Weighted Fourier Factorizations: Optimal Gaussian Noise for Differentially Private Marginal and Product Queries

CHRISTIAN JANOS LEBEDA, Inria, Idesp, Inserm, University of Montpellier, France
ALEKSANDAR NIKOLOV, University of Toronto, Canada
HAOHUA TANG, University of Toronto, Canada

We revisit the task of releasing marginal queries under differential privacy with additive (correlated) Gaussian noise. We first give a construction for answering arbitrary workloads of weighted marginal queries, over arbitrary domains. Our technique is based on releasing queries in the Fourier basis with independent noise with carefully calibrated variances, and reconstructing the marginal query answers using the inverse Fourier transform. We show that our algorithm, which is a factorization mechanism, is exactly optimal among all factorization mechanisms, both for minimizing the sum of weighted noise variances, and for minimizing the maximum noise variance. Unlike algorithms based on optimizing over all factorization mechanisms via semidefinite programming, our mechanism runs in time polynomial in the dataset and the output size. This construction recovers results of Xiao et al. [Neurips 2023] with a simpler algorithm and optimality proof, and a better running time.

We then extend our approach to a generalization of marginals which we refer to as product queries. We show that our algorithm is still exactly optimal for this more general class of queries. Finally, we show how to embed extended marginal queries, which allow using a threshold predicate on numerical attributes, into product queries. We show that our mechanism is *almost* optimal among all factorization mechanisms for extended marginals, in the sense that it achieves the optimal (maximum or average) noise variance up to lower order terms.

CCS Concepts: • **Security and privacy → Privacy-preserving protocols**.

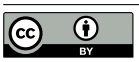
Additional Key Words and Phrases: Differential Privacy, Factorization Mechanism, Marginal Queries

ACM Reference Format:
Christian Janos Lebeda, Aleksandar Nikolov, and Haohua Tang. 2026. Weighted Fourier Factorizations: Optimal Gaussian Noise for Differentially Private Marginal and Product Queries. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 123 (May 2026), 23 pages. https://doi.org/10.1145/3801919

1 Introduction

In this work we study marginal queries and generalizations under differential privacy. We consider datasets D in which each data point is specified by d categorical attributes (we discuss our generalization to numerical queries later). A marginal query is given by a subset S of the attributes, and asks, for each possible setting t of the attributes in S, for the number of data points in D that have attributes in S with values agreeing with t. For example, consider a dataset that tracks sex, education level, place of residence, marital status, presence or absence of some genetic markers, and whether a person has been diagnosed with a certain disease. Then the answer to a marginal query corresponding to the sex attribute, one of the genetic marker attributes, and the disease

Authors' Contact Information: Christian Janos Lebeda, Inria, Idesp, Inserm, University of Montpellier, PreMeDICal, France, christian-janos.lebeda@inria.fr; Aleksandar Nikolov, Department of Computer Science, University of Toronto, Canada, sasho.nikolov@utoronto.ca; Haohua Tang, Department of Computer Science, University of Toronto, Canada, haohua.tang@mail.utoronto.ca.



ACM 2836-6573/2026/5-ART123
https://doi.org/10.1145/3801919

diagnosis attribute, is a 3-dimensional table with $2 \times 2 \times 2$ cells: one for each setting of these three attributes. The cells of the table give the number of males that have the genetic marker and have been diagnosed with the disease, the number of females that have the genetic marker and have been diagnosed, the number of males that don't have the marker and have been diagnosed, etc.

Marginal queries like these are known by different names, e.g., OLAP data cubes, and contingency tables, and are ubiquitous when summarizing high-dimensional data, including data from surveys, clinical studies, and official statistics. Often, however, the underlying data is sensitive, and protecting its privacy is sometimes even mandated by law. In these situations, releasing marginal queries can raise significant privacy concerns. Answers to a rich enough set of marginal queries can reveal enough about a dataset to enable an adversary to reconstruct most of the data [19], or to infer the membership of a given data point in the dataset [3]. For this reason, we adopt the differential privacy framework [9], and study marginal query release subject to the constraints of this framework.

A concrete motivating example to keep in mind are the marginal queries released by the US Census Bureau for the 2020 Census of Housing and Population. After it was discovered that prior disclosure avoidance systems failed to adequately protect the privacy of the census data [13, 18, 34], the US Census Bureau implemented a new differentially private algorithm, the TopDown algorithm, for the 2020 Census [1]. The TopDown algorithm releases estimates of selected marginal queries partitioned based on a geographical hierarchical structure. Since the accuracy of certain marginal queries has high practical importance, additional privacy budget is allocated for those estimates. We similarly allow the user to specify an importance weight for each marginal query.

As another motivating application, marginal queries have also been used as a subroutine when generating synthetic data [27, 28, 38]. Those techniques typically select a set of marginals that are then privately estimated using Gaussian noise. The synthetic data is generated so that it roughly matches the measured marginals.

In some applications it is also natural to consider more complex extensions of marginal queries. A particularly natural example are the extended marginal queries (called prefix-marginals in [26]), in which the attributes are partitioned into categorical and numerical, and, for a set S of attributes, the extended marginal query asks, for each possible setting of the categorical attributes in S, and each possible choice of prefix intervals for the numerical attributes in S, how many data points agree with the settings of the categorical attributes and have numerical attribute values lying in the chosen prefix intervals. Extended marginals allow us, for example, to ask how many data points correspond to males under the age of 35, or to females under the age of 45, etc. Workloads of extended marginals also appear in products released by the US Census Bureau, see examples described by McKenna et al. [26].

Marginal query release under differential privacy has been studied extensively since differential privacy was first introduced. An exhaustive account of this line of work is beyond the scope of this paper, but we highlight the results most relevant to our contributions. The early work of Barak, Chaudhuri, Dwork, Kale, McSherry, and Talwar [2], which initiated the formal study of differentially private marginal query release, considered binary attributes and proposed a method to release answers to marginal queries by adding Laplace noise to a workload of Fourier queries, i.e., queries that compute the Fourier transform of the empirical distribution of the data. They further showed how to make the query answers consistent with some real dataset. Yaroslavtsev, Cormode, Procopiuc, and Srivastava improved the error bounds of Barak et al. by adding noise with non-uniform variances, and gave a method to optimize over the choice of variances of the noise random variables added to the Fourier queries [37]. Other related work showed lower bounds on the minimum error necessary to answer marginal queries under differential privacy [3, 19], and investigated the trade-offs between privacy, accuracy, and computational complexity in answering marginal queries [10, 32, 33]. McKenna, Miklau, Hay, and Machanavajjhala proposed an efficient method to

approximately optimize over a class of private algorithms (called factorization mechanisms) for answering marginal queries [26]. In particular, they consider algorithms that first compute private estimates of other marginal queries, and then reconstruct answers to the marginal queries that were originally asked. Xiao, He, Zhang, and Kifer [36] gave an explicit algorithm for answering marginal queries which is efficient and optimal over factorization mechanisms. Concurrent to our work, the same authors, joined by Toksoz and Ding, extended their technique to support more general queries, including extended marginal queries, but did not show any optimality results for this extension [35]. We discuss their work and its relation to ours in more detail below.

1.1 Problem Setup

We consider a dataset $D := (x^{(1)}, \ldots, x^{(n)})$, consisting of a sequence of n points from a data universe $\mathcal{U}$. Each data point has d attributes, where the i-th attribute is drawn from the set $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$.[1] Marginal queries are then specified by sets $S \subseteq [d]$, and partial assignments $t \in \mathcal{U}_S := \prod_{i \in S} \mathcal{U}_i$. I.e., t has a value $t_i \in \mathcal{U}_i$ for each $i \in S$, and specifies an assignment to each attribute in S. Then the marginal queries are given by $q_{S,t}(D)$, equal to the number of data points $x^{(i)}$ in D agreeing with t on S, i.e., $q_{S,t}(D) := |\{i : x_j^{(i)} = t_j \; \forall j \in S\}|$. A marginal query workload $Q_{\mathcal{S}}$ is then specified by a collection of sets of attributes $\mathcal{S}$. We always assume that, for each $S \in \mathcal{S}$, all possible marginal queries $q_{S,t}(D)$ for all $t \in \mathcal{U}_S$ are asked. I.e., we assume that, for each $S \in \mathcal{S}$, we need to privately estimate the full table of marginals corresponding to S.

To generalize this set-up to extended marginals, we partition the set of attributes $[d]$ into the categorical attributes C, and the numerical attributes N. We can now redefine $q_{S,t}(D)$ to equal

$$q_{S,t}(D) := |\{i : x_j^{(i)} = t_j \; \forall j \in S \cap C, x_j^{(i)} \leq t_j \; \forall j \in S \cap N\}|.$$

Once again, an extended marginal query workload is specified by a collection of sets of attributes $\mathcal{S}$, and we assume that, for each $S \in \mathcal{S}$, all possible queries $q_{S,t}(D)$ for all $t \in \mathcal{U}_S$ are included in the workload.

Informally, differential privacy requires that the randomized algorithm $\mathcal{A}$ that takes as input a private dataset D, and outputs (approximate) answers to the queries in $Q_{\mathcal{S}}$, has the property that the probability distribution on outputs of $\mathcal{A}(D)$ is similar to the probability distribution on outputs of $\mathcal{A}(D')$ for any D' that is neighboring to D. In our work, we adopt the add/remove notion of neighboring, i.e., D and D' are neighboring (denoted $D \sim D'$) if and only if we can get D' from D by adding or removing at most one data point from D. We refer to Section 2.2 for the formal definitions. Our mechanisms rely on the Gaussian mechanism, which answers queries with independent Gaussian noise scaled to sensitivity, as a basic primitive. We describe it and its properties in detail in Section 2.2.

1.2 Our Contributions, and Factorization Mechanisms

Weighted marginal workloads. Our first contribution is a mechanism that privately answers any workload of marginal queries over arbitrary finite domains. Motivated by applications, such as the US Census, in which different marginal queries have different importance, we allow the queries to be weighted, and optimize a weighted average of the noise variances. In particular, we assign a non-negative weight $p(S)$ to each attribute set S and measure error as the square root of the weighted sum of variances, where the variance of $q_{S,t}$ is scaled by $\frac{p(S)}{|\mathcal{U}_S|}$. In this normalization, the weight $p(S)$ of S is split among the $|\mathcal{U}_S|$ partial assignments t defining different marginal queries on S. We call this notion of error the weighted root mean squared error. This is similar to error measures considered in prior work, e.g., by McKenna et al. [26], and Xiao et al. [36].

[1] We can accommodate any finite set by mapping it bijectively to $\{0, \ldots, m_i - 1\}$ for some m_i.

Our algorithm is inspired by the Fourier theoretic approach of [2]. We first compute the Fourier coefficients of the empirical distribution of D, and add independent Gaussian noise to each resulting Fourier query, with the variance of the noise carefully chosen to minimize the total noise added. This step already achieves the privacy guarantees. We then use the inverse Fourier transform to reconstruct answers to the marginal queries from the noisy Fourier query estimates. The resulting algorithm is roughly as efficient as the baseline Gaussian mechanism, and has running time polynomial in the dataset size, the dimension d, and the number of queries. While this approach is similar to that of Barak et al. [2], we extend it to Gaussian noise, non-binary domains, and optimize it by choosing the noise variances non-uniformly. We discuss our approach in more detail in Section 1.3. The algorithm is described precisely as Algorithm 1, and its guarantees for weighted root mean squared error are given in Theorem 8.

