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

Christian Janos Lebeda, Aleksandar Nikolov, Haohua Tang

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

CHRISTIAN JANOS LEBEDA, Inria, Université de Montpellier, INSERM, 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.

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 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 [KRSU10], or to infer the membership of a given data point in

Authors' Contact Information: Christian Janos Lebeda, Inria, Université de Montpellier, INSERM, 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.

the dataset [BUV14]. For this reason, we adopt the differential privacy framework [DMNS06], 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 [GAM19, JAS20, Uni21], the US Census Bureau implemented a new differentially private algorithm, the TopDown algorithm, for the 2020 Census [AAC+22]. 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 [MMS21, MSM19, ZWL+21]. 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 [MMHM23]), 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. [MMHM23].

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 [BCD+07], 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. Later work showed lower bounds on the minimum error necessary to answer marginal queries under differential privacy [KRSU10, BUV14], and investigated the trade-offs between privacy, accuracy, and computational complexity in answering marginal queries [UV10, TUV12, DNT15]. McKenna, Miklau, Hay, and Machanavajjhala proposed an efficient method to approximately optimize over a class of private algorithms for answering marginal queries [MMHM23]. 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 [XHZK23] gave an explicit algorithm for answering marginal queries which is efficient and optimal over the same class of private algorithms considered by McKenna et al. 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 [XHT+25]. 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., $q_{S,t}(D) := |\{i : x^{(i)}_j = t_j \ \forall j \in S\}|$. 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

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

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.

Let us take the Gaussian mechanism as a baseline [DN03, DN04, DKM$^+$06]. For a function $f : \mathcal{U}^* \to \mathbb{R}^K$, the Gaussian mechanism releases $f(D) + Z$, where Z is a K-dimensional normally distributed random vector with independent coordinates, and variance $(\Delta f)^2$ at each coordinate.[2] Here Δf is the sensitivity of f, and equals $\max_{D \sim D'} \|f(D) - f(D')\|_2$. If we take f to be the function that maps D to the true answers to all marginal queries in $\mathcal{S}$, then it is easy to see that $\Delta f = \sqrt{|\mathcal{S}|}$, since adding or removing a data point can only affect $q_{S,t}(D)$ for a single partial assignment t, and for that t we have $|q_{S,t}(D) - q_{S,t}(D')| = 1$. We can improve over this baseline by correlating the Gaussian noise, as we discuss below.

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

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

[2] In the rest of the introduction we ignore the privacy parameters. Technically, we give the results for 1-GDP.

mean squared error. This is similar to error measures considered in prior work, e.g., by McKenna et al. [MMHM23], and Xiao et al. [XHZK23].

Our algorithm is inspired by the Fourier theoretic approach of [BCD+07]. 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. [BCD+07], 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 details in Section 1.3. The algorithm is described precisely as Algorithm 2, and its guarantees for weighted root mean squared error are given in Theorem 3.8.

Optimality among Factorization Mechanisms. This algorithm is an instance of the factorization mechanisms framework of Edmonds, Nikolov, and Ullman [ENU20], which itself generalizes the matrix mechanism of Li, Miklau, Hay, McGregor, and Rastogi [LMH+15].[3] 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 [MMHM23, HUU23, XHZK23, HU25, LUZ24, Leb25] (see also the recent survey [PUC+25]). The class of factorization mechanisms is also equivalent to mechanisms that add unbiased correlated Gaussian noise to the true query answers [NT24]. Furthermore, factorization mechanism are known to achieve nearly optimal error among all differentially private mechanisms in several important settings [NTZ16, ENU20, NT24]. For example, factorization algorithms are optimal up to absolute constants among unbiased differentially private estimates of the true query answers [NT24], and among all differentially private algorithms when the dataset size n is large enough [ENU20]. Similar techniques have also been used to reduce error for hierarchical queries [DGK+23]. Nevertheless, in other settings biased and data-dependent algorithms may achieve smaller error [HRMS10].

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 5.8.

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 [ENU20]. The resulting running times are polynomial in the number of queries $|Q_{\mathcal{S}}|$ and the universe size $|\mathcal{U}|$. In the case of marginal queries, however, $|\mathcal{U}|$ is exponential in d or

[3] 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 [LMH+15].

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 2. First 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 Theorem 3.11, and its optimality is proved in Theorem 5.8.

It is worth noting that, unlike the algorithm described above, this one is not a "closed form" solution, since it relies on an optimization routine to find p^*. Nevertheless, the algorithm still has the same structure of measuring Fourier queries and reconstructing marginal query answers using the inverse Fourier transform, and only the variance of the noise added to each Fourier query depends on p^*. Moreover, p^* can be found much more efficiently than optimizing over all factorization mechanisms.

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 [ENU20, NT24], or arbitrary differentially private mechanism [BUV14]. 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 $\bmod m_j$. 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 4.1 and Theorem 4.4 in Section 4.1, and the lower bound is in Theorem 5.14.

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 4.8). This approach is similar to the design of explicit factorization mechanisms for prefix (i.e., threshold) queries by embedding the into "circulant queries", i.e., queries that count points in an interval on a circle [CMRT23, HU25]. 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 [HU25]. 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 5.15). 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 [MNT20], 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. [XHZK23] 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 martices 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 [XHT$^+$25] (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 [XHZK23, XHT$^+$25], 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 Residual Planner [XHZK23] in the following ways:

- Running time: Reconstructing estimates of marginal queries is the bottleneck for the running time of our mechanisms. [XHZK23] 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. Computing the variance of any marginal estimate is similarly easy.
- 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 symmetry argument. They argue that the covariance matrix of the noise distribution of a factorization mechanism must satisfy certain symmetries, induced by the structure of marginal queries, and that ResidualPlanner finds an optimal mechanism with these symmetries. The technical details of this argument are fairly involved. 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 [XHT$^+$25] 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 [LM13], 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[4] 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 $1/\mu$.

The Gaussian mechanism would release estimates for all 2-way marginals by adding independent noise from $\mathcal{N}(0, \binom{d}{2})$ to each query. We aim to reduce the magnitude of noise by taking advantage of the inherent correlation between marginal queries.

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 [Leb25] 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.

[4]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.

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 $t \in \{0,1\}^{\{j,k\}}$,

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

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

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

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 3.3 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 affects 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. In Section 4 we extend our mechanism to product queries and extended marginals. In Section 5 we show lower

bounds for factorization mechanisms for answering weighted marginal and product queries that match our upper bounds, as well as a lower bound for extended marginals showing our mechanism is optimal up to lower order terms.

2 Preliminaries

2.1 Marginal Queries

We consider data points $x \in \mathcal{U}$ with d attributes where the i-th attribute has domain $\mathcal{U}_i \coloneqq \{0, 1, \ldots, m_i - 1\}$. In general, we have $\mathcal{U} \coloneqq \prod_{i \in [d]} \mathcal{U}_i$. 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 $t \in \mathcal{U}_S \coloneqq \prod_{i \in S} \mathcal{U}_i$. 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. [MSM19, MMS21]), and in deployments of differential privacy (e.g. [AAC+22]). 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 $t \in \mathcal{U}_S \coloneqq \prod_{i \in S} \mathcal{U}_i$. 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 $\phi = (\phi_1, \ldots, \phi_d)$, $\phi_j : \mathcal{U}_j \to \mathbb{R}$, 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 $\phi_j(z) \coloneqq \mathbb{1}\{z = 0\}$.

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 $T_i \coloneqq \{-m_i, \ldots, 0, \ldots, m_i - 1\}$ 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

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

We show how to embed extended marginals as product queries in Section 4.2.

2.2 Differential Privacy

Differential privacy [DMNS06] 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 exists, 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 2.1 (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 2.2 (Gaussian Differential Privacy [DRS22, Definition 4])**.** A randomized mechanism $\mathcal{M}{:}\mathcal{U}^* \to \mathcal{R}$ 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.

The Gaussian mechanism [DN03, DN04, DKM$^+$06] 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 2.3 (The Gaussian mechanism)**.** *Let $q{:}\mathcal{U}^* \to \mathbb{R}^d$ 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 2.4 (Gaussian noise for marginal queries)**.** *Let $\mathcal{S} = (S_1, \ldots, S_m)$ 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 2.3. □

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

Lemma 2.5 (Post-processing [DRS22, Proposition 4])**.** *Let $\mathcal{M}{:}\mathcal{U}^* \to \mathcal{R}$ denote any μ-GDP mechanism. Then for any (randomized) function $g{:}\mathcal{R} \to \mathcal{R}'$ the composed mechanism $g \circ \mathcal{M}{:}\mathcal{U}^* \to \mathcal{R}'$ also satisfies μ-GDP.*

Lemma 2.6 (Composition [DRS22, Corollary 3.3])**.** *Let $\mathcal{M}_1{:}\mathcal{U}^* \to \mathcal{R}_1$ and $\mathcal{M}_2{:}\mathcal{U}^* \to \mathcal{R}_2$ 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.*

2.3 Complex Numbers

For any complex number z, we use $\overline{z}$ for its complex conjugate, and

$$|z| = \sqrt{z\overline{z}} = \sqrt{\mathrm{Re}(a)^2 + \mathrm{Im}(a)^2}$$

for the absolute value. We recall the standard inner product over $\mathbb{C}^d$, defined as $\langle x, y\rangle = \sum_{i=1}^d x_i\overline{y_i}$. The standard ℓ_2 norm on $\mathbb{C}^d$ is $\|x\|_2 = \sqrt{\langle x, x\rangle} = \sqrt{\sum_{i=1}^d |x_i|^2}$. For a matrix A with complex entries, we use A^* for its conjugate transpose, i.e., $A^*_{i,j} = \overline{A_{j,i}}$ for all i and j.

Definition 2.7 (Roots of unity)**.** For any positive integer n, the n-th roots of unity is the set of complex numbers z satisfying $z^n = 1$. For the primitive n-th root of unity, we write

$$\omega_n \coloneqq e^{2\pi i/n} = \cos(2\pi/n) + i \cdot \sin(2\pi/n)\,.$$

Note that all roots of unity and their powers lie on the unit circle in $\mathbb{C}$, i.e., $|\omega_n| = 1$.

Lemma 2.8 (Multidimensional fast Fourier transform [CT65])**.** *Let $\mathcal{U} \coloneqq \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, where $\mathcal{U}_i \coloneqq \{0, \ldots, m_i - 1\}$, and let $x \in \mathbb{C}^{\mathcal{U}}$ be a d-dimensional vector of complex values indexed by $\mathcal{U}$. Consider for any $t \in \mathcal{U}$ the d-dimensional discrete Fourier transform (DFT)*

$$y_t = \sum_{a \in \mathcal{U}} x_a \prod_{i=1}^d \omega_{m_i}^{a_i t_i}\,.$$

All entries of $y \in \mathbb{C}^{\mathcal{U}}$ can be computed from x in time $O(m \log m)$ where $m = \prod_{i\in[d]} m_i$ using a multi-dimensional DFT (see e.g. [CLRS09, Section 30-3] for a construction).

