-Means Clustering Algorithm. in *Advances in Knowledge Discovery and Data Mining*, ed. by Q. Yang, Z.-H. Zhou, Z. Gong, M.-L. Zhang, & S.-J. Huang, vol. 11439, pp. 612–624. Springer International Publishing.

[39] McSherry, F. D. (2009) Privacy integrated queries: an extensible platform for privacy-preserving data analysis. in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 19–30. ACM.

[40] Nissim, K. & Stemmer, U. (2018) Clustering algorithms for the centralized and local models. in *Algorithmic Learning Theory*, pp. 619–653. PMLR.

[41] Nock, R., Canyasse, R., Boreli, R. & Nielsen, F. (2016) k-variates++: more pluses in the k-means++. in *International Conference on Machine Learning*, pp. 145–154.

[42] Park, M., Foulds, J., Chaudhuri, K. & Welling, M. (2016) DP-EM: Differentially Private Expectation Maximization. .

[43] Qardaji, W., Yang, W. & Li, N. (2013) Differentially private grids for geospatial data. in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pp. 757–768. IEEE.

[44] Rahimi, A. & Recht, B. (2008) Random features for large-scale kernel machines. in *Advances in neural information processing systems*, pp. 1177–1184.

[45] Schellekens, V., Chatalic, A., Houssiau, F., De Montjoye, Y.-A., Jacques, L. & Gribonval, R. (2019) Differentially Private Compressive k-Means. in *ICASSP 2019 - 44th International Conference on Acoustics, Speech, and Signal Processing*, pp. 1–5. IEEE.

[46] Schellekens, V. & Jacques, L. (2018) Quantized Compressive K-Means. *IEEE Signal Processing Letters*, **25**(8), 1211–1215.

[47] Stemmer, U. & Kaplan, H. (2018) Differentially private k-means with constant multiplicative error. in *Advances in Neural Information Processing Systems*, pp. 5431–5441.

[48] Su, D., Cao, J., Li, N., Bertino, E. & Jin, H. (2016) Differentially private k-means clustering. in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 26–37. ACM.

[49] Sweeney, L. (2002) k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **10**(05), 557–570.

[50] Team, D. et al. (2017) Learning with privacy at scale. *Online at: https://machinelearning. apple. com/2017/12/06/learning-with-privacy-at-scale. html*.

[51] Testa, M., Valsesia, D., Bianchi, T. & Magli, E. (2019) *Compressed Sensing for Privacy-Preserving Data Processing*. Springer.

[52] Treschev, D. & Zubelevich, O. (2009) *Introduction to the perturbation theory of Hamiltonian systems*. Springer Science & Business Media.

[53] Upadhyay, J. (2018) The Price of Privacy for Low-rank Factorization. p. 12.

[54] Wagner, I. & Eckhoff, D. (2018) Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, **51**(3), 57.

[55] Waldspurger, I. & Waters, A. (2018) Rank optimality for the Burer-Monteiro factorization. .

[56] Wang, D., Ye, M. & Xu, J. (2017) Differentially private empirical risk minimization revisited: Faster and more general. in *Advances in Neural Information Processing Systems*, pp. 2722–2731.

[57] Wang, W., Ying, L. & Zhang, J. (2016) On the Relation Between Identifiability, Differential Privacy, and Mutual-Information Privacy. *IEEE Transactions on Information Theory*, **62**(9), 5018–5029.

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

[59] Wu, Y., Wu, Y., Peng, H., Zeng, J., Chen, H. & Li, C. (2016) Differentially private density estimation via Gaussian mixtures model. in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pp. 1–6. IEEE.

[60] Zhang, J., Xiao, X., Yang, Y., Zhang, Z. & Winslett, M. (2013) PrivGene: differentially private model fitting using genetic algorithms. in *Proceedings of the 2013 international conference on Management of data - SIGMOD '13*, p. 665. ACM Press.

[61] Zhou, S., Ligett, K. & Wasserman, L. (2009) Differential privacy with compression. in *2009 IEEE International Symposium on Information Theory*, pp. 2718–2722.

A Results on Nonresonant Frequencies

In order to prove the sharpness of the sensitivity computed in Lemma 7, we rely on some results from diophantine approximation theory. We recall the definition of nonresonant frequencies.

Definition 23. *The vectors $(\omega_j)_{1\leq j\leq m} \in \mathbb{R}$ are called nonresonant frequencies if they are linearly independent over the rationals. The vectors $(\boldsymbol{\omega}_j)_{1\leq j\leq m} \in \mathbb{R}^d$ are called nonresonant frequency vectors if there exists a vector $\mathbf{v} \in \mathbb{R}^d$ such that the scalars $(\boldsymbol{\omega}_j^T\mathbf{v})_{1\leq j\leq m}$ are nonresonant frequencies.*

Before proving Lemma 6, we introduce a variant of the result in dimension 1.

Lemma 29. *Let $(\omega_j, \varphi_j)_{1\leq j\leq m}$ be real numbers, and f a 2π-periodic function such that there exists z at which f is continuous and reaches its maximum. If the $(\omega_j)_{1\leq j\leq m}$ are linearly independent over the rationals, then $\sup_{x\in\mathbb{R}} \inf_{j\in[\![1;m]\!]} f(\omega_j x - \varphi_j) = \sup_{x\in\mathbb{R}} f(x)$.*

Proof. Let $z \in [0, 2\pi[$ be a point at which f reaches its maximum, i.e., $z \in \arg\max_{[0,2\pi[} f(z)$, and at which f is continuous. Using this continuity assumption, the result amounts to saying that one can find $t \in \mathbb{R}$ such that the $(\omega_j t - \varphi_j - z)_{1\leq j\leq m}$ are simultaneously arbitrary close to $2\pi\mathbb{Z}$. Denoting $d(x, S) = \inf\{|x - s| : s \in S\}$, this is equivalent to saying that for any $\varepsilon > 0$, we can find a real t such that we simultaneously have $d((\omega_j t - \varphi_j - z)/(2\pi), \mathbb{Z}) < \varepsilon$ for all $j \in [\![1, m]\!]$. This derives directly from Kronecker's theorem [37] on diophantine approximation, given that the $\omega_j/(2\pi)$ are linearly independent over the rationals. □

Proof of Lemma 6. We propose to convert the problem to its one-dimensional counterpart.

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{j\in[\![1;m]\!]} f(\boldsymbol{\omega}_j^T\mathbf{x} - \varphi_j) = \sup_{\mathbf{v}\in\mathbb{R}^d:\|\mathbf{v}\|=1} \sup_{x\in\mathbb{R}} \inf_{j\in[\![1;m]\!]} f(x\boldsymbol{\omega}_j^T\mathbf{v} - \varphi_j) \tag{30}$$

Let $\mathbf{v}$ be such that the scalars $a_j \triangleq \boldsymbol{\omega}_j^T\mathbf{v}$ (for $1 \leq j \leq m$) are nonresonant, which exists because the vectors $(\boldsymbol{\omega}_j)_{1\leq j\leq m}$ are themselves nonresonant. The quantity (30) is upper-bounded by $\sup_{x\in\mathbb{R}} f(x) = f(z)$, and can be lower-bounded by

$$\sup_{x\in\mathbb{R}} \inf_{j\in[\![1;m]\!]} f(xa_j - \varphi_j) = \sup_{x\in\mathbb{R}} f(x)$$

where the last equality comes from Lemma 29, the a_j being nonresonant. □

B Results without subsampling

Proof of Lemma 12. We have

$$\begin{aligned}
\Delta_2^{\mathrm{B}}(\boldsymbol{\Sigma}^{\mathrm{RFF}})^2 &= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \|\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}(\mathbf{y})\|_2^2 = \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m |\boldsymbol{\Phi}(\mathbf{x})_j - \boldsymbol{\Phi}(\mathbf{y})_j|^2 \\
&= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m |(\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j) + i\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \tfrac{\pi}{2})) - (\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) + i\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2}))|^2 \\
&= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m (\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j) - \rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j))^2 + (\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \tfrac{\pi}{2}) - \rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2}))^2 \\
&= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m 2(1 - (\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j)\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) + \rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \tfrac{\pi}{2})\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2}))
\end{aligned} \tag{31}$$