Optimality among Factorization Mechanisms. This algorithm is an instance of the factorization mechanisms framework of Edmonds, Nikolov, and Ullman [12], which itself generalizes the matrix mechanism of Li, Miklau, Hay, McGregor, and Rastogi [23].[2] In general, a factorization mechanism "factors" the queries Q into strategy queries R and a reconstruction matrix L. The strategy queries should be linear, in the sense that we can write them as $R(D) = \sum_{i=1}^{n} R(x^{(i)})$. The true query answers are given by $LR(D)$. To use this factorization as a private mechanism, we release $R(D)$ using the Gaussian mechanism, and multiply the resulting private estimate by L. In the case of our algorithm, the strategy queries are given by the Fourier queries, and the reconstruction matrix is derived from the inverse Fourier transform.

Factorization mechanisms have been the subject of intense research, and can significantly improve over the Gaussian mechanism baseline in many settings [16, 17, 21, 24, 26, 36] (see also the recent survey [31]). The class of factorization mechanisms is also equivalent to mechanisms that add unbiased correlated Gaussian noise to the true query answers [30]. Furthermore, factorization mechanisms are known to achieve nearly optimal error among all differentially private mechanisms in several important settings [12, 29, 30]. For example, factorization algorithms are optimal up to absolute constants among unbiased differentially private estimates of the true query answers [30], and among all differentially private algorithms when the dataset size n is large enough [12]. Similar techniques have also been used to reduce error for hierarchical queries [5]. Nevertheless, in other settings biased and data-dependent algorithms may achieve smaller error [14].

As our next contribution, we show that our mechanism achieves *optimal weighted root mean squared error* among all factorization mechanisms for any marginal workload. More precisely, no factorization mechanism can achieve smaller weighted average noise variance. This result is given in Theorem 10.

Let us remark here that it is possible to optimize error (measured as any linear function of the noise variances) over all factorization mechanisms using semidefinite programming [12]. The resulting running times are polynomial in the number of queries $|Q_S|$ and the universe size $|\mathcal{U}|$. In the case of marginal queries, however, $|\mathcal{U}|$ is exponential in d or worse, and this running time is prohibitive. Our results, by contrast, achieve running times polynomial in d and give explicit factorizations.

Next, we consider the problem of minimizing the maximum variance over all queries in an arbitrary workload of marginals. Here we reuse our algorithm for weighted root mean squared error. We choose weights $p^*(S)$ for $S \in \mathcal{S}$ that sum to 1, and maximize the weighted root mean squared error. Since the resulting maximization problem is concave in the weights, we can solve it efficiently using standard methods. Once the $p^*(S)$ weights are computed, we simply run Algorithm 1. First

[2]It is common to refer to all factorization mechanisms as matrix mechanisms. We prefer the name factorization mechanisms, and reserve the name matrix mechanism for the original mechanisms proposed in [23].

order optimality conditions show that, for these weights, the weighted root mean squared error equals the maximum variance, as p^* is supported on queries for which the noise has maximum variance. Once again, the algorithm runs in polynomial time in n, d, and $|Q_{\mathcal{S}}|$, and it achieves *optimal maximum variance* among all factorization mechanisms. The guarantees of the algorithm are given in Lemma 9, and its optimality is proved in Theorem 10.

We note that there are known asymptotic expressions for the optimal error achievable on the workload of all k-way marginals over binary domains by either factorization [12, 30], or arbitrary differentially private mechanism [3]. By contrast, we are interested in exactly optimal factorizations and exact expressions for their error.

Extensions. We extend our technique to more expressive workloads, which we refer to as product queries. We associate a function $\phi_j : \mathcal{U}_j \to \mathbb{R}$ to each attribute $j \in [d]$ and define, for $S \subseteq [d]$ and $t \in \mathcal{U}_S$, the queries $q^{\phi}_{S,t}(D)$ as $\sum_{i=1}^{n} \prod_{j \in S} \phi_j(t_j - x_j^{(i)})$, where the difference is interpreted mod m_j. Using the same approach as for marginals, we give factorization mechanisms for arbitrary workloads of product queries, and prove their optimality among all factorization mechanisms with respect to weighted root mean squared error, and maximum variance. The mechanisms are roughly as computationally efficient as the ones for marginals. The upper bound is given as Theorem 12 and Theorem 14 in Appendix A, and the lower bound is in Theorem 17.

We then show that we can embed any workload of extended marginal queries into a workload of product queries, to which we can apply our mechanism (Theorem 16). This approach is similar to the design of explicit factorization mechanisms for prefix (i.e., threshold) queries by embedding into the "circulant queries", i.e., queries that count points in an interval on a circle [4, 16]. Indeed, prefix queries are a special case of extended marginals when there is only one numerical attribute, and our mechanism for this special case reduces to the factorization mechanism of Henzinger and Upadhyay [16]. We also show lower bounds for extended marginal queries that match the error of our mechanisms up to lower order terms, both for root mean squared error and maximum variance (Theorem 18). In particular, the lower and upper bounds converge to the same value as the minimum domain size of numerical attributes grows to infinity. For the special case of prefix queries, our lower bounds are slightly weaker than the best known lower bound [25], but only by an additive constant. At the same time they are significantly more general.

Comparison with ResidualPlanner. In closely related prior work, Xiao et al. [36] proposed a mechanism, ResidualPlanner, which efficiently computes private estimates for an arbitrary workload of marginals, and achieves optimal error among all factorization mechanisms for a class of error measures that includes weighted root mean squared error and maximum variance. The authors define a *subtraction matrix* and construct *residual queries* by combining subtraction matrices using the Kronecker product. These residual queries serve as the strategy queries for the mechanism. ResidualPlanner computes all residual queries required for the marginal workload, adds *correlated* Gaussian noise to each vector of residual query answers, and recovers marginal estimates by inverting the transformation.

To speed up processing time, ResidualPlanner represents matrices implicitly. The authors show how to optimize the privacy budget used for each residual query for each error measure in the class they consider. Notably, for weighted root mean squared error, they give closed form expressions for the optimal privacy budget allocation. The extended version of the paper [35] (which is concurrent with this work) introduces ResidualPlanner+, which supports more general workloads similar to our product queries, but requires an externally provided *strategy replacement matrix*, which is used to construct the subtraction matrix.

There is significant overlap between our results and [35, 36], but the techniques we use are significantly different. Next we go into more details about how our results compare with ResidualPlanner and ResidualPlanner+, and argue that our Fourier-theoretic approach offers some advantages. First, we improve over ResidualPlanner [36] in the following ways:

- Running time: Reconstructing estimates of marginal queries is the bottleneck for the running time of our mechanisms. [36] reconstruct estimates to a k-way marginal with domain $\mathcal{U}_S$ in time $O(k|\mathcal{U}_S|^2)$. We reconstruct estimates in time $O(|\mathcal{U}_S| \log(|\mathcal{U}_S|))$.
- Simplicity: We argue that our mechanism is simpler than ResidualPlanner. Our strategy queries compute Fourier coefficients, rather than using Kronecker products of custom matrices. Since our technique releases a number of sensitivity 1 queries, our privacy proofs are significantly simpler.
- Explicit factorization: A technical difference from our technique is that the noise added to the answers of a residual query in ResidualPlanner is correlated for non-binary data, which is not standard for the factorization framework. While ResidualPlanner can be expressed equivalently as a standard factorization mechanism, Xiao et al. state they avoid doing so because of the complexity of the re-formulation. By contrast, we show that our mechanisms can be expressed as standard factorization mechanisms in a straightforward way.
- Simpler lower bounds: Xiao et al. show that ResidualPlanner is optimal among all factorization mechanisms for a class of error measures via a technically involved symmetry argument. By contrast, we derive simple necessary and sufficient optimality conditions for *any* factorization mechanism. Verifying that our mechanism satisfies these optimality conditions is then straightforward.

We improve over ResidualPlanner+ for generalizations of marginals [35] in the following ways:

- External input: We give simple explicit algorithms for estimating any workload of product queries. ResidualPlanner+ relies on *strategy replacement* matrices as external input to the framework.
- Optimality results: ResidualPlanner+ provides no strong theoretical optimality guarantees for the error it achieves, beyond the ones already known for ResidualPlanner. The error they achieve is also heavily dependent on the externally provided strategy replacement matrices. In contrast, we give matching upper and lower bounds for product queries both for weighted root mean squared error and maximum variance, using the same technique we used for marginal queries. We also give a lower bound for extended marginal queries that matches the leading term of our upper bound. Here we use the singular value lower bound on the error of factorization mechanisms [22], and use insights from the lower bound for product queries in order to compute the necessary estimates of sums of singular values for extended marginals. Note that, by contrast, it is unclear how to adapt the symmetry argument used to show the optimality of ResidualPlanner to extended marginal queries.

1.3 Technical Intuition

Here we present the central idea behind our approach. For simplicity, we present the results for 1-GDP[3] and focus on estimating all 2-way marginal queries over binary attributes. In the end of this section we discuss how we generalize the technique to other settings. All results apply naturally to μ-GDP by scaling all noise samples by $1/\mu$.

[3]The definition of GDP is deferred to Section 2.2. In this section, we only use the fact that the Gaussian mechanism satisfies differential privacy, or, formally, 1-GDP, and that post-processing preserves it. The results hold for other variants of differential privacy as well (see Remark 1).

We privately estimate aggregate queries in the Fourier basis. In the case of binary attributes, the queries we are interested in are relatively simple. For all subsets of $[d]$ with at most 2 elements we answer aggregate queries of the form

$$F_\emptyset(D) = \sum_{i\in[|D|]} (-1)^0 = |D|\,; \quad F_{\{j\}}(D) = \sum_{i\in[|D|]} (-1)^{x_j^{(i)}}\,; \quad F_{\{j,k\}}(D) = \sum_{i\in[|D|]} (-1)^{x_j^{(i)}+x_k^{(i)}}\,.$$

These queries are, up to scaling, the Fourier coefficients up to level 2 of the empirical distribution of dataset points.

It is easy to see that each of these queries has sensitivity 1. We can thus release these queries privately by adding noise from $\mathcal{N}(0, \binom{d}{2} + d + 1)$ to each of them. However, we can do slightly better if we add less noise to the queries that we will reuse for multiple marginal queries later. Similarly to [21] whose mechanism handles the case of 1-way marginals, we scale the variance inversely proportional to the square root of the number of queries in which each Fourier coefficient appears. Specifically, for all $j, k \in [d]$ we release the following:

$$\sigma^2 = \binom{d}{2} + d\sqrt{d-1} + \sqrt{\binom{d}{2}} \qquad \tilde{F}_\emptyset(D) = F_\emptyset(D) + \mathcal{N}\left(0, \sigma^2/\sqrt{\binom{d}{2}}\right);$$

$$\tilde{F}_{\{j\}}(D) = F_{\{j\}}(D) + \mathcal{N}\left(0, \sigma^2/\sqrt{d-1}\right) \qquad \tilde{F}_{\{j,k\}}(D) = F_{\{j,k\}}(D) + \mathcal{N}(0, \sigma^2)\,.$$

We can release all the noisy queries above under our privacy constraints. We omit the proof here.