Definition 2.9 (Complex Gaussian distribution)**.** We denote the complex zero-mean normal distribution as $\mathcal{CN}(0, \sigma^2)$, where $\sigma^2 \geq 0$ is the variance. If $Z \in \mathbb{C}$ is a sample from the distribution $Z \sim \mathcal{CN}(0, \sigma^2)$, then the real and imaginary parts of Z are distributed as independent samples from $\mathcal{N}(0, \sigma^2/2)$.

More generally, we denote the complex d-variate zero-mean normal distribution as $\mathcal{CN}(0, \Sigma)$, where the covariance matrix $\Sigma \in \mathbb{C}^{d\times d}$ is a Hermitian positive semidefinite matrix. $\mathcal{CN}(0, \Sigma)$ is the probability distribution on $\mathbb{C}^d$ with probability density function

$$p(z) \coloneqq \frac{1}{\pi^d \sqrt{\det(\Sigma)}} \exp\left(-\langle \Sigma z, z\rangle\right).$$

The following lemma, which is standard, will be useful in analyzing our algorithms.

Lemma 2.10. *Suppose that $Z \sim \mathcal{CN}(0, \Sigma)$ is a d-variate normally distributed random vector, and that A is an $\ell \times d$ matrix with complex entries. Then $AZ \sim \mathcal{CN}(0, A\Sigma A^*)$. In particular, if $Z_1, \ldots, Z_d$ are independent, and $Z_i \sim \mathcal{CN}(0, \sigma_i^2)$, and $a \in \mathbb{C}^d$ then $\sum_{i=1}^d a_i Z_i \sim \mathcal{CN}(0, \sum_{i=1}^d |a_i|^2 \sigma_i^2)$.*

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 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 are used for more marginal queries than others. By allocating privacy budget weighted by the importance of the queries, we can further reduce the error of our factorization. In Section 5, we show that our factorization is, in fact, optimal!

3.1 Marginal Queries in the Fourier Basis

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

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

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 $\mathcal{U}$. In the common special case where $|\mathcal{U}_1| = |\mathcal{U}_2| = \cdots = |\mathcal{U}_d| = m$, the queries take the form

$$F_a(D) := \sum_{i=1}^{|D|} \omega_m^{-\langle a,x \rangle}. \tag{2}$$

We denote the support of a by $\mathrm{supp}(a) \coloneqq \{j : a_j \neq 0\}$. The size of the support, i.e., the weight of a, is denoted by $\|a\|_0$ and is an important parameter.

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 Definition 2.7 for background). We use this fact later to bound the sensitivity for privatizing the queries. Note that in the special case where $a = 0$ all values of $\mathcal{U}$ evaluate to 1, so $F_0(D) = |D|$ is the dataset size.

Next, we show how we recover marginal queries using the aggregate Fourier queries above using the 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,

$$q_{S,t}(x) = \prod_{j \in S} \mathbb{1}\{t_j = x_j\} = \prod_{j \in S} \frac{\sum_{a=0}^{m_j-1} \omega_{m_j}^{at_j} \omega_{m_j}^{-ax_j}}{m_j} = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \chi_a(t)\overline{\chi_a(x)}. \tag{3}$$

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 $\mathrm{supp}(a) \subseteq S$, and χ_a only depends on coordinates in $\mathrm{supp}(a)$.

Summing over dataset points now gives us

$$q_{S,t}(D) = \sum_{i=1}^{|D|} q_{S,t}(x^{(i)}) = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \text{supp}(a) \subseteq S}} \chi_a(t) F_a(D)\,. \tag{4}$$

3.2 Warm-up: Estimating All k-way Marginals

In this subsection we show how to privately answer all k-way marginals using the Fourier representation discussed above. We focus on the setting where all attributes are defined on the same domain, such that $|\mathcal{U}_1| = |\mathcal{U}_2| = \cdots = |\mathcal{U}_d| = m$. We use this simpler problem as a warm-up before discussing the more general setting at the end of the section.

First, we show that we can privately release estimates of the complex-valued Fourier aggregate queries using complex Gaussian noise. Then, we show that correlating the noise across marginal queries by simply reusing estimates gives a utility improvement. We further improve the variance by distributing the privacy budget similar to [Leb25]. In fact, we recover Lebeda's mechanism in the simplest setting of estimating all 1-way marginals for binary attributes ($m = 2$).

We start with a basic lemma that the complex Gaussian mechanism satisfies GDP, since it is equivalent to running the standard real-valued Gaussian mechanism separately for the real and imaginary components.

Lemma 3.1. *Let $q{:}\mathcal{U}^* \to \mathbb{C}^d$ 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.*

Proof. Define an alternative query set $\hat{q}{:}\mathcal{U}^* \to \mathbb{R}^{2d}$ as

$$\hat{q}_i(D) = \begin{cases} \text{Re}(q_i(D)) & \text{if } i \le d, \\ \text{Im}(q_{i-d}(D)) & \text{otherwise.} \end{cases}$$

Then for any pair of datasets D and D' we have

$$\begin{aligned} \|\hat{q}(D) - \hat{q}(D')\|_2 &= \sqrt{\sum_{i \in [2d]} (\hat{q}(D)_i - \hat{q}(D')_i)^2} \\ &= \sqrt{\sum_{i \in [d]} (\hat{q}(D)_i - \hat{q}(D')_i)^2 + (\hat{q}(D)_{i+d} - \hat{q}(D')_{i+d})^2} = \|q(D) - q(D')\|_2\,, \end{aligned}$$

where the last equality follows from $(\hat{q}(D)_i - \hat{q}(D')_i)^2 + (\hat{q}(D)_{i+d} - \hat{q}(D')_{i+d})^2 = |q(D)_i - q(D')_i|^2$ by definition of $\hat{q}$. As such, we have $\Delta q = \Delta \hat{q}$ and we can release $\hat{q}(D) + \hat{Z}$ where $\hat{Z} \sim \mathcal{N}\left(0, \frac{(\Delta q)^2}{\mu^2} I_{2d}\right)$ under μ-GDP by Lemma 2.3. If we post processes $\hat{q}(D) + \hat{Z}$ by constructing complex numbers such that $\text{Re}(\tilde{q}_i(D)) = (\hat{q}(D) + \hat{Z})_i$ and $\text{Im}(\tilde{q}_i(D)) = (\hat{q}(D) + \hat{Z})_{i+d}$ then $\tilde{q}$ is distributed as $q(D) + Z$. The lemma therefore holds by the post processing property of GDP (Lemma 2.5). □

We now consider the setting where we want to privately estimate all k-way marginals. Notice in Equation (4) that queries $F_a(D)$ for which $\|a\|_0 \le |S|$ are 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. In total, to estimate all

k-way marginals, we need to estimate $F_a(D)$ for each a of weight $\|a\|_0 \leq k$, which brings the number of aggregate queries to

$$\sum_{j=0}^{k} \binom{d}{j} (m-1)^j .$$

We can then estimate any k-way marginal query by post-processing private estimates of the m^k relevant Fourier queries. As a final step, we always remove any imaginary part of the estimate, since that must come from noise. Estimating all $F_a(D)$ queries privately and reusing values already gives us a slight improvement over the baseline where the error is distributed as $\mathcal{N}(0, \binom{d}{k}/\mu^2)$ since $m^{-k} \sum_{j=0}^{k} \binom{d}{j}(m-1)^j \leq \binom{d}{k}$. However, we can further reduce the error by releasing more accurate answers to Fourier coefficient that are reused. We scale the variance of each estimate inversely proportional to the square root of the number of queries it is used for. This scaling is optimized for ℓ_2^2 error with Gaussian noise as shown for different settings in [ALNP24, DJY+24, Leb25]. The pseudocode of our mechanism is in Algorithm 1.

Algorithm 1 Differentially private estimates of all k-way marginals.

For each $a \in \mathcal{U}$ of weight $\|a\|_0 \leq k$, compute the value

$$F_a(D) := \sum_{i=1}^{|D|} \omega_m^{-\langle a, x^{(i)} \rangle}.$$

Let

$$\tau := \frac{1}{\mu^2} \sum_{j=0}^{k} \binom{d}{j} (m-1)^j \sqrt{\binom{d-j}{k-j}} \ .$$

Privately release estimate of each $F_a(D)$, $\|a\|_0 \leq k$, by adding independent noise such that

$$\tilde{F}_a(D) = F_a(D) + Z_a, \text{ where } Z_a \sim \mathcal{CN}\left(0, 2\tau \Big/ \sqrt{\binom{d-\|a\|_0}{k-\|a\|_0}}\right) \ .$$

Estimates for k-way marginal queries can be recovered using post-processing by computing

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

Next, we show the privacy properties and error of our mechanism. Later we discuss how to speed up computation time over a trivial implementation by computing estimates using FFT.

Lemma 3.2. *Algorithm 1 satisfies μ-GDP.*

Proof. Since the released estimates $\tilde{q}_{S,t}(D)$ are just post-processing of the Fourier query estimates $\tilde{F}_a(D)$, it is enough to show that releasing $\tilde{F}_a(D)$ for all $a \in \mathcal{U}$ of weight $\|a\|_0 \leq k$ satisfied μ-GDP. We show this via composition. Observe that, since the sensitivity of $F_a(D)$ is 1, by Lemma 3.1 releasing $\tilde{F}_a(D)$ satisfies μ_a-GDP, where $\mu_a^2 := \sqrt{\binom{d-\|a\|_0}{k-\|a\|_0}} \Big/ \tau$. Moreover, notice that there are $\binom{d}{\ell}(m-1)^\ell$ choices of a with weight $\|a\|_0 = \ell$. Therefore, the lemma

follows from the composition property of GDP (Lemma 2.6), since

$$\sum_{a:\|a\|_0\le k}\mu_a^2=\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell\frac{\sqrt{\binom{d-\ell}{k-\ell}}}{\tau}=\mu^2. \qquad \square$$

Lemma 3.3. *For any $S\subseteq[d]$, $|S|=k$, and any $t\in\{0,\dots,m-1\}^S$, the estimate $\tilde{q}_{S,t}(D)$ computed by Algorithm 1 has error $\tilde{q}_{S,t}(D)-q_{S,t}(D)$ distributed as $\mathcal{N}(0,\sigma^2)$, where*

$$\sigma=\frac{1}{\mu m^k\sqrt{\binom{d}{k}}}\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell\sqrt{\binom{d-\ell}{k-\ell}}.$$

Proof. Let $\tilde{q}'_{S,t}(D)$ be $\tilde{q}_{S,t}(D)$ without removing the imaginary part. By equation (4),

$$\tilde{q}'_{S,t}(D)-q_{S,t}(D)=\frac{1}{m^k}\sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}}\chi_a(t)Z_a.$$