- For **unquantized features**, we have $\rho = \cos$ and $\rho(\cdot - \frac{\pi}{2}) = \sin$, hence

$$\begin{aligned}\Delta_2^{\mathrm{B}}(\mathbf{\Sigma}^{\mathrm{RFF}})^2 &= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m 2(1 - (\cos(\boldsymbol{\omega}_j^T\mathbf{x} + u_j)\cos(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) + \sin(\boldsymbol{\omega}_j^T\mathbf{x} + u_j)\sin(\boldsymbol{\omega}_j^T\mathbf{y} + u_j)) \\ &= \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m 2(1 + \cos(\boldsymbol{\omega}_j^T(\mathbf{x} - \mathbf{y}) - \pi)) \\ &= 2\left(m + \sup_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^m \cos(\boldsymbol{\omega}_j^T\mathbf{z} - \pi)\right) \\ &= 4m\end{aligned}$$

  by Lemma 6 using the nonresonant property of the frequencies.
- For **quantized features**, we reuse the quantities defined in the proof of Lemma 7, i.e. we denote $f(\cdot) \triangleq \rho(\cdot) + \rho(\cdot - \frac{\pi}{2})$ and, for any $\boldsymbol{\varphi} = [\varphi_1, ..., \varphi_m]$, define $f_{\boldsymbol{\varphi}}(\mathbf{x}) = \sum_{j=1}^m f(\boldsymbol{\omega}_j^T\mathbf{x} - \varphi_j)$. Starting from the generic expression (31) we get

$$\Delta_2^{\mathrm{B}}(\mathbf{\Sigma}^{\mathrm{RFF}})^2 = 2\left(m - \inf_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m (\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j)\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) + \rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \tfrac{\pi}{2})\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2})\right)$$

  For any fixed $\mathbf{x} \in \mathbb{R}^d$, we have $\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j) = \pm 2^{-1/2}$ and $\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \frac{\pi}{2}) = \pm 2^{-1/2}$, thus using the same arguments developed in the proof of Lemma 7, these are some $\varphi_j \in \{0, \pi/2, \pi, 3\pi/2\}$ such that

$$\begin{aligned}&\inf_{\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m (\rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j)\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) + \rho(\boldsymbol{\omega}_j^T\mathbf{x} + u_j - \tfrac{\pi}{2})\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2}) \\ &= 2^{-1/2} \inf_{\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m \pm\rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j) \pm \rho(\boldsymbol{\omega}_j^T\mathbf{y} + u_j - \tfrac{\pi}{2}) \\ &= 2^{-1/2} \inf_{\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m f(\boldsymbol{\omega}_j^T\mathbf{y} + u_j + \varphi_j) \\ &= -2^{-1/2} \sup_{\mathbf{y}\in\mathbb{R}^d} \sum_{j=1}^m f_{\pi - \boldsymbol{\varphi} - \mathbf{u}}(\boldsymbol{\omega}_j^T\mathbf{y}) \\ &= -m\end{aligned}$$

  which is independent of the choice of $\mathbf{x}$ and concludes the proof.

□

C Proofs on Sketching with Subsampling

C.1 General results

Proof of Lemma 15. We define the permutation of a set of masks as $\sigma((\mathbf{h}_1, ..., \mathbf{h}_n)) = (\mathbf{h}_{\sigma(1)}, ..., \mathbf{h}_{\sigma(n)})$ for $\sigma \in \mathcal{S}_n$. For any set of masks $H \in \mathcal{H}^n$, and any dataset $\mathcal{X}$ such that $|\mathcal{X}| = n$, we denote $p_{\mathcal{X}}(\cdot|H) = p_{\mathbf{\Sigma}_{\mathcal{L},H}(\mathcal{X})}(\cdot)$ the density of $\mathbf{\Sigma}_{\mathcal{L},H}(\mathcal{X})$. Unless otherwise specified, $p_{\mathcal{X}}$ denotes the density of $\bar{\mathbf{\Sigma}}_{\mathcal{L}}(\mathcal{X})$.

We prove the result for a real-valued feature map $\mathbf{\Phi}$, and discuss the complex case at the end of the proof. We will prove that $\sup_{\mathcal{X},\mathcal{Y}\in\mathcal{D}:\mathcal{X}\overset{\cup}{\sim}\mathcal{Y}} \sup_{\mathbf{s}\in\mathcal{Z}} p_{\mathcal{X}}(\mathbf{s})/p_{\mathcal{Y}}(\mathbf{s}) = \exp(\varepsilon^*)$, which is equivalent to the lemma statement. If $H_{n-1} = (\mathbf{h}_1, ..., \mathbf{h}_{n-1})$ is a set of masks and $\mathbf{h}_n$ a single mask, defining $H = (\mathbf{h}_1, ..., \mathbf{h}_n)$ we use the notations $\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\cdot) \triangleq \bar{\mathbf{\Sigma}}_H(\cdot)$ and $p(\mathbf{s}|H_{n-1}, \mathbf{h}_n) \triangleq p(\mathbf{s}|H)$. In the following $\mathbf{h}_n$, H_{n-1} and H are implicitly drawn (independently) from respectively $p_{\mathbf{h}}$, $p_{\mathbf{h}}^{n-1}$ and $p_{\mathbf{h}}^n$, where $p_{\mathbf{h}}$ is the probability distribution of the masks from Definition 21. Considering $\mathcal{X}, \mathcal{Y} \in \mathcal{D}$ such that $\mathcal{X} \overset{\cup}{\sim} \mathcal{Y}$ we distinguish two cases, depending whether $|\mathcal{X}| = |\mathcal{Y}| + 1$ or $|\mathcal{X}| = |\mathcal{Y}| - 1$.

Case $|\mathcal{X}| = |\mathcal{Y}|+1$ For any $\mathcal{X} \overset{\cup}{\sim} \mathcal{Y}$, denoting $n = |\mathcal{X}|$ and assuming for now that $|\mathcal{X}| = |\mathcal{Y}|+1$, there is by Definition 9 a permutation $\sigma \in \mathcal{S}_n$ such that $\sigma(\mathcal{X})$ satisfies $\sigma(\mathcal{X}) \overset{\cup}{\approx} \mathcal{Y}$. We have $\bar{\mathbf{\Sigma}}_H(\sigma(\mathcal{X})) = \bar{\mathbf{\Sigma}}_{\sigma^{-1}(H)}(\mathcal{X})$, and as the masks are drawn i.i.d. according to $p_{\mathbf{h}}$, we obtain

$$\begin{aligned} p_{\mathcal{X}}(\mathbf{s}) &= \mathbf{E}_H[p_{\mathcal{X}}(\mathbf{s}|H)] = \mathbf{E}_H\big[p_{\mathcal{X}}(\mathbf{s}|\sigma^{-1}(H))\big] = \mathbf{E}_H\Big[p_{\sigma(\mathcal{X})}(\mathbf{s}|H)\Big] \\ &= \mathbf{E}_{H_{n-1},\mathbf{h}_n}\Big[p_{\sigma(\mathcal{X})}(\mathbf{s}|H_{n-1},\mathbf{h}_n)\Big] = \mathbf{E}_{\mathbf{h}_n}\mathbf{E}_{H_{n-1}}\Big[p_{\sigma(\mathcal{X})}(\mathbf{s}|H_{n-1},\mathbf{h}_n)\Big] \\ p_{\mathcal{Y}}(\mathbf{s}) &= \mathbf{E}_{H_{n-1}}[p_{\mathcal{Y}}(\mathbf{s}|H_{n-1})] \end{aligned}$$