From these noisy estimates, we can recover unbiased estimates for the 2-way marginals as post-processing. Notice that for any $j \in [d]$, $t_j \in \{0,1\}$, and any $x \in \{0,1\}^d$,

$$\frac{1 + (-1)^{t_j}(-1)^{x_j}}{2} = \begin{cases} 1 & t_j = x_j \\ 0 & t_j \neq x_j \end{cases}.$$

Therefore, for any $j, k \in [d]$ and $t \in \{0,1\}^{\{j,k\}}$,

$$\begin{aligned} q_{\{j,k\},t}(x) &:= \mathbf{1}\{t_j = x_j\} \cdot \mathbf{1}\{t_k = x_k\} = \frac{1 + (-1)^{t_j}(-1)^{x_j}}{2} \cdot \frac{1 + (-1)^{t_k}(-1)^{x_k}}{2} \\ &= \frac{1}{4}(1 + (-1)^{t_j}(-1)^{x_j} + (-1)^{t_k}(-1)^{x_k} + (-1)^{t_j+t_k}(-1)^{x_j+x_k}) \\ &= \frac{1}{4}(F_\emptyset(x) + (-1)^{t_j}F_{\{j\}}(x) + (-1)^{t_k}F_{\{k\}}(x) + (-1)^{t_j+t_k}F_{\{j,k\}}(x))\,. \end{aligned}$$

As a result, a marginal query on the attributes $\{j,k\}$ with assignment $t \in \{0,1\}^{\{j,k\}}$ can be estimated by

$$\tilde{q}_{\{j,k\},t}(D) = \frac{1}{4}\left(\tilde{F}_\emptyset(D) + (-1)^{t_j} \cdot \tilde{F}_{\{j\}}(D) + (-1)^{t_k} \cdot \tilde{F}_{\{k\}}(D) + (-1)^{t_j+t_k} \cdot \tilde{F}_{\{j,k\}}(D)\right).$$

Then the error $\tilde{q}_{\{j,k\},t}(D) - q_{\{j,k\},t}(D)$ is distributed as a mean zero Gaussian with variance

$$\frac{1}{4^2\binom{d}{2}}\left(\binom{d}{2} + d\sqrt{d-1} + \sqrt{\binom{d}{2}}\right)^2.$$

The expression above approaches $\binom{d}{2}/16$ as d increases, where the main source of the error is the estimate of $\tilde{F}_{\{j,k\}}(D)$. By contrast, a straightforward application of the Gaussian mechanism would add noise with variance $\binom{d}{2}$ to each query. The approach above generalizes naturally to k-way marginals by estimating Fourier coefficients up to level k. Each marginal query for binary attributes can be recovered using 2^k Fourier queries, and the standard deviation of the error

approaches $2^{-k}\sqrt{\binom{d}{k}}$ as d increases. In the more general setting where an attribute has one of m values the Fourier coefficients are complex roots of unity, but the sensitivity is still bounded by 1. The improvement over i.i.d. noise is not as large as for binary attributes, since the marginals are not as correlated. Nevertheless, we still achieve an improvement and the standard deviation of the error approaches $(1-1/m)^k\sqrt{\binom{d}{k}}$.

Then, in Section 3 we consider much more general workloads, in which attributes can have different domain size and we assign a weight to the error of marginal queries for each subset of attributes. The algorithm is slightly more complicated in this setting, but the underlying idea is the same. We privately estimate the Fourier coefficients required to recover all non-zero weight marginals, and we allocate additional privacy budget to important coefficients that are reused for many marginals. Finally, in Section 4 we extend our technique to product queries, and to extended marginals. The underlying idea of the mechanism still remains the same, but the product queries affect the privacy budget allocation.

Organization

The remainder of the paper is organized as follows. In Section 2 we present the problem of privately estimating marginal queries and provide the relevant background. In Section 3 we present our mechanisms for privately releasing marginal queries and provide matching lower bounds. In Section 4 we discuss our results for product queries and extended marginals. We present some additional results for product queries and extended marginals in the appendix. Due to space limitations, we defer most proofs and some technical details to the extended version of the paper available online.

2 Preliminaries

2.1 Marginal Queries

We consider data points $x \in \mathcal{U}$ with d attributes where the i-th attribute has domain $\mathcal{U}_i := \{0, 1, \dots, m_i - 1\}$. In general, we have $\mathcal{U} := \prod_{i\in[d]} \mathcal{U}_i$. A commonly studied special case is binary attributes where we have $x \in \{0,1\}^d$.

A k-way marginal query is parameterized by a set $S \subseteq [d]$ where $|S| = k$ and an assignment $t \in \mathcal{U}_S := \prod_{i\in S} \mathcal{U}_i$. For a data point $x \in \mathcal{U}$ a marginal query $q_{S,t} : \mathcal{U} \to \{0,1\}$ evaluates to

$$q_{S,t}(x) = \begin{cases} 1 & \text{if } \forall i \in S : x_i = t_i, \\ 0 & \text{otherwise.} \end{cases}$$

We define a marginal query of a dataset D with n data points as $q_{S,t}(D) = \sum_{i=1}^{|D|} q_{S,t}(x^{(i)})$, where $x^{(i)}$ is the i-th data point in D for some arbitrary order. Our goal is to privately estimate the value of marginal queries for a dataset under differential privacy. Note that for any $S \subseteq [d]$, there are a total of $|\mathcal{U}_S| = \prod_{i\in S} m_i$ marginal queries, one for each possible assignment of attributes in S. We always assume that, once we choose S, we ask each possible marginal query for each possible setting of the attributes in S. This is standard practice both in research papers (e.g. [27, 28]), and in deployments of differential privacy (e.g. [1]). Specifically, we are given a workload $Q_{\mathcal{S}}$ of marginal queries defined by a collection $\mathcal{S}$ of subsets of $[d]$. For each subset $S \in \mathcal{S}$, the workload contains all queries $q_{S,t}$ for all $t \in \mathcal{U}_S := \prod_{i\in S} \mathcal{U}_i$. We use the notation $\mathcal{S}_\downarrow$ for the collection of all sets $R \subseteq [d]$ such that $R \subseteq S$ for some $S \in \mathcal{S}$. I.e., this is the closure of $\mathcal{S}$ under taking subsets.

Extensions. We extend our technique to a generalization of marginal queries that we call product queries. We define product queries by functions $\phi = (\phi_1, \dots, \phi_d)$, $\phi_j : \mathcal{U}_j \to \mathbb{R}$, and a query $q^{\phi}_{S,t}$ on

a single data point is defined by

$$q^{\phi}_{S,t}(x) \coloneqq \prod_{j \in S} \phi_j((t_j - x_j) \bmod m_j).$$

We recover standard marginals $q_{S,t}$ from product queries by setting $\phi_j(z) \coloneqq \mathbf{1}\{z = 0\}$.

We also consider an extension of marginals where attributes can be categorical or numerical. The queries for categorical attributes match standard marginal queries. For numerical attributes, we consider prefix or suffix predicates. We thus define $T_i \coloneqq \mathcal{U}_i$ for $i \in C$, and $T_i \coloneqq \{-m_i, \dots, 0, \dots, m_i - 1\}$ for $i \in N$, and $T_S \coloneqq \prod_{i \in S} T_i$, where C and N are the set of categorical and numerical attributes. We can now redefine the query $q_{S,t}$ for $S \subseteq [d]$ and $t \in T_S$ as

$$q_{S,t}(x) \coloneqq \left(\prod_{j \in S \cap C} \mathbf{1}\{x_j = t_j\} \right) \left(\prod_{\substack{j \in S \cap N \\ t_j \geq 0}} \mathbf{1}\{x_j \leq t_j\} \right) \left(\prod_{\substack{j \in S \cap N \\ t_j < 0}} \mathbf{1}\{x_j \geq |t_j|\} \right)$$

We show how to embed extended marginals as product queries in Appendix A.2.

2.2 Differential Privacy

Differential privacy [9] is a framework for preserving privacy by ensuring that the distributions of a mechanism for any pair of neighboring datasets are not too far apart. The difference between neighboring datasets, as defined below, corresponds to adding or removing all data about any individual data point. Several definitions of differential privacy exist, and our results apply to any variant satisfied by the Gaussian mechanism. We present our results using Gaussian Differential Privacy because it exactly describes the privacy guarantees of additive Gaussian noise. Informally, GDP ensures that the pair of distributions for neighboring datasets is at least as hard to distinguish as two normal distributions.

Definition 1 (Neighboring datasets)**.** *Two datasets D and D' are neighboring, denoted $D \sim D'$, if we can obtain one dataset of the pair by adding one data point to the other dataset.*

Definition 2 (Gaussian Differential Privacy [7, Definition 4])**.** *A randomized mechanism $\mathcal{M} \colon \mathcal{U}^* \to \mathcal{R}$ satisfies μ-GDP if for all pairs of neighboring datasets $D \sim D'$ it holds that*

$$T(\mathcal{M}(D), \mathcal{M}(D')) \geq T(\mathcal{N}(0,1), \mathcal{N}(\mu,1)),$$

where $T(P, Q) : [0,1] \to [0,1]$ denotes the trade-off function for two distributions P and Q defined on the same space. The tradeoff function is defined as

$$T(P,Q)(\alpha) = \inf\{\beta_\phi : \alpha_\phi \leq \alpha\},$$

where the infimum is taken over all (measurable) rejection rules ϕ, and α_ϕ and β_ϕ denote the type I and type II error rates, respectively.

Remark 1. We present our results in μ-GDP throughout the paper because this definition exactly matches the privacy properties of factorization mechanisms. The results hold under other common privacy definitions, such as zero Concentration Differential Privacy (ρ-zCDP) and Approximate Differential Privacy ((ε, δ)-DP). All mechanisms discussed in this paper satisfy ρ-zCDP for $\rho = \mu^2/2$ and $(\varepsilon, \delta(\varepsilon))$-DP for any $\delta(\varepsilon) > 0$ such that

$$\delta(\varepsilon) \geq \Phi\left(\frac{\mu}{2} - \frac{\varepsilon}{\mu}\right) - e^{\varepsilon} \Phi\left(-\frac{\mu}{2} - \frac{\varepsilon}{\mu}\right).$$

The Gaussian mechanism [6, 8, 11] is one of the most important tools in differential privacy. The mechanism achieves the desired privacy guarantees by adding unbiased Gaussian noise to all queries scaled by the ℓ_2 sensitivity. Applying the Gaussian mechanism directly to our setting gives us a baseline for privately estimating marginal queries.