Therefore, by Lemma 2.10, and since $|\chi_a(t)|=1$, $\tilde{q}'_{S,t}(D)-q_{S,t}(D)\sim\mathcal{CN}(0,\sigma_{\mathbb{C}}^2)$, where

$$\sigma_{\mathbb{C}}^2=\frac{2\tau}{m^{2k}}\sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}}\frac{1}{\sqrt{\binom{d-\|a\|_0}{k-\|a\|_0}}}=\frac{2\tau}{m^{2k}}\sum_{\ell=0}^{k}\frac{\binom{k}{\ell}(m-1)^\ell}{\sqrt{\binom{d-\ell}{k-\ell}}}.$$

The second equality holds because there are $\binom{k}{\ell}(m-1)^\ell$ choices of a with $\mathrm{supp}(a)\subseteq S$ and $\|a\|_0=\ell$. Using the identity $\binom{d}{k}\binom{k}{\ell}=\binom{d}{\ell}\binom{d-\ell}{k-\ell}$, and plugging in the value of τ, we get

$$\sigma_{\mathbb{C}}^2=\frac{2\tau}{m^{2k}}\sum_{\ell=0}^{k}\frac{\binom{d}{\ell}(m-1)^\ell\sqrt{\binom{d-\ell}{k-\ell}}}{\binom{d}{k}}=\frac{2}{\mu^2m^{2k}\binom{d}{k}}\left(\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell\sqrt{\binom{d-\ell}{k-\ell}}\right)^2.$$

The lemma now follows after observing that $\tilde{q}_{S,t}(D)-q_{S,t}(D)$ is distributed as the real part of $\tilde{q}'_{S,t}(D)-q_{S,t}(D)$ which has variance $\sigma^2=\frac{1}{2}\sigma_{\mathbb{C}}^2$. $\square$

The improvement of Lemma 3.3 over the Gaussian mechanism baseline with standard deviation $\sqrt{\binom{d}{k}}/\mu$ depends on the parameters d, k, and m. When $d=k$ both mechanisms have $\sigma=1/\mu$, and for fixed k and large m and d we have $\sigma\approx((m-1)/m)^k\sqrt{\binom{d}{k}}/\mu$ in Lemma 3.3. The intuition behind this improvement factor is that $(m-1)^k$ of the coefficients used in Equation (4) are unique to S, while the remaining $m^k-(m-1)^k$ coefficients are reused for other marginals. When d is sufficiently large, we can estimate these remaining coefficients with little noise at only a small privacy cost, because they are reused in many marginals queries. We plot the relative improvement over the baseline for small parameters in Figure 1.

Next, we discuss the computation time of Algorithm 1. We focus on the time for adding noise, since all baselines (even non-private solutions) must compute all $q_{S,t}(D)$. Note that the mechanism that adds i.i.d. noise to each marginal query runs in time $O(\lambda\binom{d}{k}m^k)$, where $O(\lambda)$ is the time needed for sampling a standard Gaussian[5]. Our mechanism uses fewer Gaussian

[5]In practice, implementations for (approximately) sampling a standard Gaussian often have randomized running time (e.g. [CKS22]). For simplicity, we assume that sampling runs in time $O(\lambda)$ deterministically. Our results technically only bound the expected running time of our algorithms.

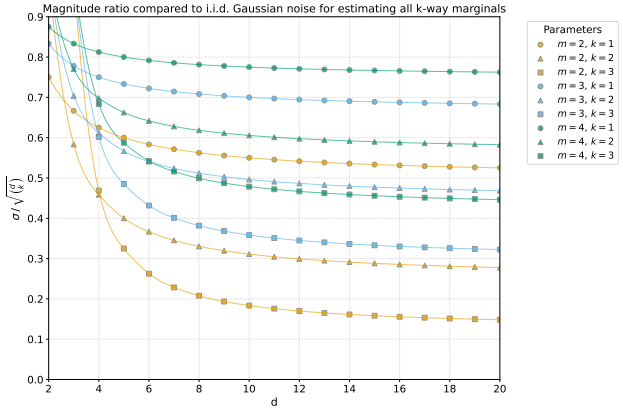

Fig. 1. Comparison for small values of k, m, and d of the standard deviation from Theorem 3.5 relative to the baseline that adds i.i.d. noise with magnitude $\sqrt{\binom{d}{k}}$. When d increases the improvement ratio approaches $(1-1/m)^k$.

samples because we reuse samples across marginals. However, we combine m^k samples for each marginal estimate. This results in a running time overhead of $O(\binom{d}{k}m^{2k}k)$ using a straightforward implementation. We can speed up the computation time if we compute all assignments for each S at the same time using FFT.

Lemma 3.4. *Assume that we sample a standard Gaussian in time $O(\lambda)$, and that we have access to all non-private k-way marginals $q_{S,t}(D)$. Then all private k-way marginal estimates $\tilde{q}_{S,t}(D)$ from Algorithm 1 can be computed in time $O(\binom{d}{k}m^k k\log(m)+\lambda\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell)$.*

Proof. It is easy to see that only $2\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell$ Gaussian samples are needed. Thus, it suffices to prove that for any fixed S of size k, we can compute all marginal queries with support S in time $O(m^k k\log m)$ given access to the relevant Gaussian samples. This is a standard multi-dimensional FFT, which can be computed by a multi-dimensional FFT algorithm within time $O(m^k k\log(m))$ (see Lemma 2.8). □

We are now ready to state one of our main results by summarizing the properties of our technique.

Theorem 3.5. *Let D be a dataset containing data points $x\in\mathcal{U}$, where $\mathcal{U}:=\{0,1,\dots,m-1\}^d$. Then there exists a μ-GDP mechanism that estimates all k-way marginal queries of D, where the error of each estimate of $q_{S,t}(D)$ for $|S|=k$ is distributed as $\mathcal{N}(0,\sigma^2)$ where*

$$\sigma=\frac{1}{\mu m^k\sqrt{\binom{d}{k}}}\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell\sqrt{\binom{d-\ell}{k-\ell}}\ .$$

Additionally, the noise for all marginal estimates can be sampled in time $O(\binom{d}{k}m^k k\log(m)+\lambda\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell)$ where $O(\lambda)$ is the time required for sampling a standard Gaussian.

Proof. That mechanism is Algorithm 1. The privacy guarantees, error distribution, and running time follow from Lemma 3.2, 3.3, and 3.4, respectively. □

3.3 Estimating Arbitrary Marginal Query Workloads

We now consider a general set up for answering arbitrary workloads of marginal queries. Recall that the 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

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

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. The error definition allows for zero-weight queries. This is required when we optimize for maximum variance later in the section. Whenever a zero-weight query is a subset of a non-zero weight query we can release an unbiased estimate at no additional privacy cost. This is the case e.g. when releasing all k-way marginals with Algorithm 1, as we can estimate all $1, 2, \ldots, (k-1)$-way marginals from the Fourier queries. In the rest of the section we do not explicitly handle zero-weight queries that are not a subset of a non-zero weight query. We cannot release an unbiased estimate for these queries but since they do not affect the error measure it does not matter how they are estimated.

Our algorithm in this setting is an extension of Algorithm 1. We answer all Fourier aggregate queries F_a necessary for reconstructing the answers to queries in $Q_{\mathcal{S}}$. The amount of noise added to F_a is proportional to how much this noise contributes to $\mathrm{err}_p(q, \tilde{q})^2$ in expectation. The full description is given in Algorithm 2. In Section 5 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 the proofs of the privacy and error guarantees.

It is easy to check that Algorithm 1 is a special case of Algorithm 2 when $\mathcal{S}$ consists of all subsets of $[d]$ of cardinality k, and $p(S) = \frac{1}{\binom{d}{k}}$ for all $S \in \mathcal{S}$.

As for the intuition behind τ, note that the error $\tilde{q}_{S,t}(D) - q_{S,t}(D)$ is a sum of $|\mathcal{U}_S|$ random variables. Specifically, each F_a where $\mathrm{supp}(a) \subseteq S$ contributes a random variable to the sum with variance

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

where σ_a^2 is half the variance of Z_a. Since the noise variables 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 variances for each of the $|\mathcal{U}_S|$ random variables. Likewise, the noise of the estimate for F_a contributes to the error for all marginals where $\mathrm{supp}(a) \subseteq S$. In total, the private estimate of the query $F_a(D)$ adds

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

Algorithm 2 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 the expectation of $\text{err}_p(q, \tilde{q})^2$. As such, we set τ_a to the square root of the weight for $\tilde{F}_a(D)$. This leads to the optimal values for all σ_a when minimizing $\text{err}_p(q, \tilde{q})^2$. See the work of Lebeda and Pagh for more details on calibrating Gaussian noise for ℓ_2^2 error when queries have different scales [Leb23, Chapter 4].

We first give the privacy guarantees.

Lemma 3.6. *Algorithm 2 satisfies μ-GDP.*

Proof. The proof is analogous to that of Lemma 3.2. We argue that releasing the Fourier query estimates $\tilde{F}_a(D)$ satisfies μ-GDP, and then the privacy of the estimates $\tilde{q}_{S,t}(D)$ follows by post-processing. Once again, Lemma 3.1 gives us that $\tilde{F}_a(D)$ satisfies μ_a-GDP, where $\mu_a^2 := \frac{\tau_a}{\tau}$. Now the lemma follows from composition and the definition of τ. □

Next we compute the error of the algorithm. Recall that 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.

Lemma 3.7. *The estimates $\tilde{q}_{S,t}(D)$ computed by Algorithm 2 for $S \in \mathcal{S}$ and $t \in \mathcal{U}_S$ have weighted root mean squared error*

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

PROOF. Take some $S \in \mathcal{S}$ for which $p(S) > 0$, and some $t \in \mathcal{U}_S$. By equation (4),

$$\tilde{q}_{S,t}(D) - q_{S,t}(D) = \mathrm{Re}\left(\frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \chi_a(t) Z_a\right).$$

By Lemma 2.10, $\frac{1}{|\mathcal{U}_S|}\sum_a \chi_a(t) Z_a$ is a complex mean zero normal random variable, and its real part is also normally distributed with half the variance. Again by Lemma 2.10, and since $|\chi_a(t)| = 1$, we get that $\tilde{q}_{S,t}(D) - q_{S,t}(D) \sim \mathcal{N}(0, \sigma_S^2)$, where

$$\sigma_S^2 = \frac{\tau}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \frac{1}{\tau_a} = \frac{\tau}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \left(\sum_{\substack{T \in \mathcal{S} \\ \mathrm{supp}(a) \subseteq T}} \frac{p(T)}{|\mathcal{U}_T|^2}\right)^{-1/2}. \tag{5}$$

Here we used the observation that $\tau_a > 0$ for any a such that $\mathrm{supp}(a) \subseteq S$, since $p(S) > 0$ contributes a positive amount to τ_a. Then the average squared error, weighted by p, is

$$\mathrm{err}_p(q, \tilde{q})^2 = \sum_{S \in \mathcal{S}} p(S)\sigma_S^2 = \tau \sum_{S \in \mathcal{S}} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \frac{p(S)}{|\mathcal{U}_S|^2} \left(\sum_{\substack{T \in \mathcal{S} \\ \mathrm{supp}(a) \subseteq T}} \frac{p(T)}{|\mathcal{U}_T|^2}\right)^{-1/2}.$$

Let $\mathcal{S}_{p,\downarrow}$ be the collection of all $R \subseteq [d]$ such that $R \subseteq S$ for some $S \in \mathcal{S}$ with $p(S) > 0$.[6] Changing the order of summation above, we get

$$\begin{aligned}\mathrm{err}_p(q, \tilde{q})^2 &= \tau \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_{p,\downarrow}}} \sum_{S \supseteq \mathrm{supp}(a)} \frac{p(S)}{|\mathcal{U}_S|^2} \left(\sum_{T \supseteq \mathrm{supp}(a)} \frac{p(T)}{|\mathcal{U}_T|^2}\right)^{-1/2} \\ &= \tau \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{S \supseteq \mathrm{supp}(a)} \frac{p(S)}{|\mathcal{U}_S|^2}}.\end{aligned}$$

After plugging in the value of τ, and recalling that, for any R, there are $\prod_{j \in R}(m_j - 1)$ choices of a with $\mathrm{supp}(a) = R$, we have

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

□

We now may conclude with the following theorem:

[6] It is natural to assume that $p(S) > 0$ for all $S \in \mathcal{S}$, in which case $\mathcal{S}_{p,\downarrow} = \mathcal{S}_\downarrow$. We do not make this assumption here because later we optimize over choices of p for a fixed $\mathcal{S}$.