As a consequence we have

$$\frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} = \frac{\mathbf{E}_{\mathbf{h}_n}\mathbf{E}_{H_{n-1}}\Big[p_{\sigma(\mathcal{X})}(\mathbf{s}|H_{n-1},\mathbf{h}_n)\Big]}{\mathbf{E}_{H_{n-1}}[p_{\mathcal{Y}}(\mathbf{s}|H_{n-1})]} = \frac{\mathbf{E}_{\mathbf{h}_n}\mathbf{E}_{H_{n-1}}\exp(-\frac{1}{b}\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X}))\|_1)}{\mathbf{E}_{H_{n-1}}\exp(-\frac{1}{b}\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y})\|_1)} \tag{32}$$

Note that for any $H_{n-1}, \mathbf{h}_n$ we have $\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X})) = \bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y}) + \frac{1}{\alpha}\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n$ by definition of σ and thus for any $H_{n-1}, \mathbf{h}_n, \mathbf{s}$ we have

$$\begin{aligned} -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X}))\|_1 &= -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y}) - \frac{1}{\alpha}\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n\|_1 \\ &\leq -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y})\|_1 + \|\frac{1}{\alpha}\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n\|_1. \end{aligned} \tag{33}$$

Equality holds iff for all $j \in [\![1,m]\!]$, $(\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X})))_j$ and $(\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n)_j$ have the same sign or any of the two terms is null. Define $c \triangleq \max_{1\leq i\leq n}\|\mathbf{\Phi}(\mathbf{x}_i)\|_\infty$. For any choice of binary masks H_{n-1}, we have $\|\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X}))\|_\infty \leq nc\frac{1}{\alpha} \triangleq M$. In particular, if we define $\tilde{\mathbf{s}} \triangleq M\operatorname{sign}(\mathbf{\Phi}(\mathbf{x}_n))$, where sign is applied pointwise, $\tilde{\mathbf{s}}$ yields equality in Equation (33) for all $H_{n-1}, \mathbf{h}_n$ simultaneously. Using Equation (33) in Equation (32) and taking the supremum over $\mathbf{s}$, we get

$$\begin{aligned} \sup_{\mathbf{s}\in\mathcal{Z}}\frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} &\leq \sup_{\mathbf{s}\in\mathcal{Z}}\frac{\mathbf{E}_{\mathbf{h}_n}\mathbf{E}_{H_{n-1}}\exp(-\frac{1}{b}\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y})\|_1 + \frac{1}{b}\|\frac{1}{\alpha}\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n\|_1)}{\mathbf{E}_{H_{n-1}}\exp(-\frac{1}{b}\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{Y})\|_1)} \\ &= \mathbf{E}_{\mathbf{h}_n}\exp\Big(\frac{1}{b}\frac{1}{\alpha}\|\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n\|_1\Big) \end{aligned}$$

but we also have

$$\sup_{\mathbf{s}\in\mathcal{Z}}\frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} \geq \frac{p_{\mathcal{X}}(\tilde{\mathbf{s}})}{p_{\mathcal{Y}}(\tilde{\mathbf{s}})} = \mathbf{E}_{\mathbf{h}_n}\exp\Big(\frac{1}{b}\frac{1}{\alpha}\|\mathbf{\Phi}(\mathbf{x}_n)\odot\mathbf{h}_n\|_1\Big),$$

therefore equality holds.

Case $|\mathcal{X}| = |\mathcal{Y}| - 1$ We assumed so far $|\mathcal{X}| = |\mathcal{Y}| + 1$, but now if $|\mathcal{X}| + 1 = |\mathcal{Y}| = n$, there is σ such that $\sigma(\mathcal{Y}) \overset{\cup}{\approx} \mathcal{X}$ and we have $\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{X}) = \bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{Y})) - \frac{1}{\alpha}\mathbf{\Phi}(\mathbf{y}_n)\odot\mathbf{h}_n$. Another triangular inequality yields

$$\begin{aligned} -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{X})\|_1 &= -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{Y})) + \frac{1}{\alpha}\mathbf{\Phi}(\mathbf{y}_n)\odot\mathbf{h}_n\|_1 \\ &\leq -\|\mathbf{s}-\bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{X}))\|_1 + \|-\frac{1}{\alpha}\mathbf{\Phi}(\mathbf{y}_n)\odot\mathbf{h}_n\|_1. \end{aligned} \tag{34}$$

Using Jensen's inequality (all quantities are positive and $x \mapsto 1/x$ is convex on $\mathbb{R}_+$) we get

$$\begin{aligned}
\frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} = \frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\sigma(\mathcal{Y})}(\mathbf{s})} &= \frac{\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{X}))\|_1}{\mathbf{E}_{\mathbf{h}_n}\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{Y}))\|_1)} \\
&\leq \mathbf{E}_{\mathbf{h}_n} \frac{\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1}}(\mathcal{X})\|_1)}{\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{Y}))\|_1)} \\
&\leq \mathbf{E}_{\mathbf{h}_n} \frac{\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\mathcal{Y})\|_1) \exp(\frac{1}{b}\|\frac{1}{\alpha}\mathbf{\Phi}(\mathbf{y}_n) \odot \mathbf{h}_n\|_1)}{\mathbf{E}_{H_{n-1}} \exp(-\frac{1}{b}\|\mathbf{s} - \bar{\mathbf{\Sigma}}_{H_{n-1},\mathbf{h}_n}(\sigma(\mathcal{Y}))\|_1)} \\
&= \mathbf{E}_{\mathbf{h}_n} \exp(\frac{1}{b}\frac{1}{\alpha}\|\mathbf{\Phi}(\mathbf{y}_n) \odot \mathbf{h}_n\|_1).
\end{aligned}$$

Conclusion Previous results hold for any dataset size $|\mathcal{X}| \in \mathbb{N}$. We now take the supremum over $\mathcal{X}, \mathcal{Y}$, which includes both cases $|\mathcal{X}| = |\mathcal{Y}| + 1$ and $|\mathcal{Y}| = |\mathcal{X}| + 1$; the supremum is the same in both cases, and we have the equality from the first case. Thus

$$\sup_{\mathcal{X},\mathcal{Y} \in \mathcal{D}:\mathcal{X} \overset{\cup}{\sim} \mathcal{Y}} \sup_{\mathbf{s} \in \mathcal{Z}} \frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} = \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}_n} \exp\left(\frac{1}{b}\frac{1}{\alpha}\|\mathbf{\Phi}(\mathbf{x}) \odot \mathbf{h}_n\|_1\right) = \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right),$$

which concludes the proof.