Lemma 3 (The Gaussian mechanism). *Let $q\colon \mathcal{U}^* \to \mathbb{R}^d$ be a set of queries with ℓ_2 sensitivity $\Delta q \coloneqq \max_{D\sim D'} \|q(D)-q(D')\|_2$. Then the mechanism that outputs $q(X)+Z$ where $Z \sim \mathcal{N}\left(0, \frac{(\Delta q)^2}{\mu^2} I_d\right)$ satisfies μ-GDP.*

Lemma 4 (Gaussian noise for marginal queries). *Let $\mathcal{S} = (S_1, \dots, S_m)$ be a collection of sets such that $S_i \subseteq [d]$. Then the mechanism that for each $i \in [m]$ and each assignment $t \in \mathcal{U}_{S_i}$ independently samples noise $Z_{S_i,t} \sim \mathcal{N}(0, m/\mu^2)$ and releases $q_{S_i,t}(D) + Z_{S_i,t}$ satisfies μ-GDP.*

Proof. Notice that for each S_i, adding or removing a data point changes exactly one marginal query by 1 while the remaining $(\prod_{i\in S} |\mathcal{U}_i|) - 1$ queries are unaffected. The ℓ_2 sensitivity for all queries in $\mathcal{S}$ is thus $\sqrt{\sum_{i\in[m]} 1^2} = \sqrt{m}$. The privacy guarantee follows from Lemma 3. □

We use some standard properties of differential privacy in our proofs.

Lemma 5 (Post-processing [7, Proposition 4]). *Let $\mathcal{M}\colon \mathcal{U}^* \to \mathcal{R}$ denote any μ-GDP mechanism. Then for any (randomized) function $g\colon \mathcal{R} \to \mathcal{R}'$ the composed mechanism $g \circ \mathcal{M}\colon \mathcal{U}^* \to \mathcal{R}'$ also satisfies μ-GDP.*

Lemma 6 (Composition [7, Corollary 3.3]). *Let $\mathcal{M}_1\colon \mathcal{U}^* \to \mathcal{R}_1$ and $\mathcal{M}_2\colon \mathcal{U}^* \to \mathcal{R}_2$ denote a pair of mechanisms that satisfies μ_1-GDP and μ_2-GDP, respectively. Then the mechanism $\mathcal{M}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D))$ that outputs the result from both mechanisms satisfies $\sqrt{\mu_1^2 + \mu_2^2}$-GDP.*

The privacy properties of our mechanisms follow from the Gaussian mechanism which is typically defined for real-valued queries. We use a variant for complex-valued queries defined below. It is easy to show that the complex-valued Gaussian mechanism is equivalent to a real-valued Gaussian mechanism, see the extended version of the paper for details.

Lemma 7 (The complex Gaussian mechanism). *Let $q\colon \mathcal{U}^* \to \mathbb{C}^d$ be a set of queries with ℓ_2 sensitivity $\Delta q \coloneqq \max_{D\sim D'} \|q(D) - q(D')\|_2$. Then the mechanism that outputs $q(D) + Z$ where $Z \sim \mathcal{CN}\left(0, \frac{2(\Delta q)^2}{\mu^2} I_d\right)$ satisfies μ-GDP.*

We use $\omega_n \coloneqq e^{2\pi i/n} = \cos(2\pi/n) + i \cdot \sin(2\pi/n)$ to denote the primitive n-th root of unity. We refer to the extended version of the paper for preliminaries for complex numbers.

3 Optimal Factorization for Weighted Marginal Queries

In this section we introduce our technique for adding noise to marginal queries. We first show how the marginal queries can be recovered from aggregate queries in the Fourier basis of $\mathcal{U}$. This observation immediately yields a correlated Gaussian noise mechanism, or, equivalently, a factorization mechanism, that achieves noise with lower variance than the standard Gaussian mechanism. It is then easy to observe that some of the queries in the Fourier basis have a larger impact on the error than others. By allocating privacy budget weighted by the importance of the queries, we can further reduce the error of our factorization mechanism. In Section 3.3, we show that our mechanism is, in fact, optimal among all factorization mechanisms!

Due to space limitations we omit proofs from the main body. Full proofs and some additional results can be found in an extended version of the paper.

3.1 Marginal Queries in the Fourier Basis

Let $a \in \mathcal{U}$ be a choice of value for each attribute, and recall that $\omega_{m_i} := \exp(2\pi i/m_i)$ is a primitive m_i-th root of unity. Our algorithm relies on aggregate queries of the form

$$F_a(D) := \sum_{i=1}^{|D|} \overline{\chi_a(x^{(i)})} \text{ where } \chi_a(x) := \prod_{j=1}^{d} \omega_{m_j}^{a_j \cdot x_j}. \tag{1}$$

As in the example of binary domains from Section 1.3, these queries give, up to scaling, the Fourier coefficients of the empirical distribution of D. The functions χ_a are the Fourier characters, which form an orthogonal basis for functions on $\mathcal{U}$. We denote the support of a by $\text{supp}(a) := \{j : a_j \neq 0\}$. The size of the support, i.e., the weight of a, is denoted by $\|a\|_0$.

It is easy to see that adding or removing one data point from D always changes the value of any $F_a(D)$ by a unit complex number (see the extended version for preliminaries on roots of unity). We use this fact later to bound the sensitivity for privatizing the queries. Note that $\chi_0(x) = 1$ for all $x \in \mathcal{U}$, so $F_0(D) = |D|$ is the dataset size.

Next, we show how we recover marginal queries using the aggregate Fourier queries above using an inverse discrete Fourier transform. The main observation is that, for any $x \in \mathcal{U}$, $j \in [d]$, and $t_j \in \mathcal{U}_j$,

$$\frac{\sum_{a=0}^{m_j-1} \omega_{m_j}^{at_j} \omega_{m_j}^{-ax_j}}{m_j} = \frac{\sum_{a=0}^{m_j-1} \omega_{m_j}^{a(t_j-x_j)}}{m_j} = \begin{cases} 1 & t_j = x_j \\ 0 & t_j \neq x_j \end{cases},$$

and, therefore,

$$q_{S,t}(x) = \prod_{j \in S} \mathbf{1}\{t_j = x_j\} = \prod_{j \in S} \frac{\sum_{a=0}^{m_j-1} \omega_{m_j}^{at_j} \omega_{m_j}^{-ax_j}}{m_j} = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \text{supp}(a) \subseteq S}} \chi_a(t)\overline{\chi_a(x)}\,. \tag{2}$$

Above, we slightly abused notation: we used $\chi_a(t)$ even though t_j is defined only for $j \in S$. Note, nevertheless, that this is well defined since $\text{supp}(a) \subseteq S$, and $\chi_a(t)$ only depends on the coordinates of t in $\text{supp}(a)$.

Summing over dataset points now gives us

$$q_{S,t}(D) = \sum_{i=1}^{|D|} q_{S,t}(x^{(i)}) = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \text{supp}(a) \subseteq S}} \chi_a(t) F_a(D)\,. \tag{3}$$

3.2 Estimating Arbitrary Marginal Query Workloads

We now consider a general set up for answering arbitrary workloads of marginal queries using queries in the Fourier basis. Recall that a workload $Q_{\mathcal{S}}$ of marginal queries on the universe $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, where $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$, is defined by a collection $\mathcal{S}$ of subsets of $[d]$. For each subset $S \in \mathcal{S}$, the workload contains all queries $q_{S,t}$ for all $t \in \mathcal{U}_S := \prod_{i \in S} \mathcal{U}_i$.

We first consider an error metric which is a weighted version of root mean squared error. In this case, together with the workload $Q_{\mathcal{S}}$, we are also given a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$. Although it is not essential to our algorithm, we will assume the weights are normalized, i.e., $\sum_{S \in \mathcal{S}} p(S) = 1$. Then the goal is to compute estimates $\tilde{q}_{S,t}(D)$ for all $S \in \mathcal{S}$ and all $t \in \mathcal{U}_S$ that minimize the error

$$\text{err}_p(q, \tilde{q}) := \left(\sum_{S \in \mathcal{S}} \frac{p(S)}{|\mathcal{U}_S|} \sum_{t \in \mathcal{U}_S} \mathbb{E}[(\tilde{q}_{S,t}(D) - q_{S,t}(D))^2] \right)^{1/2}.$$

Notice that, in this definition, the weight $p(S)$ of a set of attributes S is evenly split between the $\mathcal{U}_S$ marginal queries corresponding to S. That is, we assume that the estimates for all assignments of S have equal importance. This assumption aligns with practical deployments such as the US Census [1], and was also made in prior work [26, 36].

We now discuss our approach for estimating weighted marginal queries in the Fourier basis. We privately estimate all Fourier aggregate queries F_a necessary for reconstructing the answers to queries in $Q_{\mathcal{S}}$. Notice in Equation (3) that any query $F_a(D)$ is used to estimate $q_{S,t}(D)$ for any set of attributes S that contains $\text{supp}(a)$. Rather than privately estimating $F_a(D)$ separately for each such S, we can, of course, reuse the estimate. Reusing values already gives us a slight improvement over the Gaussian mechanism baseline. However, we can further reduce the error by releasing more accurate answers to Fourier coefficient that are reused a lot. The amount of noise added to F_a is proportional to how much this noise contributes to $\text{err}_p(q,\tilde{q})^2$ in expectation. As a final step, we always remove any imaginary part of the estimate, since that must come from noise. The full description is given in Algorithm 1. In Section 3.3 we also show that this choice of noise magnitudes gives an optimal factorization for $Q_{\mathcal{S}}$. Below we present some intuition behind the design of our algorithm and then present its privacy and error guarantees.

Algorithm 1 Differentially private estimate of a workload $Q_{\mathcal{S}}$ of weighted marginals.