THEOREM 3.8. *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*

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

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 error bound is already proven in Lemma 3.7. The time complexity of sampling noise is straightforward with FFT, with the same argument as in Lemma 3.4. □

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 2, with "worst case" choice of p, i.e., the p that maximizes the error in Lemma 3.7. The privacy of this algorithm follows directly from Lemma 3.6. We also have the following error bound.

Lemma 3.9. *Let $\tilde{q}_{S,t}(D)$ be the estimates produced by Algorithm 2 with weights*

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

Then,

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

Lemma 3.9 follows easily from the following lemma, which gives first order optimality conditions for the optimization problem in (6).

Lemma 3.10. *The function $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ defined in (6) is such that $\sum_{\substack{S \in \mathcal{S} \\ S \supseteq R}} \frac{p(S)}{|\mathcal{U}_S|^2} > 0$ for all $R \in \mathcal{S}_\downarrow$. Moreover, for any $S \in \mathcal{S}$, and any $S' \in \mathcal{S}$ such that $p^*(S') > 0$, we have the inequality*

$$\frac{1}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \left(\sum_{\substack{T \in \mathcal{S} \\ T \supseteq \mathrm{supp}(a)}} \frac{p^*(T)}{|\mathcal{U}_T|^2} \right)^{-1/2} \leq \frac{1}{|\mathcal{U}_{S'}|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S'}} \left(\sum_{\substack{T \in \mathcal{S} \\ T \supseteq \mathrm{supp}(a)}} \frac{p^*(T)}{|\mathcal{U}_T|^2} \right)^{-1/2}.$$

PROOF. Let us denote the objective in the optimization problem in (6) by

$$f(p) := \sum_{R \in \mathcal{S}_\downarrow} f_R(p) \quad \text{where} \quad f_R(p) := \left(\prod_{j \in R} (m_j - 1) \right) \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq R}} \frac{p(S)}{|\mathcal{U}_S|^2}}.$$

Let p^* be as in (6). We first argue that $f_R(p^*) > 0$ for all $R \in \mathcal{S}_\downarrow$, which is equivalent to the first claim. Assume this were not the case for some $R \in \mathcal{S}_\downarrow$, and consider increasing $p^*(S)$ by a small amount β for some $S \in \mathcal{S}$ that contains R, and decreasing $p^*(S')$ by β for some $S' \in \mathcal{S}$ such that $p^*(S') > 0$. Increasing $p^*(S)$ increases $f_R(p^*)$, and thus also $f(p^*)$, by at least $c\sqrt{\beta}$ for a constant $c > 0$ independent of β (but possibly dependent on $\mathcal{S}$). At the same time, decreasing $p^*(S')$ only affects $f_R(p^*)$ when $R \subseteq S'$. For any such R, $p^*(S') > 0$ implies $f_R(p^*) > 0$, and, therefore, $\frac{\partial f_R(p^*)}{\partial p(S')}(p^*)$ exists, and $f_R(p^*)$ decreases by at most $c'\beta$ for all small enough β and some constant c' independent of β. From this, we conclude that decreasing $p^*(S')$ decreases $f(p^*)$ by at most $c''\beta$ for another constant c'' and all small enough β. The overall change in $f(p^*)$ is thus at least $c\sqrt{\beta} - c''\beta > 0$ as long as β is small enough, contradicting the optimality of p^*.

Let us now choose any $S' \in \mathcal{S}$ such that $p^*(S') > 0$. For any $S \in \mathcal{S}$, first order optimality conditions imply that

$$\frac{\partial f}{\partial p(S)}(p^*) - \frac{\partial f}{\partial p(S')}(p^*) \leq 0, \tag{7}$$

where the fact that $f_R(p^*) > 0$ for all $R \in \mathcal{S}_\downarrow$ guarantees that the partial derivatives exist at p^*. Indeed, if (7) did not hold, then adding β to $p^*(S)$ and subtracting β from $p^*(S')$ for a small enough $\beta > 0$ would increase the value of f, contradicting the optimality of p^*. We calculate the partial derivatives as

$$\begin{aligned}
\frac{\partial f}{\partial p(S)}(p^*) &= \frac{1}{2|\mathcal{U}_S|^2} \sum_{\substack{R \subseteq [d] \\ R \subseteq S}} \left(\prod_{j \in R} (m_j - 1) \right) \left(\sum_{\substack{T \in \mathcal{S} \\ T \supseteq R}} \frac{p^*(T)}{|\mathcal{U}_T|^2} \right)^{-1/2} \\
&= \frac{1}{2|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \operatorname{supp}(a) \subseteq S}} \left(\sum_{\substack{T \in \mathcal{S} \\ T \supseteq \operatorname{supp}(a)}} \frac{p^*(T)}{|\mathcal{U}_T|^2} \right)^{-1/2},
\end{aligned}$$

where the second equality holds because the number of choices of a with $\operatorname{supp}(a) = R$ equals $\prod_{j \in R}(m_j - 1)$. Plugging back into (7) and multiplying both sides by 2 completes the proof. □

PROOF OF LEMMA 3.9. Recalling (5), Lemma 3.10 implies that $\sigma_S^2 \leq \sigma_{S'}^2$ for any $S \in \mathcal{S}$, and any $S' \in \mathcal{S}$ such that $p(S') > 0$, where $\sigma_S^2 = \mathbb{E}[(\tilde{q}_{S,t}(D) - q_{S,t}(D))^2]$ for any $t \in \mathcal{U}_S$. Therefore,

$$\operatorname{err}_{p^*}(q, \tilde{q})^2 = \sum_{S' \in \mathcal{S}} \frac{p^*(S')}{|\mathcal{U}_{S'}|} \sum_{t \in \mathcal{U}_{S'}} \mathbb{E}[(\tilde{q}_{S',t}(D) - q_{S',t}(D))^2] = \sum_{S' \in \mathcal{S}} p^*(S') \sigma_{S'}^2 = \max_{S \in \mathcal{S}} \sigma_S^2.$$

The lemma now follows from Lemma 3.7. □

The following theorem is our main result for maximum error when releasing answers to general workloads of marginal queries.

THEOREM 3.11. *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 workload $Q_{\mathcal{S}}$ of marginal queries on 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 $t \in \mathcal{U}_S := \prod_{i \in S} \mathcal{U}_i$. Let $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ be as defined in (6). There exists a μ-GDP mechanism that estimates all marginal queries in $Q_{\mathcal{S}}$ and satisfies*

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

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 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 weight function p^* can be computed is the solution to a concave maximization problem in $|\mathcal{S}|$ variables, and the objective and its gradients can be computed to arbitrary precision in time polynomial in $|\mathcal{S}_\downarrow| + \max_{S \in \mathcal{S}} |S|$ (see, e.g., [Bub15]). Therefore, p^* can be computed within polynomial time in $|\mathcal{S}_\downarrow| + \max_{S \in \mathcal{S}} |S|$. The error bound was proven in Lemma 3.9. The time complexity of sampling noise is straightforward with FFT, with the same argument as in Lemma 3.4. □

Assuming, without loss of generality that $m_i > 1$ for each $i \in [d]$, one can check that

$$|\mathcal{S}_\downarrow| \leq \sum_{R \in \mathcal{S}_\downarrow} \prod_{i \in R} (m_i - 1) \leq \sum_{S \in \mathcal{S}} |\mathcal{U}_S| = |Q_{\mathcal{S}}|,$$

and the running time bound above is certainly polynomial in the number of queries.

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

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

Remark 3.12. One may also consider error measures that interpolate between weighted root mean squared error and maximum variance, as was done in [NT24, LUZ24]. For example, we can consider, for an exponent $r \geq 1$, the error measure

$$\mathbb{E}\left[\sum_{S \in \mathcal{S}, t \in \mathcal{U}_S} |\tilde{q}_{S,t}(D) - q_{S,t}(D)|^{2r} \right]^{1/2r} \lesssim \sqrt{r} \left(\sum_{S \in \mathcal{S}, t \in \mathcal{U}_S} \mathbb{E}[|\tilde{q}_{S,t}(D) - q_{S,t}(D)|^2]^r \right)^{1/2r},$$

where the inequality follows from standard Gaussian moment bounds - see [NT24, LUZ24] for the details. The quantity $\left(\sum_{S \in \mathcal{S}, t \in \mathcal{U}_S} \mathbb{E}[|\tilde{q}_{S,t}(D) - q_{S,t}(D))|^2]^r \right)^{1/r}$ is the ℓ_r norm of the vector of noise variances for each query. We can derive a mechanism minimizing this error measure analogously to the proof of Theorem 3.11. In particular, we can show an analogous

result to Lemma 3.9, with p^* defined as

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

where $r' = \frac{r}{r-1}$. We can then show that, for this choice of p^*,

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

More generally, we can extend this approach to any error measure which is a norm of the vector of noise variances by defining p^* as the solution to an analogous optimization problem where the optimization is over the dual norm ball. We omit the details of these extensions from this paper, and focus on the two most commonly studied and natural error measures - the weighted root mean squared error, and the maximum variance measures.

4 Upper Bounds for Product Queries and Extended Marginals

In this section, we first introduce a generalization of marginal queries, which we call product queries. Our algorithms extend, with only small modifications, to answering arbitrary workloads of product queries. Later, in Section 5.4, we also show that the resulting upper bounds are optimal within the class of factorization mechanisms. We further show how to use product queries to answer extended marginal queries, in which the data points have both categorical and numerical attributes, and the predicates on numerical attributes are threshold functions.