Complex case If $\mathbf{\Phi}$ is complex, the same proof holds using the canonical isomorphism between $\mathbb{C}^m$ and $\mathbb{R}^{2m}$. Indeed, an equivalent of Equation (32) can be established using Definition 11 of a complex Laplace random variable. The triangle inequality Equation (33) holds in a similar manner by considering complex and real parts independently, and $\tilde{\mathbf{s}}$ can be defined as $\tilde{\mathbf{s}} = M(\operatorname{sign}(\Re\mathbf{\Phi}(\mathbf{x}_n)) + i \operatorname{sign}(\Im\mathbf{\Phi}(\mathbf{x}_n)))$. We get

$$\begin{aligned}
\sup_{\mathcal{X},\mathcal{Y} \in \mathcal{D}:\mathcal{X} \overset{\cup}{\sim} \mathcal{Y}} \sup_{\mathbf{s} \in \mathcal{Z}} \frac{p_{\mathcal{X}}(\mathbf{s})}{p_{\mathcal{Y}}(\mathbf{s})} &= \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}_n} \exp\left(\frac{1}{b}\frac{1}{\alpha}(\|\Re\mathbf{\Phi}(\mathbf{x}) \odot \mathbf{h}_n\|_1 + \|\Im\mathbf{\Phi}(\mathbf{x}) \odot \mathbf{h}_n\|_1)\right) \\
&= \sup_{\mathbf{x} \in E} \mathbf{E}_{\mathbf{h}} \exp\left(\frac{1}{b} Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})\right),
\end{aligned}$$

which concludes the proof. □

C.2 Random Fourier Features

Proof of Lemma 17. This proof bears strong similarities with the proof of Lemma 7, and we therefore use the same notations and tools. In particular, we recall that $f(\cdot) \triangleq \rho(\cdot) + \rho(\cdot - \pi/2)$, and that $\sup_{x \in \mathbb{R}} f(x) = \sqrt{2}$ for both complex exponential case and one-bit quantization. We also denote $\operatorname{supp}(\mathbf{h}) = \{j \in [\![1, m]\!] \,|\, h_j \neq 0\}$ the support of $\mathbf{h}$.

By analogy with Equations (10) and (11), but summing only on the frequencies that appear in the mask $\mathbf{h}$, denoting $f_{\boldsymbol{\varphi},\mathbf{h}}(\mathbf{x}) \triangleq \sum_{j \in \operatorname{supp}(\mathbf{h})} f(\boldsymbol{\omega}_j^T \mathbf{x} - \varphi_j)$, the quantities $Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h})$ and $Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h})$ can be expressed as

$$\begin{aligned}
Q_1^{\mathrm{U}}(\mathbf{x}, \mathbf{h}) &= \frac{1}{\alpha} \sum_{j \in \operatorname{supp}(\mathbf{h})} |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j)| + |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j - \frac{\pi}{2})| \\
&= \frac{1}{\alpha} \sup_{\boldsymbol{\varphi} \in \{0, \pi/2, \pi, 3\pi/2\}^m} f_{\boldsymbol{\varphi} - \mathbf{u}, \mathbf{h}}(\mathbf{x}). \\
Q_1^{\mathrm{B}}(\mathbf{x}, \mathbf{y}, \mathbf{h}) &= \frac{1}{\alpha} \sum_{j \in \operatorname{supp}(\mathbf{h})} |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j) - \rho(\boldsymbol{\omega}_j^T \mathbf{y} + u_j)| + |\rho(\boldsymbol{\omega}_j^T \mathbf{x} + u_j - \frac{\pi}{2}) - \rho(\boldsymbol{\omega}_j^T \mathbf{y} + u_j - \frac{\pi}{2})| \\
&= \frac{1}{\alpha} \sup_{\boldsymbol{\varphi} \in \{0, \pi/2, \pi, 3\pi/2\}^m} f_{\boldsymbol{\varphi} - \mathbf{u}, \mathbf{h}}(\mathbf{x}) - f_{\boldsymbol{\varphi} - \mathbf{u}, \mathbf{h}}(\mathbf{y}) \\
&= \frac{1}{\alpha} \sup_{\boldsymbol{\varphi} \in \{0, \pi/2, \pi, 3\pi/2\}^m} f_{\boldsymbol{\varphi} - \mathbf{u}, \mathbf{h}}(\mathbf{x}) + f_{\boldsymbol{\varphi} - \mathbf{v}, \mathbf{h}}(\mathbf{y}).
\end{aligned}$$

where $v_j = u_j + \pi$ for $1 \le j \le m$. The frequencies being nonresonant, a direct consequence of Lemma 6 is that for each $\boldsymbol{\varphi} \in \mathbb{R}^m$, $\sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}\in\mathcal{H}_r} f_{\boldsymbol{\varphi},\mathbf{h}}(\mathbf{x}) = \sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}\in\mathcal{H}_r} \sum_{j\in\mathrm{supp}(\mathbf{h})} f(\boldsymbol{\omega}_j^T\mathbf{x} - \varphi_j) = r\sup_{x\in\mathbb{R}} f(x) = r\sqrt{2}$. The supremum being independent of $\boldsymbol{\varphi}$ this yields

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}\in\mathcal{H}_r} \sup_{\boldsymbol{\varphi}\in\{0,\pi/2,\pi,3\pi/2\}^m} f_{\boldsymbol{\varphi}-\mathbf{u},\mathbf{h}}(\mathbf{x}) \ge \sup_{\boldsymbol{\varphi}\in\{0,\pi/2,\pi,3\pi/2\}^m} \sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}\in\mathcal{H}_r} f_{\boldsymbol{\varphi}-\mathbf{u},\mathbf{h}}(\mathbf{x}) = r\sqrt{2}.$$

As we also have (even for resonant frequencies) the upper bound

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}\in\mathcal{H}_r} \sup_{\boldsymbol{\varphi}\in\mathbb{R}^m} f_{\boldsymbol{\varphi},\mathbf{h}}(\mathbf{x}) \le \sup_{\mathbf{x}\in\mathbb{R}^d} \sup_{\mathbf{h}\in\mathcal{H}_r} \sup_{\boldsymbol{\varphi}\in\mathbb{R}^m} f_{\boldsymbol{\varphi},\mathbf{h}}(\mathbf{x}) \le r\sqrt{2}$$

we get for each $\mathbf{h} \in \mathcal{H}_r$

$$m\sqrt{2} \le \sup_{\mathbf{x}\in\mathbb{R}^d} \inf_{\mathbf{h}'\in\mathcal{H}_r} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}') \le \sup_{\mathbf{x}\in\mathbb{R}^d} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}) \le \sup_{\mathbf{x}\in\mathbb{R}^d} \sup_{\mathbf{h}'\in\mathcal{H}_r} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h}') = m\sqrt{2}.$$

In the BDP setting, the supremum is taken independently on $\mathbf{x}$ and $\mathbf{y}$, thus for any $\mathbf{h}$ we have $\sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}) = 2\sup_{\mathbf{x}\in\mathbb{R}^d} Q_1^{\mathrm{U}}(\mathbf{x},\mathbf{h})$ and

$$2m\sqrt{2} = \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \inf_{\mathbf{h}'\in\mathcal{H}_r} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}') \le \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}) \le \sup_{\mathbf{x},\mathbf{y}\in\mathbb{R}^d} \sup_{\mathbf{h}'\in\mathcal{H}_r} Q_1^{\mathrm{B}}(\mathbf{x},\mathbf{y},\mathbf{h}') = 2m\sqrt{2}.$$

□

D Derivation of the noise-signal ratio

Lemma 30. *Let X denote the mean of n' samples taken without replacement from a collection $x_1, ..., x_n$. Let $\sigma^2 = \frac{1}{n}\sum_{i=1}^n |x_i - \mu|^2$, then we have*

$$Var(X) = \frac{\sigma^2}{n'}\frac{n-n'}{n-1}.$$

Proof. Denote $X = \frac{1}{n}\sum_{i=1}^n g_i x_i$, with $g_i = 1$ if x_i is selected, and 0 otherwise (and as a consequence, $\sum_{i=1}^n g_i = n'$). For any $1 \le i < j \le n$, the marginal of g_i is uniform and $\mathbf{E}(g_i g_j) = P[g_i g_j = 1] = P[g_i = 1 \text{ and } g_j = 1] = P[z = 2]$ for z a random variable having an hypergeometric law of parameters $(n, 2/n, n')$.