For any $a \in \mathcal{U}$, let

$$\tau_a := \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \text{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2}}, \qquad \text{and} \qquad \tau := \frac{1}{\mu^2}\sum_{a\in\mathcal{U}} \tau_a.$$

For any a such that $\tau_a > 0$, privately release an estimate of $F_a(D)$ (see Equation (1)) by adding independent noise such that

$$\tilde{F}_a(D) = F_a(D) + Z_a, \text{ where } Z_a \sim \mathcal{CN}\left(0, \frac{2\tau}{\tau_a}\right).$$

Estimates for marginal queries for $S \in \mathcal{S}$ can be recovered using post-processing by computing

$$\tilde{q}_{S,t}(D) = \text{Re}\left(\frac{1}{|\mathcal{U}_S|}\sum_{\substack{a\in\mathcal{U}\\ \text{supp}(a)\subseteq S}} \chi_a(t)\tilde{F}_a(D)\right), \text{ where } \chi_a(t) = \prod_{i\in S}\omega_{m_i}^{a_i t_i}.$$

To gain intuition behind the choice of τ in Algorithm 1, note that each estimate's error $\tilde{q}_{S,t}(D) - q_{S,t}(D)$ is a sum of $|\mathcal{U}_S|$ random variables. Specifically, for each a where $\text{supp}(a) \subseteq S$, F_a contributes to the sum a random variable with variance

$$\mathbb{E}\left[\left(\text{Re}\left(\frac{1}{|\mathcal{U}_S|}\chi_a(t)(\tilde{F}_a(D) - F_a(D))\right)\right)^2\right] = \mathbb{E}\left[\left(\text{Re}\left(\frac{\tilde{F}_a(D) - F_a(D)}{|\mathcal{U}_S|}\right)\right)^2\right] = \frac{\sigma_a^2}{|\mathcal{U}_S|^2},$$

where σ_a^2 is half the variance of Z_a. Since the noise added to all F_a queries are independent, the variance of $\tilde{q}_{S,t}(D) - q_{S,t}(D)$ is simply the sum of the variances for each of the $|\mathcal{U}_S|$ random variables, i.e., $\sum_{a\in\mathcal{U}_S}\sigma_a^2/|\mathcal{U}_S|^2$. Summing the contribution of the estimate of $F_a(D)$ to the expectation of $\text{err}_p(q,\tilde{q})^2$ over all $S \supseteq \text{supp}(a)$, we have

$$\sum_{\substack{S\in\mathcal{S}\\ \text{supp}(a)\subseteq S}} \frac{p(S)}{|\mathcal{U}_S|}\sum_{t\in\mathcal{U}_S}\frac{\sigma_a^2}{|\mathcal{U}_S|^2} = \sigma_a^2 \sum_{\substack{S\in\mathcal{S}\\ \text{supp}(a)\subseteq S}} \frac{p(S)}{|\mathcal{U}_S|^2}.$$

We set τ_a to the square root of the multiplier of σ_a^2 on the right hand side above. This leads to the optimal values for all σ_a when minimizing $\mathrm{err}_p(q,\tilde{q})^2$. We refer to the works of Yaroslavtsev et al. [37] and of Lebeda and Pagh for more details on calibrating Gaussian noise for ℓ_2^2 error when queries have different scales [20, Chapter 4].

Recall that $\mathcal{S}_\downarrow$ is the collection of all sets $R \subseteq [d]$ such that $R \subseteq S$ for some $S \in \mathcal{S}$. The properties of Algorithm 1 are summarized in the following theorem:

Theorem 8. *Let D be a dataset containing data points $x \in \mathcal{U}$, where $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, $\mathcal{U}_i := \{0,\ldots,m_i-1\}$. Let the workload $Q_\mathcal{S}$ of marginal queries on the universe $\mathcal{U}$ be defined by a collection $\mathcal{S}$ of subsets of $[d]$ such that for each subset $S \in \mathcal{S}$, the workload contains all queries $q_{S,t}$ for all $t \in \mathcal{U}_S := \prod_{i\in S}\mathcal{U}_i$. Given a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, there exists a μ-GDP mechanism that estimates all marginal queries in $Q_\mathcal{S}$ and satisfies*

$$\mathrm{err}_p(q,\tilde{q}) = \frac{1}{\mu}\sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j-1)\right)\sqrt{\sum_{\substack{S\in\mathcal{S}\\S\supseteq R}}\frac{p(S)}{|\mathcal{U}_S|^2}}\,.$$

Additionally, the noise for all marginal estimates can be sampled in time

$$O\left(\sum_{S\in\mathcal{S}}\left(\prod_{i\in S}m_i\right)\log\left(\prod_{i\in S}m_i\right)+\lambda\sum_{R\in\mathcal{S}_\downarrow}\prod_{i\in R}(m_i-1)\right)$$

where $O(\lambda)$ is the time required for sampling a standard Gaussian.

Proof. The mechanism is Algorithm 1. The proofs are in the extended version of the paper. The proofs for privacy and error bounds are standard. The time complexity of sampling noise utilizes FFT. □

Here we give a short intuition behind the proof of each property of the theorem. The privacy proof is standard. We show that we can release all $\tilde{F}_a(D)$ under μ-GDP using the (complex) Gaussian mechanism (Lemma 7). We recover the marginal estimates using post-processing which does not affect the privacy guarantees. The noise for each marginal estimate is the sum of Gaussian random variables. The error is therefore also distributed as a Gaussian. We find the value of $\mathrm{err}_p(q,\tilde{q})$ by simply summing the error terms for all marginal estimates. For the running time, notice that each marginal estimate is a sum of $|\mathcal{U}_S|$ terms, and the post-processing step therefore takes time $O(|\mathcal{U}_S|)$ for a single marginal estimate. However, we can use a multi-dimensional discrete fast Fourier transform to speed up computation of all marginal estimates of S from $O(|\mathcal{U}_S|^2)$ to $O(|\mathcal{U}_S|\log(|\mathcal{U}_S|))$.

Next we turn to the setting in which our measure of error is the maximum standard deviation of $\tilde{q}_{S,t}(D) - q_{S,t}(D)$ over all $S \in \mathcal{S}$ and all $t \in \mathcal{U}_S$. We reuse Algorithm 1, with a "worst case" choice of p, i.e., the p that maximizes the error in Theorem 8. The privacy of this algorithm follows directly from the theorem. We also have the following error bound.

Lemma 9. *Let $\tilde{q}_{S,t}(D)$ be the estimates produced by Algorithm 1 with weights*

$$p^* \in \arg\max\left\{\sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j-1)\right)\sqrt{\sum_{\substack{S\in\mathcal{S}\\S\supseteq R}}\frac{p(S)}{|\mathcal{U}_S|^2}} : \sum_{S\in\mathcal{S}}p(S)=1, p(S)\geq 0\ \forall S\in\mathcal{S}\right\}. \tag{4}$$

Then,

$$\max_{S\in\mathcal{S},t\in\mathcal{U}_S}(\mathbb{E}[(\tilde{q}_{S,t}(D)-q_{S,t}(D))^2])^{1/2}=\frac{1}{\mu}\sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j-1)\right)\sqrt{\sum_{\substack{T\in\mathcal{S}\\T\supseteq R}}\frac{p^*(T)}{|\mathcal{U}_T|^2}}.$$

The lemma follows from proofs in the extended version of the paper. On a high level, to prove the lemma we analyze the first order optimality conditions of (4), and observe that they guarantee that $p^*(S)>0$ only for those $S\in\mathcal{S}$ for which the noise variance $\mathbb{E}[(\tilde{q}_{S,t}(D)-q_{S,t}(D))^2]$ achieves its maximum value over all $S\in\mathcal{S}$. Because of this, the average squared error with weights $p^*(S)$ equals the maximum noise variance. In Section 3.3 we show that the error bound of Lemma 9 is also optimal among all factorization mechanisms.

It is worth noting also that, since the noise added to each query is normally distributed, standard concentration bounds show that

$$\mathbb{E}\left[\max_{S\in\mathcal{S},t\in\mathcal{U}_S}|\tilde{q}_{S,t}(D)-q_{S,t}(D)|\right]\le\sqrt{2\ln|Q_{\mathcal{S}}|}\max_{S\in\mathcal{S},t\in\mathcal{U}_S}(\mathbb{E}[(\tilde{q}_{S,t}(D)-q_{S,t}(D))^2])^{1/2}.$$

Remark 2. It is likely that our approach extends to other measures considered in prior work [24, 30, 36], but in this paper we focus only on the most commonly used error measures of average and maximum squared error.

3.3 Lower Bounds

One of our technical contributions is a simple proof that our mechanism is optimal among all factorization mechanisms. The technical details and our proofs are in the extended version of the paper. Here we discuss factorization mechanisms and present our main result.

We start with some necessary notation. For a matrix M, let $\|M\|_{1\to2}$ be the maximum ℓ_2 norm of a column of M, and $\|M\|_{2\to\infty}$ the maximum ℓ_2 norm of a row of M. Let $\|M\|_F$ be the Frobenius norm of M, i.e., $\|M\|_F=\sqrt{\mathrm{tr}(M^TM)}$ for real matrices, and $\sqrt{\mathrm{tr}(M^*M)}$ for complex matrices.

Let us recall the factorization mechanisms framework. Suppose we are given a query workload Q over a universe $\mathcal{U}$: Q is a set of queries, where each q is a function from $\mathcal{U}$ to $\mathbb{R}$, and induces a query on datasets $D=(x^{(1)},\dots,x^{(n)})$ given by $q(D):=\sum_{i=1}^n q(x^{(i)})$. We also use $Q(D)$ to denote the vector of query answers $(q(D))_{q\in Q}$. We represent Q by a workload matrix $W\in\mathbb{R}^{Q\times\mathcal{U}}$ with entries $W_{q,x}:=q(x)$. We also represent the dataset D by a histogram vector $h\in\mathbb{Z}^{\mathcal{U}}$, where $h_x:=|\{i:x^{(i)}=x\}|$ is the number of occurrences of x in D. These definitions turn the workload into a linear function of the histogram: $Q(D)=Wh$.

For a factorization $W=LR$ of the workload matrix, we define a corresponding private factorization mechanism for answering queries in Q as follows. We draw a normally distributed vector $Z\sim\mathcal{N}\left(0,\frac{\|R\|_{1\to2}^2}{\mu^2}I_r\right)$, where r is the number of rows of R, and output the vector of query answers $L(Rh+Z)=Wh+LZ=Q(D)+LZ$. Note that, using r_x to denote the column of R indexed by $x\in\mathcal{U}$, $Rh=\sum_{i=1}^n r_{x^{(i)}}$ is simply another workload of queries with sensitivity $\max_{x\in\mathcal{U}}\|r_x\|_2=\|R\|_{1\to2}$. These are usually called the strategy queries, and we can view the factorization mechanism as answering a well chosen set of strategy queries and using them to reconstruct answers to the queries in Q. In our algorithm the strategy queries compute weighted Fourier coefficient of the histogram vector. We show how our mechanisms are expressed in the factorization framework in the extended version of the paper.

It is easy to verify that a factorization mechanism satisfies μ-GDP: outputting $Rh+Z$ satisfies μ-GDP by Lemma 3, and then multiplying by L is just post-processing. Moreover, an easy calculation

shows that the factorization mechanism achieves error bounds

$$\mathbb{E}[\|Wh - (L(Rh + Z))\|_2^2]^{1/2} = \mathbb{E}[\|LZ\|_2^2]^{1/2} = \frac{1}{\mu}\|L\|_F\|R\|_{1\to 2};$$

$$\max_{q\in Q}\mathbb{E}[|(Wh)_q - (L(Rh + Z))_q|^2]^{1/2} = \max_{q\in Q}\mathbb{E}[|(LZ)_q|^2]^{1/2} = \frac{1}{\mu}\|L\|_{2\to\infty}\|R\|_{1\to 2}.$$

This motivates the definitions of the $\gamma_F(W)$ and $\gamma_2(W)$ norms: they correspond to the minimal error bounds achievable by a factorization mechanism. For a $K \times N$ real matrix W, we have the factorization norms

$$\gamma_F(W) := \inf\{\|L\|_F\|R\|_{1\to 2} : LR = W\}; \quad \gamma_2(W) := \inf\{\|L\|_{2\to\infty}\|R\|_{1\to 2} : LR = W\}.$$

Note also that if we are interested in weighted root mean squared error, i.e., if each query $q \in Q$ is assigned non-negative weight $p(q)$, then we get the error bound

$$\mathbb{E}\left[\sum_{q\in Q} p(q)((Wh)_q - (L(Rh + Z))_q)^2\right]^{1/2} = \mathbb{E}[\|P^{1/2}LZ\|_2^2] = \frac{1}{\mu}\|P^{1/2}L\|_F\|R\|_{1\to 2},$$

where P is the diagonal matrix indexed by queries and with diagonal entries $P_{q,q} := p(q)$. This error bound is optimized by the factorization that achieves $\gamma_F(P^{1/2}W)$.