4.1 Estimating Product Queries

Let us now extend our Fourier queries framework to product queries, as defined in Section 2.1. For any $j \in [d]$, and $a \in \mathcal{U}_j$, let

$$\widehat{\phi_j}(a) := \sum_{z=0}^{m_j - 1} \phi_j(z) \omega_{m_j}^{-az} \tag{8}$$

be the Fourier coefficient of ϕ_j corresponding to a. Then, by the inverse (discrete) Fourier transform, for any $t, z \in \mathcal{U}_j$ we have

$$\phi_j((t - z) \bmod m_j) = \frac{1}{m_j} \sum_{a=0}^{m_j - 1} \widehat{\phi_j}(a) \omega_{m_j}^{a(t-z)}. \tag{9}$$

Therefore,

$$q^{\phi}_{S,t}(x) = \prod_{j \in S} \left(\frac{1}{m_j} \sum_{a_j=0}^{m_j - 1} \widehat{\phi_j}(a_j) \omega_{m_j}^{a_j \cdot (t_j - x_j)} \right) = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \left(\prod_{j \in S} \widehat{\phi_j}(a_j) \right) \chi_a(t) \overline{\chi_a(x)}, \tag{10}$$

and, summing over data points,

$$q^{\phi}_{S,t}(D) = \frac{1}{|\mathcal{U}_S|} \sum_{\substack{a \in \mathcal{U} \\ \text{supp}(a) \subseteq S}} \left(\prod_{j \in S} \widehat{\phi_j}(a_j) \right) \chi_a(t) F_a(D). \tag{11}$$

This is a generalization of our formulas for marginals, since when $\phi_j(z) = \mathbb{1}\{z = 0\}$, then $\widehat{\phi_j}(a) = 1$ for all $a \in \mathcal{U}_j$.

Algorithm 3 Algorithm for estimating a workload $Q^{\phi}_{\mathcal{S}}$ of weighted product queries.

For any $a \in \mathcal{U}$, define $\widehat{\phi_1}(a_1), \ldots, \widehat{\phi_d}(a_d)$ as in Equation (8), and let

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

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

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

The privacy proof is analogous to the proof of Lemma 3.6. For the error analysis, observe that, reasoning as we did with Algorithm 2, we have that for any S and $t \in \mathcal{U}_S$, $\tilde{q}^{\phi}_{S,t} - q^{\phi}_{S,t}$ is distributed as a real Gaussian with mean 0 and variance

$$\sigma_S^2 := \frac{\tau}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \text{supp}(a) \subseteq S}} \left(\prod_{j \in S} |\widehat{\phi_j}(a_j)|^2 \right) \frac{1}{\tau_a}.$$

The rest of the error analysis is also analogous to the proof of Lemma 3.7.

To implement Algorithm 3 efficiently, we first compute, for each $j \in [d]$ that is contained in some $S \in \mathcal{S}$, the Fourier coefficients $\widehat{\phi_j}(0), \ldots \widehat{\phi_j}(m_j - 1)$ using standard (one-dimensional) FFT in time $O(m_j \log m_j)$. The total time to compute all the Fourier coefficients is then $O\left(\sum_{j:\{j\} \in \mathcal{S}_\downarrow} m_j \log(m_j)\right)$. Then, we compute the noisy Fourier query answers $\tilde{F}_a(D)$ for those a such that $\tau_a > 0$, which is a subset of those a such that $\text{supp}(a) \in \mathcal{S}_\downarrow$. Finally, for any $S \in \mathcal{S}$, to reconstruct $\tilde{q}^{\phi}_{S,t}$ for all $t \in \mathcal{U}_S$, we apply the multi-dimensional FFT (Lemma 2.8) to the values $\prod_{j \in S} \widehat{\phi_j}(a_j) \tilde{F}_a(D)$, allowing us to perform the reconstruction in time $O(|\mathcal{U}_S| \log(|\mathcal{U}_S|))$.

In summary, we have the following theorem, generalizing Theorem 3.8.

THEOREM 4.1. *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*

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

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 4.2. *Let $\tilde{q}^\phi_{S,t}(D)$ be the estimates produced by Algorithm 3 with weights*

$$p^* \in \arg\max \left\{ \sum_{\substack{a \in \mathcal{U} \\ \operatorname{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \operatorname{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}} : \sum_{S \in \mathcal{S}} p(S) = 1, p(S) \geq 0 \ \forall S \in \mathcal{S} \right\}. \tag{12}$$

Then,

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

The proof of Lemma 4.2, which is analogous to the proof of Lemma 3.9, relies on the following lemma, whose proof is analogous to the proof of Lemma 3.10.

Lemma 4.3. *The function $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ defined in (12) is such that*

$$\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \operatorname{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2} > 0$$

for all $a \in \mathcal{U}$ for which there is at least one $S \in \mathcal{S}$ satisfying $S \supseteq \mathrm{supp}(a)$ and $\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2 \neq 0$. Moreover, for any $S \in \mathcal{S}$, and any $S' \in \mathcal{S}$ such that $p^*(S') > 0$, we have the inequality

$$\frac{1}{|\mathcal{U}_S|^2} \sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}} \left(\sum_{\substack{T\in\mathcal{S}\\ T\supseteq \mathrm{supp}(a)}} \frac{p^*(T)\prod_{j\in T}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_T|^2} \right)^{-1/2} \leq \frac{1}{|\mathcal{U}_{S'}|^2} \sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S'}} \left(\sum_{\substack{T\in\mathcal{S}\\ T\supseteq \mathrm{supp}(a)}} \frac{p^*(T)\prod_{j\in T}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_T|^2} \right)^{-1/2}.$$

The next Theorem follows from Lemma 4.2.

THEOREM 4.4. *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 (12). 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 4.5. 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.

4.2 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

$$q_{S,t}(x) := \left(\prod_{j \in S \cap C} \mathbb{1}\{x_j = t_j\} \right) \left(\prod_{\substack{j \in S \cap N \\ t_j \geq 0}} \mathbb{1}\{x_j \leq t_j\} \right) \left(\prod_{\substack{j \in S \cap N \\ t_j < 0}} \mathbb{1}\{x_j \geq |t_j|\} \right). \tag{13}$$

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 $\mathbb{1}\{x_j \geq 0\}$ suffix query because it is equivalent to a $\mathbb{1}\{x_j \leq m_j - 1\}$ prefix query. Given a collection of subsets $\mathcal{S}$ of $[d]$, let us define $Q_{\mathcal{S}}^{\text{pr-suf}}$ 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: $Q_{\mathcal{S}}^{\text{pref}}$ contains all queries $q_{S,t}$ for $S \in \mathcal{S}$ and $t \in \mathcal{U}_S$.

Remark 4.6. When $t_i = m_i - 1$, $\mathbb{1}\{x_i \leq t_i\}$ always evaluates to 1, and when $t_i = -m_i$, $\mathbb{1}\{x_i \geq |t_i|\}$ always evaluates to 0. It is possible to achieve slightly stronger results by removing these values. Then 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 [CMRT23], and is implicit in the group algebra based factorization mechanism of Henzinger and Upadhyay [HU25] (see also Section 5.1 of [HKU25]).

Lemma 4.7. *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 $\phi_i(z) := \mathbb{1}\{z = 0\}$ for $i \in C$, and $\phi_i(z) := \mathbb{1}\{z \leq m_i - 1\}$ for $i \in N$. Notice that, for $t, z \in \{0, \ldots, 2m_i - 1\}$,

$$(t - z) \bmod 2m_i = \begin{cases} t - z & z \leq t \\ 2m_i + (t - z) & z > t \end{cases}. \tag{14}$$

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

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

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

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

We can then use Algorithm 3 to answer the extended marginal queries $Q_{\mathcal{S}}$. Doing so gives us the guarantees in the following theorem. In Section 5.5 we give a lower bound for $Q_{\mathcal{S}}^{\text{pref}}$. 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 $Q_{\mathcal{S}}^{\text{pref}}$. Nevertheless, no optimal construction is known even for a single prefix query despite significant attention from the research community (see [HKU25] for the current best known bounds).

THEOREM 4.8. *Let D be a dataset containing data points $x \in \mathcal{U}$, where $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, $\mathcal{U}_i := \{0, \ldots, m_i - 1\}$, 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 $Q_{\mathcal{S}}^{pr\text{-}suf}$ defined above, and a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, there exists a μ-GDP mechanism that estimates all queries in $Q_{\mathcal{S}}^{pr\text{-}suf}$ and satisfies*

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

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

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

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 $Q_{\mathcal{S}}^{pref}$.

PROOF. The proof is the same for $Q_{\mathcal{S}}^{\text{pr-suf}}$ and $Q_{\mathcal{S}}^{\text{pref}}$, since the noise variance $\mathbb{E}[(\tilde{q}_{S,t}(D) - q_{S,t}(D))^2]$ is independent of t for each S. Moreover, the proofs for weighted mean squared error and maximum variance are the same, and only differ in whether we use Theorem 4.1 or Theorem 4.4. We present the proof for weighted root mean squared error.

The privacy and running time guarantees follow directly from Theorems 4.1 and 4.4. For error, we get that for any weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, the weighted root mean squared

error is

$$\frac{1}{\mu} \sum_{\substack{a \in \mathcal{U}' \\ \mathrm{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|T_S|^2}}.$$

To evaluate this expression, we need to compute $\widehat{\phi_j}(a_j)$. When $j \in C$, we have $\widehat{\phi_j}(a) = 1$. When $j \in N$, we have $\widehat{\phi_j}(0) = m_j$, and, for $a_j > 0$,

$$\widehat{\phi_j}(a_j) = \sum_{z=0}^{m_j - 1} \omega_{2m_j}^{-a_j \cdot z} = \frac{1 - \omega_{2m_j}^{-a_j \cdot m_j}}{1 - \omega_{2m_j}^{-a_j}}; \qquad |\widehat{\phi_j}(a_j)|^2 = \frac{|1 - \omega_{2m_j}^{-a_j \cdot m_j}|^2}{|1 - \omega_{2m_j}^{-a_j}|^2} = \frac{1 - \cos(\pi a_j)}{1 - \cos\left(\frac{\pi a_j}{m_j}\right)}.$$

Since $\omega_{2m_j}^{-a_j} \neq 1$ for $a_j > 0$, we could use the standard formula for a finite geometric series in the left equation. In the last equality we used the identity $|1 - \exp(i\theta)|^2 = 2(1 - \cos(\theta))$. Note that when $a_j \neq 0$ is even, $\cos(\pi a_j) = 1$, and when a_j is odd, $\cos(\pi a_j) = -1$, so

$$|\widehat{\phi_j}(a_j)|^2 = \frac{2 \cdot \mathbb{1}\{a_j \text{ is odd}\}}{1 - \cos\left(\frac{\pi a_j}{m_j}\right)} = \frac{\mathbb{1}\{a_j \text{ is odd}\}}{\sin\left(\frac{\pi a_j}{2m_j}\right)^2}.$$