$$\begin{aligned}
\mathrm{Var}(g_i) &= \mathbf{E}|g_i|^2 - |\mathbf{E}g_i|^2 = \frac{n'}{n}(1-\frac{n'}{n}) = \frac{n'(n-n')}{n^2} \\
\mathrm{Cov}(g_i, g_j) &= \mathbf{E}(g_i g_j) - \mathbf{E}(g_i)\mathbf{E}(g_j) \\
&= \frac{n'(n'-1)}{n(n-1)} - \frac{(n')^2}{n^2} \text{ (hypergeometric law)} \\
&= \frac{n'}{n}\left(\frac{n'-1}{n-1} - \frac{n'}{n}\right) = \frac{n'(n'-n)}{n^2(n-1)} \\
\mathrm{Var}(\frac{1}{n'}\sum_{i=1}^n g_i x_i) &= \frac{1}{(n')^2}\left(\sum_{i=1}^n \mathrm{Var}_{\mathbf{g}}(x_i g_i) + 2\sum_{1\le i<j\le n} \mathrm{Cov}(x_i g_i, x_j g_j)\right) \\
&= \frac{1}{(n')^2}\left(\frac{n'(n-n')}{n^2}\sum_{i=1}^n x_i^2 + 2\frac{n'(n'-n)}{n^2(n-1)}\sum_{1\le i<j\le n} x_i x_j\right) \\
&= \frac{1}{n'}\frac{n-n'}{n^2}\left(\sum_{i=1}^n x_i^2 - 2\frac{1}{n-1}\sum_{1\le i<j\le n} x_i x_j\right)
\end{aligned}$$

Let $\mu = \frac{1}{n}\sum_{i=1}^n x_i$, and $\sigma^2 = \frac{1}{n}\sum_{i=1}^n (x_i - \mu)^2$. Note that

$$\begin{aligned} n\sigma^2 &= \sum_{i=1}^n (x_i - \mu)^2 \\ &= \sum_{i=1}^n x_i^2 - n\mu^2 \\ &= \sum_{i=1}^n x_i^2 - \frac{1}{n}\Big(\sum x_i^2 + 2\sum_{i<j} x_i x_j\Big) \\ &= \frac{n-1}{n}\left(\sum x_i^2 - 2\frac{1}{n-1}\sum_{i<j} x_i x_j\right) \end{aligned}$$

As a consequence

$$\begin{aligned} \mathrm{Var}\Big(\frac{1}{n}\sum_{i=1}^n g_i x_i\Big) &= \frac{1}{n'}\frac{n-n'}{n^2}\frac{n^2}{n-1}\sigma^2 \\ &= \frac{\sigma^2}{n'}\frac{n-n'}{n-1}. \end{aligned}$$

□

We can now give the proof.

Proof of Lemma 26. We define the error as $\mathbf{e} \triangleq \mathbf{s}(\mathcal{X}) - \mathbf{s}$ for some reference signal $\mathbf{s}$, which can be either $\mathbf{z}_{\mathcal{X}}$ or the true sketch $\mathbf{z}$. The noise level is $\mathbf{E}\|\mathbf{e}\|_2^2$, and the noise-to-signal ratio is defined as $\mathrm{NSR} = \mathbf{E}\|\mathbf{e}\|_2^2/\|\mathbf{s}\|_2^2$. In these expressions, the expectations are taken w.r.t. the randomness of the sketching mechanism when $\mathbf{z}_{\mathcal{X}}$ is chosen as the reference signal, and w.r.t. both the randomness of the mechanism and the draw of $\mathcal{X}$ when $\mathbf{z}$ is the reference signal. We denote $\boldsymbol{\Sigma}$ the clean sum of features, $n' = \beta n$, $\boldsymbol{\Sigma}_{n'}$ the sum of features computed on a random subset of the collection, $\boldsymbol{\Sigma}_{H,n'}$ the mechanism combining both types of subsampling, i.e.

$$\begin{aligned} \boldsymbol{\Sigma}(\mathcal{X}) &= \sum_{i=1}^n \boldsymbol{\Phi}(\mathbf{x}_i) \\ \boldsymbol{\Sigma}_{n'}(\mathcal{X}) &= \sum_{i=1}^n g_i \boldsymbol{\Phi}(\mathbf{x}_i) \\ \boldsymbol{\Sigma}_{H,n'}(\mathcal{X}) &= \frac{1}{\alpha}\sum_{i=1}^n g_i(\boldsymbol{\Phi}(\mathbf{x}_i) \odot \mathbf{h}_i) \\ \mathbf{s}(\mathcal{X}) &= \frac{1}{n'}(\boldsymbol{\Sigma}_{H,n'}(\mathcal{X}) + \boldsymbol{\xi}). \end{aligned}$$

Thus the error can be decomposed as

$$\begin{aligned} \mathbf{e} &= \frac{1}{n}\widetilde{\boldsymbol{\Sigma}}(\mathcal{X}) - \mathbf{s} \\ &= \underbrace{\frac{1}{n}\Sigma(\mathcal{X}) - \mathbf{s}}_{\mathbf{e}_{\mathcal{X}}} + \underbrace{\tfrac{1}{n'}\boldsymbol{\Sigma}_{n'}(\mathcal{X}) - \tfrac{1}{n}\Sigma(\mathcal{X})}_{\mathbf{e}_{n'}} + \underbrace{\tfrac{1}{n'}\big(\boldsymbol{\Sigma}_{H,n'}(\mathcal{X}) - \boldsymbol{\Sigma}_{n'}(\mathcal{X})\big)}_{\mathbf{e}_H} + \underbrace{\tfrac{1}{n'}\boldsymbol{\xi}}_{\mathbf{e}_\xi}. \end{aligned}$$

We now estimate the noise level of each of these components separately.

Without noise nor subsampling. When no noise is added ($\boldsymbol{\xi} = \zeta = 0$), and all features of all samples are used ($r = m$, no subsampling), then $\mathbf{s}(\mathcal{X}) = \mathbf{z}_{\mathcal{X}} = \mathcal{A}(\pi_{\mathcal{X}}) = \frac{\boldsymbol{\Sigma}(\mathcal{X})}{|\mathcal{X}|}$. When the true sketch is

chosen as the reference signal, we have:

$$\begin{aligned}
\mathbf{e}_{\mathcal{X}} &= \frac{1}{n}\sum \mathbf{\Phi}(\mathbf{x}_i) - \mathbf{z} \\
\mathbf{E}_{\mathcal{X}}\, \mathbf{e}_{\mathcal{X}} &= 0 \\
\|\mathbf{e}_{\mathcal{X}}\|_2^2 &= \|\mathbf{z}_{\mathcal{X}} - \mathbf{z}\|_2^2 \\
\mathbf{E}_{\mathcal{X}}\, \|\mathbf{e}_{\mathcal{X}}\|_2^2 &= \sum_{j=1}^{m} \mathrm{Var}_{\mathcal{X}}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{\Phi}(\mathbf{x}_i)_j\right) = \frac{1}{n}\sum_{j=1}^{m} \mathrm{Var}_{\mathbf{x}}\big(\mathbf{\Phi}(\mathbf{x})_j\big) \\
\boxed{\mathbf{E}_{\mathcal{X}}\, \|\mathbf{e}_{\mathcal{X}}\|_2^2 = \frac{1}{n}\big(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|^2 - \|\mathbf{z}\|^2\big)}
\end{aligned}$$

If $\mathbf{z}_{\mathcal{X}}$ is chosen as the reference signal, then $\mathbf{e}_{\mathcal{X}} = 0$, $\boxed{\mathbf{E}\|\mathbf{e}_{\mathcal{X}}\|_2^2 = 0}$.