Our main result is that the factorizations underlying our mechanisms are optimal for weighted root mean squared error and maximum error, respectively.

Theorem 10. *For a workload of marginal queries $Q_{\mathcal{S}}$ over a universe $\mathcal{U}$ with workload matrix W, a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, and a corresponding diagonal matrix P indexed by queries (S, t), $S \in \mathcal{S}$, $t \in \mathcal{U}_S$, with entries $P_{(S,t),(S,t)} = \frac{p(S)}{|\mathcal{U}_S|}$, the factorization $LR = W$ which corresponds to Algorithm 1 (given in the extended version of the paper) satisfies*

$$\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F\|R\|_{1\to 2} = \sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j - 1)\right)\sqrt{\sum_{\substack{T\in\mathcal{S}\\ T\supseteq R}}\frac{p(T)}{|\mathcal{U}_T|^2}}. \tag{5}$$

Moreover, for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (4), the factorization satisfies

$$\gamma_2(W) = \|L\|_{2\to\infty}\|R\|_{1\to 2} = \sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j - 1)\right)\sqrt{\sum_{\substack{T\in\mathcal{S}\\ T\supseteq R}}\frac{p^*(T)}{|\mathcal{U}_T|^2}},$$

where we recall that $\mathcal{S}_\downarrow$ is the collection of sets R which are subsets of S for some $S \in \mathcal{S}$.

4 Extensions

In Appendix A we extend our technique to a generalization of marginal queries that we call product queries. Recall that product queries are defined by functions $\phi = (\phi_1, \ldots, \phi_d)$, $\phi_j : \mathcal{U}_j \to \mathbb{R}$, and a query $q_{S,t}$ on a single data point is defined by

$$q^{\phi}_{S,t}(x) := \prod_{j\in S}\phi_j((t_j - x_j) \bmod m_j).$$

We recover standard marginals $q_{S,t}$ by setting $\phi_j(z) := \mathbf{1}\{z = 0\}$. Making other choices of ϕ can allow for more expressive queries, and we explore this in Section A.2.

We omit most technical details from the main body due to space limitations. We note that our mechanism generalizes naturally to these more expressive queries with two simple changes. We still use the Fourier queries as the strategy queries, and reconstruct answers to the product queries

using the inverse Fourier transform. During reconstruction, we need to scale the contribution of each Fourier query using the Fourier coefficients of the ϕ_j functions, and, to achieve optimal error, we also correspondingly modify the importance weight τ_a of each Fourier query.

Algorithm 2 Algorithm for estimating a workload $Q^{\phi}_{\mathcal{S}}$ of weighted product queries.

For any $a \in \mathcal{U}$, let

$$\widehat{\phi_j}(a) := \sum_{z=0}^{m_j-1} \phi_j(z)\omega_{m_j}^{-a\cdot z},\ \tau_a := \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}},\ \text{and}\ \tau := \frac{1}{\mu^2}\sum_{a\in\mathcal{U}}\tau_a.$$

For any a such that $\tau_a > 0$, privately release an estimate of $F_a(D)$ by adding independent noise such that

$$\tilde{F}_a(D) = F_a(D) + Z_a,\ \text{where}\ Z_a \sim \mathcal{CN}\left(0, \frac{2\tau}{\tau_a}\right).$$

Estimates for product queries for $S \in \mathcal{S}$ can be recovered using post-processing by computing

$$\tilde{q}^{\phi}_{S,t}(D) = \mathrm{Re}\left(\frac{1}{|\mathcal{U}_S|}\sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}}\left(\prod_{j\in S}\widehat{\phi_j}(a_j)\right)\chi_a(t)\tilde{F}_a(D)\right),\ \text{where}\ \chi_a(t) = \prod_{i\in S}\omega_{m_i}^{a_i t_i}.$$

Theorem 11. *(simplified) Let D be a dataset containing data points $x \in \mathcal{U}$, where $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$. For a choice of functions $\phi := (\phi_1, \ldots, \phi_d)$, $\phi_i : \mathcal{U}_i \to \mathbb{R}$, let the workload $Q^{\phi}_{\mathcal{S}}$ of product queries $q^{\phi}_{S,t}$ defined by a collection $\mathcal{S}$ of subsets of $[d]$ be such that, for each subset $S \in \mathcal{S}$, the workload contains all queries $q^{\phi}_{S,t}$ for all $t \in \mathcal{U}_S := \prod_{i\in S}\mathcal{U}_i$. Given a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, Algorithm 2 is an optimal factorization mechanism for the weighted error defined as*

$$\mathrm{err}_p(q, \tilde{q}) := \left(\sum_{S\in\mathcal{S}}\frac{p(S)}{|\mathcal{U}_S|}\sum_{t\in\mathcal{U}_S}\mathbb{E}[(\tilde{q}^{\phi}_{S,t}(D) - q^{\phi}_{S,t}(D))^2]\right)^{1/2}.$$

Additionally, we can optimize over weight functions similarly to Lemma 9 to recover an optimal factorization for maximum variance.

Proof. We present our upper bound for weighted error in Theorem 12, and our result for maximum error in Theorem 14. We provide matching lower bounds in Theorem 17. □

Finally, we show in Appendix A.2 how to embed extended marginal as product queries. Extended marginals combine prefix (and suffix) queries over numerical attributes with standard marginals over categorical attributes. Here we show that our mechanism is *almost* optimal. The small gap between our upper and lower bounds arises because we embed prefix and suffix queries into a slightly more expressive product query. However, note that no optimal explicit factorization is known even for a single prefix query despite extensive study of this problem due to its relevance for private machine learning (e.g. [15–17, 31]). Our technique matches the error of the factorization mechanism of [16] for a single prefix query, and is only a small additive constant worse than the best known explicit factorization [15]. We note that the only known explicit factorization that outperforms ours on prefix queries is concurrent work [15], and our mechanism answers a much more general class of queries.

We also present matching lower bounds for factorization mechanisms over product queries in Appendix B.1, and nearly matching lower bounds for extended marginals in Appendix B.2.

Acknowledgements

We thank Ryan McKenna for pointing out relevant related work. The work of Christian Janos Lebeda is supported by grant ANR-20-CE23-0015 (Project PRIDE) and the ANR-22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR. Aleksandar Nikolov and Haohua Tang were supported by an NSERC Discovery Grant (RGPIN-2021-03206), and the Canada Research Chairs program (CRC-2020-00004).

A Additional Upper Bounds for Product Queries and Extended Marginals

Here we present results for product queries and extended marginals that we left out of the main body. We omit the proofs due to space limitations and refer to the extended version of our paper for more details.

A.1 Estimating Product Queries

We have the following theorem for Algorithm 2, generalizing Theorem 8.

Theorem 12. *Let D be a dataset containing data points $x \in \mathcal{U}$, where $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$. For a choice of functions $\phi := (\phi_1, \ldots, \phi_d)$, $\phi_i : \mathcal{U}_i \to \mathbb{R}$, let the workload $Q^{\phi}_{\mathcal{S}}$ of product queries $q^{\phi}_{S,t}$ defined by a collection $\mathcal{S}$ of subsets of $[d]$ be such that, for each subset $S \in \mathcal{S}$, the workload contains all queries $q^{\phi}_{S,t}$ for all $t \in \mathcal{U}_S := \prod_{i \in S} \mathcal{U}_i$. Given a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, there exists a μ-GDP mechanism that estimates all queries in $Q^{\phi}_{\mathcal{S}}$, and satisfies*

$$\begin{aligned}\mathrm{err}_p(q, \tilde{q}) &= \left(\sum_{S \in \mathcal{S}} \frac{p(S)}{|\mathcal{U}_S|} \sum_{t \in \mathcal{U}_S} \mathbb{E}[(\tilde{q}^{\phi}_{S,t}(D) - q^{\phi}_{S,t}(D))^2] \right)^{1/2} \\ &= \frac{1}{\mu} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_{\downarrow}}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}},\end{aligned}$$

where $\widehat{\phi_j}(a_j) := \sum_{z=0}^{m_j - 1} \phi_j(z) \omega_{m_j}^{-a_j z}$. Additionally, the noise for all product query estimates can be sampled in time

$$O\left(\sum_{S \in \mathcal{S}} \left(\prod_{i \in S} m_i \right) \log \left(\prod_{i \in S} m_i \right) + \lambda \sum_{R \in \mathcal{S}_{\downarrow}} \prod_{i \in R} (m_i - 1) \right)$$

where $O(\lambda)$ is the time required for sampling a standard Gaussian.

We can extend this result to the maximum variance error measure, as we did for marginals.

Lemma 13. *Let $\tilde{q}^{\phi}_{S,t}(D)$ be the estimates produced by Algorithm 2 with weights*

$$p^* \in \arg\max \left\{ \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_{\downarrow}}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}} : \sum_{S \in \mathcal{S}} p(S) = 1, p(S) \geq 0 \ \forall S \in \mathcal{S} \right\}. \quad (6)$$

Then,

$$\max_{S\in\mathcal{S},t\in\mathcal{U}_S}(\mathbb{E}[(\tilde{q}^{\phi}_{S,t}(D)-q^{\phi}_{S,t}(D))^2])^{1/2}=\frac{1}{\mu}\sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\in\mathcal{S}_\downarrow}}\sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}}\frac{p^*(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}}.$$

The next theorem follows from Lemma 13.

Theorem 14. *Let D be a dataset containing data points $x\in\mathcal{U}$, where $\mathcal{U}:=\mathcal{U}_1\times\ldots\times\mathcal{U}_d$, $\mathcal{U}_i:=\{0,\ldots,m_i-1\}$. For a choice of functions $\phi:=(\phi_1,\ldots,\phi_d)$, $\phi_i:\mathcal{U}_i\to\mathbb{R}$, let the workload $Q^{\phi}_{\mathcal{S}}$ of product queries $q^{\phi}_{S,t}$ defined by a collection $\mathcal{S}$ of subsets of $[d]$ be such that, for each subset $S\in\mathcal{S}$, the workload contains all queries $q^{\phi}_{S,t}$ for all $t\in\mathcal{U}_S:=\prod_{i\in S}\mathcal{U}_i$. Let $p^*:\mathcal{S}\to\mathbb{R}_{\geq 0}$ be the function defined by* (6)*. There exists a μ-GDP mechanism that estimates all product queries in $Q^{\phi}_{\mathcal{S}}$ and satisfies*

$$\max_{S\in\mathcal{S},t\in\mathcal{U}_S}(\mathbb{E}[(\tilde{q}^{\phi}_{S,t}(D)-q^{\phi}_{S,t}(D))^2])^{1/2}=\frac{1}{\mu}\sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\in\mathcal{S}_\downarrow}}\sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}}\frac{p^*(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}}.$$

Additionally, given the optimal p^*, which can be computed in time polynomial in $|\mathcal{S}_\downarrow|+\max_{S\in\mathcal{S}}|S|$, the noise for all product query estimates can be sampled in time

$$O\left(\sum_{S\in\mathcal{S}}\left(\prod_{i\in S}m_i\right)\log\left(\prod_{i\in S}m_i\right)+\lambda\sum_{R\in\mathcal{S}_\downarrow}\prod_{i\in R}(m_i-1)\right)$$

where $O(\lambda)$ is the time required for sampling a standard Gaussian.