Let $\mathrm{odd}(a) := \{j \in N : a_j \text{ is odd}\}$, and $\mathrm{null}(a) := \{j \in N : a_j = 0\} = N \setminus \mathrm{supp}(a)$. If $\prod_{j \in S} |\widehat{\phi_j}(a_j)|^2 = \prod_{j \in S \cap N} |\widehat{\phi_j}(a_j)|^2 \neq 0$, then it must be the case that, for any $j \in S \cap N$, a_j is either odd or 0, i.e., $S \cap N \subseteq \mathrm{odd}(a) \cup \mathrm{null}(a)$. Notice that this contradicts $S \supseteq \mathrm{supp}(a)$ unless $\mathrm{supp}(a) \cap N = \mathrm{odd}(a)$, since, otherwise, there would be some $j \in N$ such that a_j is nonzero and even, and $S \supseteq \mathrm{supp}(a)$ would imply that this j is also in S. Therefore, for any S and a such that $\mathrm{supp}(a) \cap N = \mathrm{odd}(a)$, and $S \supseteq \mathrm{supp}(a)$ (which implies $S \supseteq \mathrm{odd}(a)$), we have

$$\prod_{j \in S} |\widehat{\phi_j}(a_j)|^2 = \prod_{j \in S \cap \mathrm{null}(a)} m_j^2 \prod_{j \in \mathrm{odd}(a)} \frac{1}{\sin\left(\frac{\pi a_j}{2m_j}\right)^2} = |\mathcal{U}_{S \cap \mathrm{null}(a)}|^2 \prod_{j \in \mathrm{odd}(a)} \frac{1}{\sin\left(\frac{\pi a_j}{2m_j}\right)^2}.$$

If $\mathrm{supp}(a) \cap N \neq \mathrm{odd}(a)$, then $\prod_{j \in S} |\widehat{\phi_j}(a_j)|^2 = 0$ for all $S \supseteq \mathrm{supp}(a)$.

We can now rewrite the error as

$$\begin{aligned} \mathrm{err}_p(q, \tilde{q}) &= \frac{1}{\mu} \sum_{\substack{a \in \mathcal{U}' \\ \mathrm{supp}(a) \in \mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) \prod_{j \in S} |\widehat{\phi_j}(a_j)|^2}{|T_S|^2}} \\ &= \frac{1}{\mu} \sum_{\substack{a \in \mathcal{U}' \\ \mathrm{supp}(a) \cap N = \mathrm{odd}(a)}} \prod_{j \in \mathrm{odd}(a)} \frac{1}{\sin\left(\frac{\pi a_j}{2m_j}\right)} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a)}} \frac{p(S) |\mathcal{U}_{S \cap \mathrm{null}(a)}|^2}{|T_S|^2}}. \end{aligned}$$

Here we used the observation that the sum equals 0 for any a where $\text{supp}(a) \cap N \neq \text{odd}(a)$ as discussed above. Using the observation that $|T_S| = |\mathcal{U}_S| \cdot 2^{|S \cap N|}$, we can rewrite further as

$$\begin{aligned}\text{err}_p(q, \tilde{q}) &= \frac{1}{\mu} \sum_{\substack{a \in \mathcal{U}' \\ \text{supp}(a) \cap N = \text{odd}(a)}} \prod_{j \in \text{odd}(a)} \frac{1}{\sin\left(\frac{\pi a_j}{2m_j}\right)} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \text{supp}(a)}} \frac{p(S)}{|\mathcal{U}_{S \setminus \text{null}(a)}|^2 \cdot 4^{|S \cap N|}}} \\ &= \frac{1}{\mu} \sum_{\substack{a \in \mathcal{U}' \\ \text{supp}(a) \cap N = \text{odd}(a)}} \prod_{j \in \text{odd}(a)} \frac{1}{m_j \sin\left(\frac{\pi a_j}{2m_j}\right)} \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq \text{supp}(a)}} \frac{p(S)}{|\mathcal{U}_{S \cap C}|^2 \cdot 4^{|S \cap N|}}}, \quad (15)\end{aligned}$$

where, in the second line, we used the observation that if $\text{supp}(a) \cap N = \text{odd}(a)$ and $S \supseteq \text{supp}(a)$, then $S \setminus \text{null}(a) = (S \cap C) \cup \text{odd}(a)$, so $|\mathcal{U}_{S \setminus \text{null}(a)}| = |\mathcal{U}_{S \cap C}||\mathcal{U}_{\text{odd}(a)}| = |\mathcal{U}_{S \cap C}| \prod_{j \in \text{odd}(a)} m_j$. Note that in these formulas we use the convention $|\mathcal{U}_\emptyset| = 1$.

We can re-write the right hand side further by first choosing the sets $R := \text{supp}(a) \cap C$, and $O := \text{odd}(a)$. For any fixed such choice of $R \subseteq C$ and $O \subseteq N$, we have

$$\begin{aligned}&\sum_{\substack{a \in \mathcal{U}' \\ \text{supp}(a) \cap C = R \\ \text{supp}(a) \cap N = \text{odd}(a) = O}} \prod_{j \in O} \frac{1}{m_j \sin\left(\frac{\pi a_j}{2m_j}\right)} \\ &\qquad = \left(\prod_{j \in R} (m_j - 1)\right) \prod_{j \in O} \left(\frac{1}{m_j} \sum_{\ell=1}^{m_j} \frac{1}{\sin\left(\frac{\pi(2\ell-1)}{2m_j}\right)}\right). \quad (16)\end{aligned}$$

Recall that in the theorem statement we defined $\eta(m) := \frac{1}{m} \sum_{\ell=1}^{m} \frac{1}{\sin\left(\frac{\pi(2\ell-1)}{2m}\right)}$. From (15) and (16), we have

$$\text{err}_p(q, \tilde{q}) = \frac{1}{\mu} \sum_{\substack{R \subseteq C \\ O \subseteq N}} \left(\prod_{j \in R} (m_j - 1)\right) \cdot \left(\prod_{j \in O} \eta(m_j)\right) \sqrt{\sum_{\substack{S \in \mathcal{S} \\ S \supseteq R \cup O}} \frac{p(S)}{|\mathcal{U}_{S \cap C}|^2 \cdot 4^{|S \cap N|}}}. \qquad \square$$

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 $\mathcal{U} = \mathcal{U}_1 = \{0, \ldots, m-1\}$. Then Theorem 4.8 gives us (for $p(S) = 1$)

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

which recovers the explicit upper bound on the error of a factorization algorithm for this workload due to Henzinger and Upadhyay [HU25]. 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 [HKU25] for a single prefix query. We note that the only known explicit factorization that outperforms ours on prefix queries is concurrent work [HKU25], and our mechanism handles extended marginals which is a much more general class of queries.

5 Lower Bounds

In this section we present lower bounds for factorization mechanisms and marginal queries that show that the algorithms from Sections 3 and 4.1 are optimal, and that the algorithm for extended marginals in Section 4.2 is optimal up to lower order terms. First, we present properties of factorization mechanisms. Then, we show how our algorithms can be represented in the factorization framework. Then, we present lower bounds for marginal queries that match our upper bounds. We also show how to extend these results to product queries, and, finally, we give a lower bound for extended marginals that nearly matches the factorization implied by the algorithms in Section 4.2.

5.1 Properties of Factorization Norms

We start with some general properties of factorization norms. For a matrix M, let $\|M\|_{1\to 2}$ 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., $\|M\|_F = \sqrt{\mathrm{tr}(M^T M)}$ for real matrices, and $\sqrt{\mathrm{tr}(M^* M)}$ for complex matrices. Finally, let $\|M\|_{tr}$ be the trace norm of M, i.e., the sum of its singular values, and let $\|M\|_{op}$ be the operator norm of M, i.e., its largest singular value.

For a $K \times N$ real matrix W, we have the factorization norms

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

Here the infima are over real matrices L and R for which $LR = W$. The γ_2 norm is classical, while the definition of the γ_F norm is due to Edmonds, Nikolov, and Ullman [ENU20], except the definition in the latter paper is normalized differently.

We deviate slightly from these definitions and allow the factors L and R to be complex matrices. This does not change the quantities we study, as explained in the following remark.

Remark 5.1. It is not hard to see that, when the matrix W has real entries, allowing the matrices L and R to have complex entries does not change $\gamma_F(W)$ or $\gamma_2(W)$. It is obvious that allowing complex entries can only decrease the norms. In the other direction, take any complex factorization $LR = W$ where $L = L_{\mathbb{R}} + iL_{\mathbb{C}}$ and $R = R_{\mathbb{R}} + iR_{\mathbb{C}}$ for real matrices $L_{\mathbb{R}}, L_{\mathbb{C}}, R_{\mathbb{R}}, R_{\mathbb{C}}$. Since $\mathrm{Re}(LR) = W$, we have the real factorization

$$\begin{pmatrix} L_{\mathbb{R}} & L_{\mathbb{C}} \end{pmatrix} \begin{pmatrix} R_{\mathbb{R}} \\ -R_{\mathbb{C}} \end{pmatrix} = W$$

achieving the same value as the complex factorization. This observation was also made by Henzinger and Upadhyay [HU25].

We also note that, for any diagonal matrix P with non-negative entries such that $\mathrm{tr}(P) = 1$, using the fact that

$$\|P^{1/2}L\|_F^2 = \sum_{i=1}^{K} P_{i,i}\|\ell_i\|_2^2 \le \max_{i=1}^{K} \|\ell_i\|_2^2 = \|L\|_{2\to\infty},$$

we have that $\gamma_F(P^{1/2}W) \le \gamma_2(W)$. Above, we used ℓ_i for the i-th row of L.

The next lemma gives a lower bound on γ_F, as well as a necessary and sufficient condition for a factorization to achieve the lower bound.

Lemma 5.2. *For any $K \times N$ matrix W, and any $N \times N$ diagonal matrix S with non-negative entries such that $\mathrm{tr}(S) = 1$, we have*

$$\gamma_F(W) \geq \|WS^{1/2}\|_{tr}.$$

Moreover, for any factorization $W = LR$, we have $\|WS^{1/2}\|_{tr} = \gamma_F(W) = \|L\|_F\|R\|_{1\to 2}$ if and only if

- *for any j such that $S_{j,j} \neq 0$, the ℓ_2 norm of the j-th column of R equals $\|R\|_{1\to 2}$, and*
- *$L^*L = c\,RSR^*$ for some real number $c \geq 0$.*

The next lemma is the analogous result for γ_2.