Additive noise (for privacy). The noise contribution due to the additive noise is $\mathbf{e}_\xi = \boldsymbol{\xi}/n'$, thus

$$\begin{aligned}
\mathbf{E}_\xi \mathbf{e}_\xi &= 0 \\
\mathbf{E}_\xi \|\mathbf{e}_\xi\|_2^2 &= \frac{1}{(n')^2} m \mathbf{E}\big[\xi_i^2\big] \\
\boxed{\mathbf{E}_\xi \|\mathbf{e}_\xi\|_2^2 = \frac{m}{(n')^2}\sigma_\xi^2}
\end{aligned}$$

and is independent from the reference signal. Here σ_ξ^2 is the noise level such that the whole mechanism (including the sampling step) is ε-DP. It is thus computed using a privacy level $\varepsilon' = \log(1+(\exp(\varepsilon)-1)/\beta)$.

Samples subsampling We consider here the noise contribution due to the dataset subsampling operation. We have

$$\begin{aligned}
\mathbf{e}_{n'} &= \tfrac{1}{n'}\mathbf{\Sigma}_{n'}(\mathcal{X}) - \tfrac{1}{n}\mathbf{\Sigma}(\mathcal{X}) \\
\mathbf{E}_{\mathbf{g}} \mathbf{e}_{n'} &= 0
\end{aligned}$$

The noise level here depends on the subsampling strategy. We consider two cases

- *sampling of n' samples out of n without replacement (denoted* $\mathrm{WOR}(n, n')$*):*

$$\begin{aligned}
\mathbf{E}_{\mathbf{g}\sim\mathrm{WOR}(n,n')}\|\mathbf{e}_{n'}\|^2 &= \sum_{j=1}^{m} \mathrm{Var}_{\mathbf{g}}\left(\frac{1}{n'}\sum_{i=1}^{n} g_i \mathbf{\Phi}(\mathbf{x}_i)_j\right) \\
&= \sum_{j=1}^{m}\left(\frac{\sum_{i=1}^{n} |\mathbf{\Phi}(\mathbf{x}_i)_j - (\mathbf{z}_{\mathcal{X}})_j|^2}{nn'}\frac{n-n'}{n-1}\right) \text{ by Lemma 30} \\
&= \frac{1}{nn'}\frac{n-n'}{n-1}\sum_{i=1}^{n}\|\mathbf{\Phi}(\mathbf{x}_i) - \mathbf{z}_{\mathcal{X}}\|_2^2 \\
\boxed{\mathbf{E}_{\mathbf{g}\sim\mathrm{WOR}(n,n')}\|\mathbf{e}_{n'}\|^2 = \frac{1}{n-1}\Big(\frac{n}{n'}-1\Big)\left(\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{\Phi}(\mathbf{x}_i)\|_2^2 - \|\mathbf{z}_{\mathcal{X}}\|_2^2\right)}
\end{aligned}$$

Taking the expectation with respect to the draw of $\mathcal{X}$ as well we obtain

$$\begin{aligned}
\mathbf{E}_{\mathcal{X}}\mathbf{E}_{\mathbf{g}\sim\mathrm{WOR}(n,n')}\|\mathbf{e}_{n'}\|^2 &= \frac{1}{n-1}\Big(\frac{n}{n'}-1\Big)\big(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2 - \mathbf{E}_{\mathcal{X}}\|\mathbf{z}_{\mathcal{X}}\|_2^2\big) \\
&= \frac{1}{n-1}\Big(\frac{n}{n'}-1\Big)\left(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2 - \big(\|\mathbf{z}\|_2^2 + \sum_{j=1}^{m}\mathrm{Var}((\mathbf{z}_{\mathcal{X}})_j)\big)\right) \\
&= \frac{1}{n-1}\Big(\frac{n}{n'}-1\Big)\left(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2 - \big(\|\mathbf{z}\|_2^2 + \frac{1}{n}\big(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|^2 - \|\mathbf{z}\|^2\big)\big)\right) \\
&= \frac{1}{n-1}\Big(\frac{n}{n'}-1\Big)(1-\frac{1}{n})\big(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2 - \|\mathbf{z}\|_2^2\big)
\end{aligned}$$

$$\boxed{\mathbf{E}_{\mathcal{X}}\mathbf{E}_{\mathbf{g}\sim\mathrm{WOR}(n,n')}\|\mathbf{e}_{n'}\|^2 = \frac{1}{n}\Big(\frac{n}{n'}-1\Big)\big(\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2 - \|\mathbf{z}\|_2^2\big)}$$

- *i.i.d. Bernoulli sampling with parameter β:*

$$\mathbf{E}_{\mathbf{g}\sim\mathrm{Bern}(\beta)^n}\|\mathbf{e}_{n'}\|^2 = \sum_{j=1}^{m}\mathrm{Var}_{\mathbf{g}}\left(\frac{1}{\beta n}\sum_{i=1}^{n} g_i\mathbf{\Phi}(\mathbf{x}_i)_j\right) = \frac{1}{\beta^2 n^2}\sum_{j=1}^{m}\sum_{i=1}^{n}|\mathbf{\Phi}(\mathbf{x}_i)_j|^2\mathrm{Var}_{\mathbf{g}}(g_i)$$

$$\boxed{\mathbf{E}_{\mathbf{g}\sim\mathrm{Bern}(\beta)^n}\|\mathbf{e}_{n'}\|^2 = \frac{1}{n}\Big(\frac{1}{\beta}-1\Big)\left(\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{\Phi}(\mathbf{x}_i)\|^2\right)}$$

Taking the expectation with respect to the draw of $\mathcal{X}$ as well we obtain

$$\boxed{\mathbf{E}_{\mathcal{X}}\mathbf{E}_{\mathbf{g}\sim\mathrm{Bern}(\beta)^n}\|\mathbf{e}_{n'}\|^2 = \frac{1}{n}\Big(\frac{1}{\beta}-1\Big)\mathbf{E}_{\mathbf{x}}\|\mathbf{\Phi}(\mathbf{x})\|_2^2}$$

Frequencies subsampling. We define the noise contribution due to frequency subsampling as

$$\begin{aligned}
\mathbf{e}_H &= \frac{1}{n'}\big(\mathbf{\Sigma}_{H,n'}(\mathcal{X}) - \mathbf{\Sigma}_{n'}(\mathcal{X})\big) \\
\text{where:}\quad \mathbf{\Sigma}_{H,n'} &= \frac{m}{r}\sum_{i=1}^{n} g_i(\mathbf{\Phi}(\mathbf{x}_i)\odot\mathbf{h}_i) \\
\mathbf{\Sigma}_{n'} &= \sum_{i=1}^{n} g_i\mathbf{\Phi}(\mathbf{x}_i) \\
\mathbf{E}_H\|\mathbf{e}_H\|_2^2 &= \frac{1}{n'}\Big(\frac{m}{r}-1\Big)\frac{1}{n'}\sum_{i=1}^{n} g_i\|\mathbf{\Phi}(\mathbf{x}_i)\|^2
\end{aligned}$$