Remark 3. The model can be generalized further by allowing the functions $(\phi_i)_{i\in S}$ to depend on the set of attributes S. We do not pursue this here, mostly to avoid introducing more complex notation.

A.2 Estimating Workloads of Extended Marginals

In this subsection we consider extended marginal queries, which we defined in Section 2.1, and we recall the definition here. The set of attributes $[d]$ is partitioned into subsets C (the categorical attributes) and N (the numerical ones). As with standard marginal queries, the domain of attribute i is $\mathcal{U}_i$, and the universe is $\mathcal{U}:=\mathcal{U}_1\times\ldots\times\mathcal{U}_d$. We allow prefix and suffix queries on the numerical attributes. To encode them, we allow, for any numerical attribute $i\in N$, t_i to be positive or negative, with positive values denoting prefix predicates, and negative values denoting suffix predicates. We thus define $T_i:=\mathcal{U}_i$ for $i\in C$, and $T_i:=\{-m_i,\ldots,0,\ldots,m_i-1\}$ for $i\in N$, and $T_S:=\prod_{i\in S}T_i$. We now redefine the query $q_{S,t}$ for $S\subseteq[d]$ and $t\in T_S$ as

$$q_{S,t}(x):=\left(\prod_{j\in S\cap C}\mathbf{1}\{x_j=t_j\}\right)\left(\prod_{\substack{j\in S\cap N\\ t_j\geq 0}}\mathbf{1}\{x_j\leq t_j\}\right)\left(\prod_{\substack{j\in S\cap N\\ t_j<0}}\mathbf{1}\{x_j\geq|t_j|\}\right). \tag{7}$$

In the case when $C=[d]$, these are just marginal queries. On the other hand, if $N=[d]$, these are multi-dimensional (prefix and suffix) range queries. Note that we do not explicitly support the $\mathbf{1}\{x_j\geq 0\}$ suffix query because it is equivalent to a $\mathbf{1}\{x_j\leq m_j-1\}$ prefix query. Given a collection of subsets $\mathcal{S}$ of $[d]$, let us define $Q^{\text{pr-suf}}_{\mathcal{S}}$ to be the workload of all queries $q_{S,t}$ for $S\in\mathcal{S}$ and $t\in T_S$. We also define a variant which contains only prefix predicates for numerical attributes: $Q^{\text{pref}}_{\mathcal{S}}$ contains all queries $q_{S,t}$ for $S\in\mathcal{S}$ and $t\in\mathcal{U}_S$.

Remark 4. When $t_i = m_i - 1$, $\mathbf{1}\{x_i \le t_i\}$ always evaluates to 1, and when $t_i = -m_i$, $\mathbf{1}\{x_i \ge |t_i|\}$ always evaluates to 0. It is possible to achieve slightly stronger results by removing these values. The query with $t_i = m_i - 1$ can then be recovered from other extended marginal queries that omit the attribute i.

Next, we show how to "embed" extended marginal queries into the product queries studied in the previous subsection. This idea was already used for prefix queries by Choquette-Choo et al. on factorization mechanisms for optimization [4], and is implicit in the group algebra based factorization mechanism of Henzinger and Upadhyay [16] (see also Section 5.1 of [15]).

Lemma 15. *Given a partition of $[d]$ into numerical attributes N and categorical attributes C, and a corresponding universe $\mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, define $\mathcal{U}' = \mathcal{U}'_1 \times \ldots \times \mathcal{U}'_d$, where $\mathcal{U}'_i = \mathcal{U}_i$ for $i \in C$, and $\mathcal{U}'_i = \{0, \ldots, 2m_i - 1\}$ for $i \in N$. There exist functions $\phi_1, \ldots, \phi_d$, $\phi_i : \mathcal{U}'_i \to \{0, 1\}$, such that for any $S \subseteq [d]$ and any $t \in T_S$, there exists a $t' \in \mathcal{U}'_S$ for which $q_{S,t}(x) = q^{\phi}_{S,t'}(x)$ for all $x \in \mathcal{U}$. Moreover, the mapping from t to t' is a bijection.*

Proof. We define $\phi_i(z) := \mathbf{1}\{z = 0\}$ for $i \in C$, and $\phi_i(z) := \mathbf{1}\{z \le m_i - 1\}$ for $i \in N$. Notice that, for $t, z \in \{0, \ldots, 2m_i - 1\}$,

$$(t - z) \bmod 2m_i = \begin{cases} t - z & z \le t \\ 2m_i + (t - z) & z > t \end{cases}. \tag{8}$$

Suppose that $z, t \in \{0, \ldots, m_i - 1\}$. When $z \le t$, clearly $t - z \le m_i - 1$. When $z > t$, since $z - t \le m_i - 1$, we get $2m_i + (t - z) > m_i - 1$. Therefore, for $z, t \in \{0, \ldots, m_i - 1\}$, $\phi_i(t - z \bmod 2m_i) = \mathbf{1}\{z \le t\}$.

Now consider $z \in \{0, \ldots, m_i - 1\}$ but $t \in \{m_i, \ldots, 2m_i - 1\}$. In this case, $z \le t$ always holds, so $(t - z) \bmod 2m_i = t - z$. Then $t - z \le m_i - 1$ if and only if $z \ge t - m_i + 1$. Therefore, for $z \in \{0, \ldots, m_i - 1\}$, and $t \in \{-m_i, \ldots, -1\}$, $\phi_i(t' - z \bmod 2m_i) = \mathbf{1}\{z \ge |t|\}$ where $t' := |t| + m_i - 1 = m_i - 1 - t$.

This means that, when $x \in \mathcal{U} \subseteq \mathcal{U}'$, $S \subseteq [d]$, and $t \in T_S$, $q^{\phi}_{S,t'}(x) = q_{S,t}(x)$, where

$$t'_i := \begin{cases} t_i & i \in C, \text{ or } i \in N, t_i \ge 0 \\ m_i - 1 - t_i & i \in N \text{ and } t_i < 0 \end{cases}.$$

It is easy to check that this mapping between t and t' is a bijection. □

We can then use Algorithm 2 to answer the extended marginal queries $Q_{\mathcal{S}}$. Doing so gives us the guarantees in the following theorem. In Appendix B.2 we give a lower bound for $Q^{\text{pref}}_{\mathcal{S}}$. Note that unlike our other results our bounds for extended marginals are "only" nearly optimal. The gap occurs because our product query embedding is slightly more expressive than $Q^{\text{pref}}_{\mathcal{S}}$. Nevertheless, no optimal construction is known even for a single prefix query despite significant attention from the research community (see [15] for the current best known bounds).

Theorem 16. *Let D be a dataset containing data points $x \in \mathcal{U}$, where $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$, and the attributes $[d]$ are partitioned into categorical attributes C, and numerical attributes N. For any collection $\mathcal{S}$ of subsets of $[d]$, the workload $Q^{\text{pr-suf}}_{\mathcal{S}}$ defined above, and a weight function $p : \mathcal{S} \to \mathbb{R}_{\ge 0}$, there exists a μ-GDP mechanism that estimates all queries in $Q^{\text{pr-suf}}_{\mathcal{S}}$ and*

satisfies

$$\mathrm{err}_p(q,\tilde{q}) := \left(\sum_{S\in\mathcal{S}} \frac{p(S)}{|T_S|} \sum_{t\in T_S} \mathbb{E}[(\tilde{q}_{S,t}(D) - q_{S,t}(D))^2]\right)^{1/2}$$
$$= \frac{1}{\mu} \sum_{\substack{R\subseteq C\\ O\subseteq N}} \left(\prod_{j\in R}(m_j-1)\right)\cdot\left(\prod_{j\in O}\eta(m_j)\right)\sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq R\cup O}} \frac{p(S)}{|\mathcal{U}_{S\cap C}|^2\cdot 4^{|S\cap N|}}},$$

where $\eta(m) := \frac{1}{m}\sum_{\ell=1}^{m} \frac{1}{\sin\left(\frac{\pi(2\ell-1)}{2m}\right)}$. *Furthermore, there is a weight function* $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ *computable in time polynomial in* $|\mathcal{S}_\downarrow| + \max_{S\in\mathcal{S}} |S|$, *such that*

$$\max_{S\in\mathcal{S}, t\in T_S} \left(\mathbb{E}[(\tilde{q}_{S,t}(D) - q_{S,t}(D))^2]\right)^{1/2}$$
$$= \frac{1}{\mu} \sum_{\substack{R\subseteq C\\ O\subseteq N}} \left(\prod_{j\in R}(m_j-1)\right)\cdot\left(\prod_{j\in O}\eta(m_j)\right)\sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq R\cup O}} \frac{p^*(S)}{|\mathcal{U}_{S\cap C}|^2\cdot 4^{|S\cap N|}}}.$$

Additionally, the noise for all extended marginal estimates can be sampled in time

$$O\left(\sum_{S\in\mathcal{S}}\left(\prod_{i\in S} m_i'\right)\log\left(\prod_{i\in S} m_i'\right) + \lambda \sum_{R\in\mathcal{S}_\downarrow}\prod_{i\in R}(m_i-1)\right)$$

where $O(\lambda)$ *is the time required for sampling a standard Gaussian, and* $m_i' = m_i$ *if* $i \in C$ *and* $m_i' = 2m_i$ *if* $i \in N$

The same guarantees hold when $Q_{\mathcal{S}}^{pr\text{-}suf}$ *is replaced by* $Q_{\mathcal{S}}^{pref}$.

In the case when $d := 1$, $m_1 := m$, $N := \{1\}$, and the only set in $\mathcal{S}$ is $S := \{1\}$, we simply have the workload of prefix and suffix queries on $\mathcal{U} = \mathcal{U}_1 = \{0,\dots,m-1\}$. Then Theorem 16 gives us (for $p(S) = 1$)

$$\mathrm{err}_p(q,\tilde{q}) = \frac{1}{2\mu}(1+\eta(m)) = \frac{1}{2\mu} + \frac{1}{2\mu m}\sum_{\ell=1}^{m}\frac{1}{\sin\left(\frac{\pi(2\ell-1)}{2m}\right)},$$

which recovers the explicit upper bound on the error of a factorization algorithm for this workload due to Henzinger and Upadhyay [16]. We remark that the function $\eta(m)$ is asymptotically $\eta(m) = \frac{1}{\pi}\ln(m) + O(1)$. Our bound is a small additive constant worse than the best known explicit factorization [15] for a single prefix query. We note that the only known explicit factorization that outperforms ours on prefix queries is concurrent work [15], and our mechanism handles extended marginals which is a much more general class of queries.