Lemma 5.3. *For any $K \times N$ matrix W, and any diagonal matrices P and S with non-negative entries, respectively of dimensions $K \times K$ and $N \times N$, and such that $\mathrm{tr}(P) = \mathrm{tr}(S) = 1$, we have*

$$\gamma_2(W) \geq \|P^{1/2}WS^{1/2}\|_{tr}.$$

Moreover, for any factorization $W = LR$, we have $\|P^{1/2}WS^{1/2}\|_{tr} = \gamma_2(W) = \|L\|_{2\to\infty}\|R\|_{1\to 2}$ if and only if

- *for any i such that $P_{i,i} \neq 0$, the ℓ_2 norm of the i-th row of L equals $\|L\|_{2\to\infty}$, and*
- *for any j such that $S_{j,j} \neq 0$, the ℓ_2 norm of the j-th column of R equals $\|R\|_{1\to 2}$, and*
- *$L^*PL = c\,RSR^*$ for some real number $c \geq 0$.*

The lower bounds in Lemmas 5.2 and 5.3 themselves are not new. A special case of the lower bound on γ_F, known as the singular value lower bound, was shown by Li and Miklau [LM13, LM15], and the lower bound for γ_2 goes back to work of Haagerup in the 1980s [Haa85], see also [LSŠ08]. Both lower bounds also follow from the duality results in [NT24]. Moreover, these duality results show that, for any W, both lower bounds are achieved for some choice of diagonal matrices. We have, however, not found the characterization of the tight cases in Lemmas 5.2 and 5.3 in the literature, which is why we include proofs of the lemmas in Appendix A.

5.2 Our Algorithms as Factorizations

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 $D = (x^{(1)}, \ldots, x^{(n)})$ 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 workload matrix, we define a corresponding private factorization mechanism for answering queries in Q as follows. We draw a normally distributed vector $Z \sim \mathcal{N}\left(0, \frac{\|R\|_{1\to 2}^2}{\mu^2} I_r\right)$, 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 r queries, and they have sensitivity $\max_{x\in\mathcal{U}}\|r_x\|_2 = \|R\|_{1\to 2}$. 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. It is easy to verify that

the mechanism satisfies μ-GDP: outputting $Rh + Z$ satisfies μ-GDP by Lemma 2.3, and then multiplying by L is just post-processing. Moreover, an easy calculation shows that the factorization mechanism achieves error bounds

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

$$\max_{q\in Q}\mathbb{E}[\|Wh - (L(Rh + Z))_q\|_2^2]^{1/2} = \max_{q\in Q}\mathbb{E}[|(LZ)_q|^2]^{1/2} = \frac{1}{\mu}\|L\|_{2\to\infty}\|R\|_{1\to 2}.$$

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. 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

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

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 algorithms fall in this framework: the Fourier queries defined in (1) are the strategy queries, and the formula (4) shows how to reconstruct the query answers. We now describe this re-interpretation of the algorithm more precisely. We only do so for Algorithm 2, since our other algorithms are special cases of it.

Let us fix a set of marginals $\mathcal{S}$ and a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, as in Section 3.3. Let W, from now on, be the workload matrix W of $Q_{\mathcal{S}}$. Recall the definition of τ_a for $a \in \mathcal{U}$ from Algorithm 2:

$$\tau_a := \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2}}\,. \tag{17}$$

Let $\tilde{V}$ be the complex matrix with columns indexed by those $a \in \mathcal{U}$ for which $\tau_a > 0$, and rows indexed by $\mathcal{U}$, and with entries $\tilde{V}_{x,a} = \chi_a(x)$. Next, let $\tilde{U}$ be the matrix for which $\tilde{U}\tilde{V}^* = W$. The rows of $\tilde{U}$ are indexed by (S, t) where $S \in \mathcal{S}$, and $t \in \mathcal{U}_S$. The columns are indexed by the same set as the columns of $\tilde{v}$, i.e., by those $a \in \mathcal{U}$ for which $\tau_a > 0$. The entries of $\tilde{U}$ can be deduced from (3), and are

$$\tilde{U}_{(S,t),a} = \begin{cases} \frac{1}{|\mathcal{U}_S|}\chi_a(t) & \mathrm{supp}(a) \subseteq S \\ 0 & \mathrm{supp}(a) \not\subseteq S \end{cases}.$$

Now $\tilde{U}\tilde{V}^* = W$ follows from (3). This factorization, however, does not, in general, correspond to Algorithm 2 because the noise variance we use for different Fourier queries is different. Instead, let L be a matrix of the same dimensions as $\tilde{U}$, and let R be a matrix of the same dimensions as $\tilde{V}^*$, and define their entries by

$$L_{(S,t),a} := \tilde{U}_{(S,t),a}\sqrt{\frac{\sum_{b\in\mathcal{U}}\tau_b}{\tau_a}} \qquad R_{a,x} := \sqrt{\frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}}\overline{\tilde{V}_{x,a}} = \sqrt{\frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}}(\tilde{V}^*)_{a,x}\,. \tag{18}$$

In other words, if we define a diagonal matrix E with rows and columns indexed by $a \in \mathcal{U}$ such that $\tau_a > 0$, and let $E_{a,a} := \frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}$, then $L := \tilde{U}E^{-1/2}$ and $R := E^{1/2}\tilde{V}^*$. Clearly $LR = \tilde{U}\tilde{V}^* = W$.

To see the equivalence with Algorithm 2, consider the mechanism that outputs $L(Rh + \tilde{Z})$ for $\tilde{Z} \sim \mathcal{CN}\big(0, \frac{2}{\mu^2} I\big)$. We can think of this mechanism as first releasing $E^{1/2}(Rh + \tilde{Z}) = \tilde{V}^* h + E^{1/2}\tilde{Z}$, and then multiplying on the left by $\tilde{U}$. Now it is immediate that that $\tilde{V}^* h$ gives the answers to the Fourier queries $F_a(D)$, and that $(E^{1/2}\tilde{Z})_a$ is distributed as Z_a in Algorithm 2. Therefore $E^{1/2}(Rh + \tilde{Z})$ is distributed identically to the estimates $\tilde{F}_a$ in Algorithm 2. Finally, multiplying this vector on the left by $\tilde{U}$ and taking real parts gives the estimates $\tilde{q}_{S,t}(D)$. Thus, Algorithm 2 corresponds to running a complex version of the factorization mechanism and taking real parts of the result. It is easy to see that this is also equivalent to making the factorization real (as described in Remark 5.1), and running the usual real-valued factorization mechanism.

The following lemma records, for future reference, the fact that the error bounds from Lemmas 3.7 and 3.9 are achieved by factorization mechanisms.

Lemma 5.4. *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 $LR = W$ in* (18) *satisfies*

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

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

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

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

Proof. The lemma follows from the calculations in Lemmas 3.7 and 3.9. First, observe that, since every entry of $\tilde{V}$ is a root of unity, so has absolute value 1, the squared ℓ_2 norm of every column of R is $\left(\sum_{a \in \mathcal{U}} \tau_a\right) / \left(\sum_{b \in \mathcal{U}} \tau_b\right) = 1$. Therefore, $\|R\|_{1\to 2} = 1$. To compute $\|P^{1/2}L\|_F$, observe that the squared ℓ_2 norm of the row of L indexed by (S, t) is

$$\frac{\sum_{b \in \mathcal{U}} \tau_b}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \frac{1}{\tau_a} = \frac{\sum_{b \in \mathcal{U}} \tau_b}{|\mathcal{U}_S|^2} \sum_{\substack{a \in \mathcal{U} \\ \mathrm{supp}(a) \subseteq S}} \left(\sum_{\substack{T \in \mathcal{S} \\ T \supseteq \mathrm{supp}(a)}} \frac{p(T)}{|\mathcal{U}_T|^2} \right)^{-1/2}.$$

This is exactly $\mu^2 \sigma_S^2$, for σ_S^2 as defined in (5). We now have $\|P^{1/2}L\|_F^2 = \mu^2 \sum_{S \in \mathcal{S}} p(S) \sigma_S^2$, and the result follows as in the proof of Lemma 3.7.

These observations also hold for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (6): the squared ℓ_2 norm of the row of L indexed by (S, t) is $\mu^2 \sigma_S^2$. Moreover, by Lemma 3.10 we know that for any $S \in \mathcal{S}$ such that $p^*(S) > 0$, we have $\sigma_S = \max_{T \in \mathcal{S}} \sigma_T$. Therefore, $\|L\|_{2\to\infty} = \|(P^*)^{1/2}L\|_F$, where P^* is defined as P above but with p^* in place of p. The result follows from the equation for $\|(P^*)^{1/2}L\|_F$ above. □

Remark 5.5. When we transform the matrices L and R in our factorization into real matrices as in Remark 5.1, we create some redundant rows of R and columns of L. Indeed, for any

$a \in \mathcal{U}$, let a' be defined by $a_j = m_j - a_j \bmod m_j$ for all $j \in [d]$. Then, for any $x \in \mathcal{U}$, $\chi_a(x) = \overline{\chi_{a'}(x)}$. Writing $R = R_{\mathbb{R}} + i\mathbb{R}_{\mathbb{C}}$ as in Remark 5.1, we see that the rows of $R_{\mathbb{R}}$ corresponding to a and a' are the same, and the rows of $R_{\mathbb{C}}$ corresponding to a and a' are either both all-zero, or negations of each other. Similar observations can be made about $L_{\mathbb{R}}$ and $L_{\mathbb{C}}$. It is possible to remove these redundancies without changing the quality of the factorization, resulting in smaller matrices. We leave the details to the reader.

5.3 Lower Bounds for Marginals

Now we are ready prove that the upper bounds on $\gamma_F(P^{1/2}W)$ and $\gamma_2(W)$ in Lemma 5.4 in fact hold with equality, and, therefore, the factorization $LR = W$ in (18) is optimal. We use the same notation as the previous subsection. Our proof relies on Lemmas 5.2 and 5.3.

We will make extensive use of the following basic identity, which states the classical and easy to check fact that the Fourier basis is orthonormal (with the correct normalization). We have that, for any $a, b \in \mathcal{U}$,

$$\frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \chi_a(x)\overline{\chi_b(x)} = \mathbb{1}\{a = b\}, \tag{19}$$

where $\mathbb{1}\{a = b\}$ is 1 if $a = b$ and 0 otherwise.

Lemma 5.6. *For any workload of marginal queries $Q_{\mathcal{S}}$, and for any weights $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, with W and P as in Lemma 5.4, the factorization $LR = W$ for L and R defined in* (18) *satisfies $\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F \|R\|_{1\to 2}$.*

Similarly, for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (6)*, the factorization satisfies $\gamma_2(W) = \|L\|_{2\to\infty}\|R\|_{1\to 2}$.*

Proof. We will use Lemma 5.2 with $S := \frac{1}{|\mathcal{U}|} I$. In the proof of Lemma 5.4 we observed that all columns of R have the same ℓ_2 norm. It then suffices to check that $L^*PL = \frac{c}{|\mathcal{U}|} RR^*$ for some real constant $c \geq 0$.