Recall that the masks entries are in $\{0,1\}$, thus $\forall i,j\,(\mathbf{h}_i)_j^2 = (\mathbf{h}_i)_j$, but also $\forall j\,\mathbf{E}_{\mathbf{h}\sim p_{\mathbf{h}}}\mathbf{h}_j = r/m$ because $p_{\mathbf{h}}\in\mathcal{P}_\alpha$. Therefore we have

$$\mathrm{Var}(\mathbf{h}_i)_j = \mathbf{E}\,((\mathbf{h}_i)_j)^2 - (\mathbf{E}\,(\mathbf{h}_i)_j)^2 = \mathbf{E}\,(\mathbf{h}_i)_j - \Big(\frac{r}{m}\Big)^2 = \frac{r}{m}(1-\frac{r}{m})$$

As a result

$$
\begin{aligned}
|n'|^2 \mathbf{E}_H \|\mathbf{e}_H\|_2^2 &= \mathbf{E}_H \left\| \frac{m}{r} \sum_{i=1}^n g_i (\mathbf{\Phi}(\mathbf{x}_i) \odot \mathbf{h}_i) - \sum_{i=1}^n g_i \mathbf{\Phi}(\mathbf{x}_i) \right\|_2^2 \\
&= \sum_{j=1}^m \mathrm{Var}_H \left(\frac{m}{r} \sum_{i=1}^n g_i \mathbf{\Phi}(\mathbf{x}_i)_j (\mathbf{h}_i)_j \right) \\
&= \left(\frac{m}{r}\right)^2 \sum_{j=1}^m \sum_{i=1}^n g_i |\mathbf{\Phi}(\mathbf{x}_i)_j|^2 \mathrm{Var}\big((\mathbf{h}_i)_j\big) \\
\mathbf{E}_H \|\mathbf{e}_H\|_2^2 &= \frac{1}{n'} \Big(\frac{m}{r} - 1 \Big) \frac{1}{n'} \sum_{i=1}^n g_i \|\mathbf{\Phi}(\mathbf{x}_i)\|^2 \\
\boxed{\mathbf{E}_{\mathbf{g}} \mathbf{E}_H \|\mathbf{e}_H\|_2^2 &= \frac{1}{n'} \Big(\frac{m}{r} - 1 \Big) \frac{1}{n} \sum_{i=1}^n \|\mathbf{\Phi}(\mathbf{x}_i)\|^2.}
\end{aligned}
$$

Taking the expectation w.r.t. the dataset, we get

$$
\boxed{\mathbf{E}_{\mathcal{X}} \mathbf{E}_{\mathbf{g}} \mathbf{E}_H \|\mathbf{e}_H\|_2^2 = \frac{1}{n'} \Big(\frac{m}{r} - 1 \Big) \mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|^2.}
$$

Total noise level For conciseness, we use the notation $\beta = n'/n$ when sampling n' samples without replacement, and $\alpha = r/m$. The total noise level for Poisson sampling is

$$
\begin{aligned}
\text{ref. } \mathbf{z}\text{: } \mathbf{E}_{\mathcal{X},\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}\|_2^2 &= \mathbf{E}_{\mathcal{X},\mathbf{g},H,\boldsymbol{\xi}} \left(\|\mathbf{e}_{\mathcal{X}}\|_2^2 + \|\mathbf{e}_{\xi}\|_2^2 + \|\mathbf{e}_{n'}\|_2^2 + \|\mathbf{e}_H\|_2^2 \right) \\
&= \frac{1}{n} \big(\mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|^2 - \|\mathbf{z}\|^2 \big) + \frac{m}{(n')^2} \sigma_\xi^2 + \frac{1}{n} \Big(\frac{1}{\beta} - 1 \Big) \mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|_2^2 + \frac{1}{n'} \Big(\frac{1}{\alpha} - 1 \Big) \mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|^2 \\
&= \frac{1}{\beta n} \frac{1}{\alpha} \mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|^2 - \frac{1}{n} \|\mathbf{z}\|^2 + \frac{m}{n^2 \beta^2} \sigma_\xi^2 \\
\text{ref. } \mathbf{z}_{\mathcal{X}}\text{: } \mathbf{E}_{\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}\|_2^2 &= \mathbf{E}_{\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}_{\xi}\|_2^2 + \|\mathbf{e}_{n'}\|_2^2 + \|\mathbf{e}_H\|_2^2 \\
&= \frac{m}{n^2 \beta^2} \sigma_\xi^2 + \frac{1}{n} \Big(\frac{1}{\beta} \frac{1}{\alpha} - 1 \Big) \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{\Phi}(\mathbf{x}_i)\|^2 \right)
\end{aligned}
$$

For WOR sampling, we get

$$
\begin{aligned}
\text{ref. } \mathbf{z}\text{: } \mathbf{E}_{\mathcal{X},\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}\|_2^2 &= \mathbf{E}_{\mathcal{X},\mathbf{g},H,\boldsymbol{\xi}} \left(\|\mathbf{e}_{\mathcal{X}}\|_2^2 + \|\mathbf{e}_{\xi}\|_2^2 + \|\mathbf{e}_{n'}\|_2^2 + \|\mathbf{e}_H\|_2^2 \right) \\
&= \frac{1}{n'} \frac{1}{\alpha} \mathbf{E}_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{x})\|^2 - \frac{1}{n'} \|\mathbf{z}\|^2 + \frac{m}{(n')^2} \sigma_\xi^2 \\
\text{ref. } \mathbf{z}_{\mathcal{X}}\text{: } \mathbf{E}_{\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}\|_2^2 &= \mathbf{E}_{\mathbf{g},H,\boldsymbol{\xi}} \|\mathbf{e}_{\xi}\|_2^2 + \|\mathbf{e}_{n'}\|_2^2 + \|\mathbf{e}_H\|_2^2 \\
&= \frac{m}{(n')^2} \sigma_\xi^2 + \frac{1}{n-1} \Big(\frac{1}{\beta} - 1 \Big) \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{\Phi}(\mathbf{x}_i)\|_2^2 - \|\mathbf{z}_{\mathcal{X}}\|_2^2 \right) + \frac{1}{n'} \Big(\frac{1}{\alpha} - 1 \Big) \frac{1}{n} \sum_{i=1}^n \|\mathbf{\Phi}(\mathbf{x}_i)\|^2 \\
&= \frac{m}{n^2 \beta^2} \sigma_\xi^2 + \left(\frac{1}{n-1} \Big(\frac{1}{\beta} - 1 \Big) + \frac{1}{\beta n} \Big(\frac{1}{\alpha} - 1 \Big) \right) \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{\Phi}(\mathbf{x}_i)\|_2^2 \right) \\
&\quad - \frac{1}{n-1} \Big(\frac{1}{\beta} - 1 \Big) \|\mathbf{z}_{\mathcal{X}}\|_2^2
\end{aligned}
$$

□

Proof of Lemma 27. We rewrite $\mathbf{s}(\mathcal{X}) = f(|\mathcal{X}| + \zeta) \widetilde{\mathbf{\Sigma}}(\mathcal{X})$, where $f(|\mathcal{X}| + \zeta)$ is an estimator of $1/|\mathcal{X}|$. We define the reference signal as $\mathbf{s} = \mathbf{z}$ or $\mathbf{z}_{\mathcal{X}}$, and the noise as $\mathbf{e} = f(|\mathcal{X}| + \zeta) \widetilde{\mathbf{\Sigma}}(\mathcal{X}) - \mathbf{s}$. In the following, the expectations are taken w.r.t. the randomness of the sketching mechanism when $\mathbf{z}_{\mathcal{X}}$ is chosen as the reference signal, and w.r.t. both the randomness of the mechanism and the draw of $\mathcal{X}$ when $\mathbf{z}$ is the

reference signal. We have $\mathbf{E}\mathbf{e} = 0$.