B Additional Lower Bounds

In this section we present lower bounds for factorization mechanisms. We show that our mechanisms are exactly optimal for marginals and product queries both under weighted and maximum error. We give a lower bound for extended marginals that nearly matches the factorization implied by the algorithms in Section 3 and Appendix A.2. Our lower bounds are key technical contributions of our work. However, we leave out all proofs due to space limitation and refer to the extended version of the paper for much more detail.

B.1 Lower Bounds for Product Queries

In the extended version of the paper we show how Algorithm 2 describes a factorization mechanism for workloads of product queries. The following theorem shows optimality of Algorithm 2 among all factorization mechanisms.

Theorem 17. *Let $Q_{\mathcal{S}}^{\phi}$ be a workload of product queries over a universe $\mathcal{U}$, with functions $\phi := (\phi_1, \ldots, \phi_d)$ associated with the attributes, and with workload matrix W. Let $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$ be a weight function, and define the corresponding diagonal matrix P indexed by queries (S, t), $S \in \mathcal{S}$, $t \in \mathcal{U}_S$, with entries $P_{(S,t),(S,t)} = \frac{p(S)}{|\mathcal{U}_S|}$. Then the factorization $LR = W$ that corresponds to Algorithm 2 satisfies*

$$\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F \|R\|_{1\to 2} = \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi}_j(a_j)|^2}{|\mathcal{U}_S|^2}}, \tag{9}$$

where we recall that $\mathcal{S}_\downarrow$ is the collection of sets R which are subsets of S for some $S \in \mathcal{S}$.

Moreover, for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (6)*, the factorization satisfies*

$$\gamma_2(W) = \|L\|_{2\to\infty} \|R\|_{1\to 2} = \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p^*(S) \prod_{j \in S} |\widehat{\phi}_j(a_j)|^2}{|\mathcal{U}_S|^2}}.$$

B.2 Lower Bounds for Extended Marginals

The following lower bound implies our mechanism is *nearly* optimal for extended marginals.

Theorem 18. *Let $Q_{\mathcal{S}}^{pref}$ be a workload of extended marginal queries over a universe $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, where $[d]$ is partitioned in categorical attributes C, and numerical attributes N. Let $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$ be a weight function, and define the corresponding diagonal matrix P indexed by queries (S, t), $S \in \mathcal{S}$, $t \in \mathcal{U}_S$, with entries $P_{(S,t),(S,t)} = \frac{p(S)}{|\mathcal{U}_S|}$. Then*

$$\gamma_F(P^{1/2}W) \geq \sum_{\substack{R \subseteq C \\ O \subseteq N}} \left(\prod_{j \in R} (m_j - 1) \right) \cdot \left(\prod_{j \in O} \zeta(m_j) \right) \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq R \cup O}} \frac{p(S) \prod_{j \in (S \cap N) \setminus O} \left(1 + \frac{1}{m_j}\right)^2}{|\mathcal{U}_{S \cap C}|^2 \cdot 4^{|S \cap N|}}},$$

where $\zeta(m) := \frac{1}{m} \sum_{\ell=1}^{m-1} \frac{1}{\sin\left(\frac{\pi \ell}{m}\right)}$.

References

[1] John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson L. Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. 2022. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review* 2 (2022).

[2] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. 2007. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, Leonid Libkin (Ed.). ACM, 273–282. https://doi.org/10.1145/1265530.1265569

[3] Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. 2014. Fingerprinting codes and the price of approximate differential privacy. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, David B. Shmoys (Ed.). ACM, 1–10. https://doi.org/10.1145/2591796.2591877

[4] Christopher A. Choquette-Choo, Hugh Brendan McMahan, J. Keith Rush, and Abhradeep Guha Thakurta. 2023. Multi-Epoch Matrix Factorization Mechanisms for Private Machine Learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*,

Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 5924–5963. https://proceedings.mlr.press/v202/choquette-choo23a.html

[5] Matthew Dawson, Badih Ghazi, Pritish Kamath, Kapil Kumar, Ravi Kumar, Bo Luan, Pasin Manurangsi, Nishanth Mundru, Harikesh Nair, Adam Sealfon, and Shengyu Zhu. 2023. Optimizing Hierarchical Queries for the Attribution Reporting API. In *AdKDD@KDD (CEUR Workshop Proceedings, Vol. 3556)*. CEUR-WS.org.

[6] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *PODS*. ACM, 202–210.

[7] Jinshuo Dong, Aaron Roth, and Weijie J Su. 2022. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 84, 1 (2022), 3–37.

[8] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4004)*. Springer, 486–503. https://doi.org/10.1007/11761679_29

[9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 265–284.

[10] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. 2015. Efficient Algorithms for Privately Releasing Marginals via Convex Relaxations. *Discret. Comput. Geom.* 53, 3 (2015), 650–673.

[11] Cynthia Dwork and Kobbi Nissim. 2004. Privacy-Preserving Datamining on Vertically Partitioned Databases. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings (Lecture Notes in Computer Science, Vol. 3152)*, Matthew K. Franklin (Ed.). Springer, 528–544.

[12] Alexander Edmonds, Aleksandar Nikolov, and Jonathan R. Ullman. 2020. The power of factorization mechanisms in local and central differential privacy. In *STOC*. ACM, 425–438.

[13] Simson L. Garfinkel, John M. Abowd, and Christian Martindale. 2019. Understanding database reconstruction attacks on public data. *Commun. ACM* 62, 3 (2019), 46–53.

[14] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proc. VLDB Endow.* 3, 1 (2010), 1021–1032.

[15] Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. 2025. Normalized Square Root: Sharper Matrix Factorization Bounds for Differentially Private Continual Counting. *CoRR* abs/2509.14334 (2025).

[16] Monika Henzinger and Jalaj Upadhyay. 2025. Improved Differentially Private Continual Observation Using Group Algebra. In *SODA*. SIAM, 2951–2970.

[17] Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. 2023. Almost Tight Error Bounds on Differentially Private Continual Counting. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*. SIAM, 5003–5039. https://doi.org/10.1137/1.9781611977554.CH183

[18] JASON. 2020. *Formal Privacy Methods for the 2020 Census*. Technical Report JSR-19-2F. U.S. Census Bureau. https://www.census.gov/programs-surveys/decennial-census/decade/2020/planning-management/plan/planning-docs/privacy-methods-2020-census.html [Accessed 14-July-2025].

[19] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam D. Smith, and Jonathan R. Ullman. 2010. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, Leonard J. Schulman (Ed.). ACM, 775–784.

[20] Christian Janos Lebeda. 2023. *Differentially Private Release of Sparse and Skewed Data*. Ph. D. Dissertation. IT University of Copenhagen. https://en.itu.dk/-/media/EN/Research/PhD-Programme/PhD-defences/2023/PhD-thesis-Final-Version--Christian-Janos-Lebeda-pdf.pdf

[21] Christian Janos Lebeda. 2025. Better Gaussian Mechanism using Correlated Noise. In *2025 Symposium on Simplicity in Algorithms (SOSA)*. 119–133. https://doi.org/10.1137/1.9781611978315.9 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611978315.9

[22] Chao Li and Gerome Miklau. 2013. Optimal error of query sets under the differentially-private matrix mechanism. In *ICDT*. ACM, 272–283.

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

[24] Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. 2024. Optimality of Matrix Mechanism on ℓ_p^p-metric. *CoRR* abs/2406.02140 (2024). https://doi.org/10.48550/ARXIV.2406.02140 arXiv:2406.02140

[25] Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. 2020. Factorization norms and hereditary discrepancy. *International Mathematics Research Notices* 2020, 3 (2020), 751–780.

[26] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2023. Optimizing Error of High-Dimensional Statistical Queries Under Differential Privacy. *Journal of Privacy and Confidentiality* 13, 1 (Aug. 2023).

https://doi.org/10.29012/jpc.791

[27] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality* 11, 3 (Dec. 2021). https://doi.org/10.29012/jpc.778

[28] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *ICML (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 4435–4444.

[29] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. 2016. The Geometry of Differential Privacy: The Small Database and Approximate Cases. *SIAM J. Comput.* 45, 2 (2016), 575–616.

[30] Aleksandar Nikolov and Haohua Tang. 2024. General Gaussian Noise Mechanisms and Their Optimality for Unbiased Mean Estimation. In *ITCS (LIPIcs, Vol. 287)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 85:1–85:23.

[31] Krishna Pillutla, Jalaj Upadhyay, Christopher A. Choquette-Choo, Krishnamurthy Dvijotham, Arun Ganesh, Monika Henzinger, Jonathan Katz, Ryan McKenna, H. Brendan McMahan, Keith Rush, Thomas Steinke, and Abhradeep Thakurta. 2025. Correlated Noise Mechanisms for Differentially Private Learning. *CoRR* abs/2506.08201 (2025). https://doi.org/10.48550/ARXIV.2506.08201 arXiv:2506.08201

[32] Justin Thaler, Jonathan R. Ullman, and Salil P. Vadhan. 2012. Faster Algorithms for Privately Releasing Marginals. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 7391)*, Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer (Eds.). Springer, 810–821. https://doi.org/10.1007/978-3-642-31594-7_68

[33] Jonathan R. Ullman and Salil P. Vadhan. 2010. PCPs and the Hardness of Generating Synthetic Data. *Electron. Colloquium Comput. Complex.* TR10-017 (2010). ECCC:TR10-017 https://eccc.weizmann.ac.il/report/2010/017

[34] United States Census Bureau. 2021. The Census Bureau's Simulated Reconstruction-Abetted Re-identification Attack on the 2010 Census. https://www.census.gov/data/academy/webinars/2021/disclosure-avoidance-series/simulated-reconstruction-abetted-re-identification-attack-on-the-2010-census.html. [Accessed 14-July-2025].

[35] Yingtai Xiao, Guanlin He, Levent Toksoz, Zeyu Ding, Danfeng Zhang, and Daniel Kifer. 2025. ResidualPlanner+: a scalable matrix mechanism for marginals and beyond. *arXiv preprint arXiv:2305.08175* (2025).

[36] Yingtai Xiao, Guanlin He, Danfeng Zhang, and Daniel Kifer. 2023. An optimal and scalable matrix mechanism for noisy marginals under convex loss functions. *Advances in Neural Information Processing Systems* 36 (2023), 20495–20539.

[37] Grigory Yaroslavtsev, Graham Cormode, Cecilia M. Procopiuc, and Divesh Srivastava. 2013. Accurate and efficient private release of datacubes and contingency tables. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, Christian S. Jensen, Christopher M. Jermaine, and Xiaofang Zhou (Eds.). IEEE Computer Society, 745–756. https://doi.org/10.1109/ICDE.2013.6544871

[38] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. PrivSyn: Differentially Private Data Synthesis. In *USENIX Security Symposium*. USENIX Association, 929–946.

Received December 2025; accepted February 2026