$$\begin{aligned}\mathbf{E}\,\|\mathbf{e}\|_2^2 &= \sum_{j=1}^{m} \mathrm{Var}(\mathbf{e}_j) = \sum_{j=1}^{m} \mathrm{Var}\Big(f(|\mathcal{X}| + \zeta)\widetilde{\mathbf{\Sigma}}(\mathcal{X})_j\Big) \\ &= \sum_{j=1}^{m} \left[\mathbf{E}(f^2)\mathrm{Var}\Big(\widetilde{\mathbf{\Sigma}}(\mathcal{X})_j\Big) + \mathrm{Var}(f)|\mathbf{s}_j|^2 n^2\right] \\ &= \mathbf{E}(f^2)n^2\mathbf{E}\|\widetilde{\mathbf{\Sigma}}(\mathcal{X})/n - \mathbf{s}\|_2^2 + \mathrm{Var}(f)\|\mathbf{s}\|^2 n^2\end{aligned}$$

Thus the noise-to-signal ratio NSR_ζ of the whole mechanism (including noise ζ) can be written as a function of the noise-to-signal ratio of $\widetilde{\mathbf{\Sigma}}(\mathcal{X})/n$ as computed in Lemma 26 (i.e. using the same parameters but without ζ), which we denote simply NSR in the rest of the proof.

$$\begin{aligned}\mathrm{NSR}_\zeta &= \mathbf{E}(f^2)n^2\mathrm{NSR} + \mathrm{Var}(f)n^2 \\ &= ((\mathbf{E}f)^2 + \mathrm{Var}(f))n^2\mathrm{NSR} + (\mathbf{E}f)^2 n^2 \frac{\mathrm{Var}(f)}{(\mathbf{E}f)^2} \\ &= (\mathbf{E}f)^2 n^2 \left[\left(1 + \frac{\mathrm{Var}(f)}{(\mathbf{E}f)^2}\right)(\mathrm{NSR} + 1) - 1\right]. \end{aligned} \tag{35}$$

For an unbiased estimator f (if there exists any), we have $(\mathbf{E}f)^2 = 1/n^2$ and the variance can be bounded via a Cramer-Rao bound.

A bound on the variance of f. Remember that ζ is drawn as $\zeta \sim \mathcal{L}(0, b)$. We want to estimate $\theta = 1/n$ from an observation x drawn with probability density (and log-density)

$$\begin{aligned}p_\theta(x) &= \frac{1}{2b_\zeta} e^{-\frac{|x - \frac{1}{\theta}|}{b_\zeta}} \\ \log(p_\theta(x)) &= -\log(2b_\zeta) - \frac{1}{b_\zeta}|x - \frac{1}{\theta}|.\end{aligned}$$

Using the Cramer-Rao bound for an unbiased estimator f, we have

$$\begin{aligned}\mathrm{Var}(f) &\geq \mathbf{E}\left[\left(\frac{d(\log p_\theta(x))}{d\theta}\right)^2\right]^{-1} = \left[\int_x \left(\frac{\mathrm{sign}(\frac{1}{\theta} - x)}{b_\zeta \theta^2}\right)^2 p_\theta(x)dx\right]^{-1} \\ &= \left[\int_x \left(\frac{1}{b_\zeta\theta^2}\right)^2 p_\theta(x)dx\right]^{-1} = b_\zeta^2\theta^4 = b_\zeta^2/n^4 = \sigma_\zeta^2/(2n^4) = (\mathbf{E}f)^2\sigma_\zeta^2/(2n^2).\end{aligned}$$

Conclusion Combining this bound with Equation (35) yields for an unbiased estimator of minimal variance (if there exists any)

$$\mathrm{NSR}_\zeta \geq \left(1 + \frac{\sigma_\zeta^2}{2n^2}\right)(\mathrm{NSR} + 1) - 1.$$

□

E Heuristic for Splitting the Privacy Budget

Proof of Lemma 28. The noise level for ζ is $b_\zeta = 1/\varepsilon_\zeta = 1/((1-\gamma)\varepsilon)$ for Laplacian noise according to Lemma 4. In the Laplacian-UDP setting, the lowest noise level yielding ε-DP is $\sigma_\xi = 2b = 2\sqrt{2}m/(\gamma\varepsilon)$ (complex Laplace distribution). We then have

$$\mathrm{NSR}_*^{\mathrm{RFF}} = \left(1 + \frac{1}{n^2(1-\gamma)^2\varepsilon^2}\right)\left(1 - \frac{1}{n} + \frac{m}{n\|\mathbf{z}\|^2}\left(\frac{1}{\alpha} + \frac{1}{n}\frac{8m^2}{\gamma^2\varepsilon^2}\right)\right) - 1,$$

For succinctness in the derivation, denote $A = 1/(n^2\varepsilon^2)$, $B = 1 - 1/n + m^2/(nr\|\mathbf{z}\|^2)$ and $C = \frac{1}{n^2\|\mathbf{z}\|^2}\frac{8m^3}{\varepsilon^2}$, so that we try to minimize

$$\mathrm{NSR}_*^{\mathrm{RFF}} = \left(1 + \frac{A}{(1-\gamma)^2}\right)\left(B + \frac{C}{\gamma^2}\right) - 1$$

Note that $\mathrm{NSR}_*^{\mathrm{RFF}}$ diverges to $+\infty$ when $\gamma \to 0_+$ or $\gamma \to 1_-$, but is continuous on $]0,1[$. Any minimizer on $]0,1[$ must cancel the quantity

$$\begin{aligned}\frac{1}{2C}\gamma^3(1-\gamma)^3\frac{d\mathrm{NSR}_*^{\mathrm{RFF}}}{d\gamma}(\gamma) &= \frac{1}{C}A\gamma^3\left(B + \frac{C}{\gamma^2}\right) - \frac{1}{C}C(1-\gamma)^3\left(1 + \frac{A}{(1-\gamma)^2}\right)\\ &= AB/C\gamma^3 + A\gamma - (1-\gamma)^3 - A(1-\gamma)\\ &= (AB/C+1)\gamma^3 - 3\gamma^2 + (2A+3)\gamma - (A+1)\end{aligned}$$

where $AB/C = \left(1 - \frac{1}{n} + \frac{m^2}{nr\|\mathbf{z}\|^2}\right)\frac{\|\mathbf{z}\|^2}{m}\frac{1}{8m^2} \ll 1$. Note that, if we start from the expression of the NSR which takes $\mathbf{z}_{\mathcal{X}}$ as a reference signal, we would get $B = 1-1/n+m^2/(nr\|\mathbf{z}\|^2)$, but the same approximation would still hold. The only real root of $\gamma^3-3\gamma^2+(2A+3)\gamma-(A+1)$ can be computed as $\gamma^* = 1-\frac{1}{3}E+\frac{2A}{E} = 1 + \frac{6A-E^2}{3E}$, where

$$E = \frac{1}{2^{1/3}}\left(27A + 3\sqrt{81A^2 + 96A^3}\right)^{1/3}$$

In this setting where $\varepsilon \ll 1/n$, $A \gg 1$ and we can use the following approximation.

$$\gamma^* = 1 + \frac{6A - E^2}{3E} \approx 1 + \frac{6A - 6A\left(1 + \frac{27A}{3\sqrt{96}A^{3/2}}\right)^{2/3}}{\sqrt{6}A^{1/2}} \approx \frac{1}{2}.$$

On the other side, if $A \ll 1$, we get

$$E \approx 3A^{1/3} \text{ and } \gamma^* = 1 + \frac{6A - E^2}{3E} \approx 1 + \frac{6A - 9A^{2/3}}{9A^{1/3}} \approx 1 - A^{1/3}.$$

□