First we compute RR^*. Recall that E is a diagonal matrix with diagonal entries $E_{a,a} := \frac{\tau_a}{\sum_{b \in \mathcal{U}} \tau_b}$ for all $a \in \mathcal{U}$ such that $\tau_a > 0$. For any a and b such that $\tau_a > 0$ and $\tau_b > 0$, we have

$$\begin{aligned}(RR^*)_{a,b} = (E^{1/2}\tilde{V}^*\tilde{V}E^{1/2})_{a,b} &= \sqrt{E_{a,a}E_{b,b}}(\tilde{V}^*\tilde{V})_{a,b} \\ &= \sqrt{E_{a,a}E_{b,b}} \sum_{x \in \mathcal{U}} \chi_b(x)\overline{\chi_a(x)} = |\mathcal{U}|\sqrt{E_{a,a}E_{b,b}}\ \mathbb{1}\{a = b\}.\end{aligned}$$

The last equality follows from (19). Therefore, $RR^* = |\mathcal{U}|E$.

Now we turn to computing L^*PL. For any a and b such that $\tau_a > 0$ and $\tau_b > 0$, we have

$$\begin{aligned}(L^*PL)_{a,b} = (E^{-1/2}\tilde{U}^*P\tilde{U}E^{-1/2})_{a,b} &= \frac{1}{\sqrt{E_{a,a}E_{b,b}}}(\tilde{U}^*P\tilde{U})_{a,b} \\ &= \frac{1}{\sqrt{E_{a,a}E_{b,b}}} \sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a) \cup \mathrm{supp}(b)}} \frac{p(S)}{|\mathcal{U}_S|^3} \sum_{t \in \mathcal{U}_S} \chi_b(t)\overline{\chi_a(t)} \\ &= \left(\frac{1}{\sqrt{E_{a,a}E_{b,b}}} \sum_{\substack{S \in \mathcal{S} \\ S \supseteq \mathrm{supp}(a) \cup \mathrm{supp}(b)}} \frac{p(S)}{|\mathcal{U}_S|^2} \right) \cdot \mathbb{1}\{a = b\}.\end{aligned}$$

The last equality again follows from (19) after observing that, since any S in the outer sum contains both $\mathrm{supp}(a)$ and $\mathrm{supp}(b)$, χ_a and χ_b can be interpreted as Fourier characters over $\mathcal{U}_S$. We then have that L^*PL is a diagonal matrix, and the diagonal entries are

$$(L^*PL)_{a,a} = \frac{1}{E_{a,a}} \sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2} = \frac{\sum_{b\in\mathcal{U}}\tau_b}{\tau_a} \sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2}$$
$$= \left(\sum_{b\in\mathcal{U}}\tau_b\right)\tau_a = \left(\sum_{b\in\mathcal{U}}\tau_b\right)^2 E_{a,a}.$$

In the penultimate equality we used the definition of τ_a in (17). In conclusion,

$$L^*PL = \left(\sum_{b\in\mathcal{U}}\tau_b\right)^2 E = \frac{\left(\sum_{b\in\mathcal{U}}\tau_b\right)^2}{|\mathcal{U}|} RR^*,$$

as we needed to show.

Let P^* be the diagonal matrix P but with p^* in place of p. By Lemma 5.3, and the proof so far, it is enough to check that for any $S \in \mathcal{S}$ such that $p^*(S) > 0$, and any $t \in \mathcal{U}_S$, the ℓ_2 norm of the row of L indexed by (S,t) equals $\|L\|_{2\to\infty}$, i.e., is maximal. The square of this ℓ_2 norm equals

$$\frac{\sum_{b\in\mathcal{U}}\tau_b}{|\mathcal{U}_S|^2} \sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}} \frac{1}{\tau_a} = \frac{\sum_{b\in\mathcal{U}}\tau_b}{|\mathcal{U}_S|^2} \sum_{\substack{a\in\mathcal{U}\\ \mathrm{supp}(a)\subseteq S}} \left(\sum_{\substack{T\in\mathcal{S}\\ T\supseteq\mathrm{supp}(a)}} \frac{p^*(T)}{|\mathcal{U}_T|^2}\right)^{-1/2}.$$

The maximality of the row norm is now implied by Lemma 3.10. □

Remark 5.7. The proof of Lemma 5.6 gives a singular value decomposition of the matrix $P^{1/2}W$. The matrix of right singular vectors is $V := \frac{1}{\sqrt{|\mathcal{U}|}}\tilde{V}$. Let Σ be a diagonal matrix indexed by those a for which $\tau_a > 0$, and with entries $\Sigma_{a,a} := \tau_a$. Then the matrix of left singular vectors is $U := \tilde{U}\Sigma^{-1}$, and the diagonal entries of Σ are the singular values. The singular value decomposition is then $P^{1/2}W = U\Sigma V^*$. Our factorization results from splitting the singular value decomposition into a left matrix $U\Sigma^{1/2}$ and a right matrix $\Sigma^{1/2}V^*$, and normalizing the right matrix to have unit norm columns.

The next theorem is our main result giving optimal factorization norms for the workload matrix of a workload of weighted marginal queries. The theorem is a direct consequence of Lemmas 5.4 and 5.6.

Theorem 5.8. *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 $LR = W$ in* (18) *satisfies*

$$\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F\|R\|_{1\to 2} = \sum_{R\in\mathcal{S}_\downarrow}\left(\prod_{j\in R}(m_j-1)\right)\sqrt{\sum_{\substack{T\in\mathcal{S}\\ T\supseteq R}}\frac{p(T)}{|\mathcal{U}_T|^2}}. \tag{20}$$

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

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

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

Remark 5.9. Since in the proof of Lemma 5.6 we showed that L^*PL is a constant multiple of RR^*, Lemma 5.2 implies that the right hand side of (20) equals the singular value lower bound of $P^{1/2}W$, i.e., the value of $\frac{1}{\sqrt{|\mathcal{U}|}}\|P^{1/2}W\|_{tr}$. This formula is similar but different from the one given in by McKenna et al. [MMHM23]. We explain the discrepancy in Appendix A.

Finally, we record as a corollary the optimal factorization norms for the workload of all k-way marginal queries when $m_j = m$ for all $j \in [d]$. This is the special cases considered in Algorithm 1, and the corollary shows the algorithm is optimal among factorization mechanisms for this workload.

Corollary 5.10. *Let $\mathcal{S}$ be the set of all subsets of $[d]$ of size k, and let $\mathcal{U}$ be a universe such that $|\mathcal{U}_j| = m_j = m$ for all $j \in [d]$. Let W be the workload matrix of $Q_\mathcal{S}$. The factorization $LR = W$ in* (18) *with $p(S) = 1/\binom{d}{k}$ for all $S \in \mathcal{S}$ satisfies*

$$\frac{1}{\sqrt{m^k\binom{d}{k}}}\gamma_F(W) = \gamma_2(W) = \|L\|_{2\to\infty}\|R\|_{1\to 2} = \frac{1}{m^k\sqrt{\binom{d}{k}}}\sum_{\ell=0}^{k}\binom{d}{\ell}(m-1)^\ell\sqrt{\binom{d-\ell}{k-\ell}}.$$

Proof. Follows from Theorem 5.8 and the observation that $p(S) = p^*(S) = \frac{1}{\binom{d}{k}}$ satisfies the conclusion in Lemma 3.10. □

Remark 5.11. Similarly to Remark 3.12, the results in this section can be extended to other factorization norms, corresponding to error measures that interpolate between root mean squared error, and maximum variance. These are the $\gamma_{(p)}$ norms defined in [NT24], and studied further in [LUZ24]. We believe our methods extend to these norms, but we do not pursue them further.

5.4 Lower Bounds for Product Queries

Here we show that Algorithm 3 describes an optimal factorization mechanism for workloads of product queries.

First we describe how to formulate Algorithm 3 as a factorization, as we did with Algorithm 2. Let us fix the functions $\phi_1, \ldots, \phi_d$, a collection $\mathcal{S}$ of subsets of $[d]$, and a weight function $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$. Let W be, for the rest of this subsection, the workload matrix of $Q^\phi_\mathcal{S}$. Recall the definition of τ_a for $a \in \mathcal{U}$ from Algorithm 3:

$$\tau_a := \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}}. \tag{21}$$

Let $\tilde{V}$ be, as before, the complex matrix with columns indexed by those $a \in \mathcal{U}$ for which $\tau_a > 0$, and rows indexed by $\mathcal{U}$, and with entries $\tilde{V}_{x,a} = \chi_a(x)$. The matrix $\tilde{U}$ is such that $\tilde{U}\tilde{V}^* = W$, and can be described explicitly using the formula (11) used in Algorithm 3. The

rows of $\tilde{U}$ are indexed by (S,t) where $S \in \mathcal{S}$, and $t \in \mathcal{U}_S$, and the columns are indexed by the same set as the columns of $\tilde{V}$. The entries of $\tilde{U}$ are

$$\tilde{U}_{(S,t),a} = \begin{cases} \frac{1}{|\mathcal{U}_S|}\left(\prod_{j\in S}\widehat{\phi_j}(a_j)\right)\chi_a(t) & \text{supp}(a) \subseteq S \\ 0 & \text{supp}(a) \not\subseteq S \end{cases}.$$

To define the factorization $LR = W$, we take L and R to have the same dimensions as, respectively $\tilde{U}$ and $\tilde{V}^*$, and define their entries by

$$L_{(S,t),a} := \tilde{U}_{(S,t),a}\sqrt{\frac{\sum_{b\in\mathcal{U}}\tau_b}{\tau_a}} \qquad R_{a,x} := \sqrt{\frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}}\overline{\tilde{V}_{x,a}} = \sqrt{\frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}}(\tilde{V}^*)_{a,x}\,. \tag{22}$$

Once again, we can equivalently write this by defining a diagonal matrix E with rows and columns indexed by $a \in \mathcal{U}$ such that $\tau_a > 0$, and letting $E_{a,a} := \frac{\tau_a}{\sum_{b\in\mathcal{U}}\tau_b}$; then $L := \tilde{U}E^{-1/2}$, and $R := E^{1/2}\tilde{V}^*$. The equivalence with Algorithm 3 is easy to check, as we did with Algorithm 2. This construction gives the following upper bounds on factorization norms of the workload matrix W, which simply record the error guarantees of Algorithm 3 in the language of factorization norms. The proof is analogous to the proof of Lemma 5.4.

Lemma 5.12. *Let $Q^{\phi}_{\mathcal{S}}$ be a workload of product queries over a universe $\mathcal{U}$, with functions $\phi := (\phi_1, \ldots, \phi_d)$ associated with the attributes, with workload matrix W. Let $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$ be a weight function, with 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 $LR = W$ in (22) satisfies*

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

Moreover, for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (12), the factorization satisfies

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

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

As with Lemma 5.4, the upper bound in Lemma 5.12 holds with equality, as shown in the following lemma, which is analogous to Lemma 5.6.

Lemma 5.13. *For any workload of product queries $Q^{\phi}_{\mathcal{S}}$, and for any weights $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, with W and P as in Lemma 5.12, the factorization $LR = W$ for L and R defined in (22) satisfies $\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F\|R\|_{1\to 2}$.*

Similarly, for $p^* : \mathcal{S} \to \mathbb{R}_{\geq 0}$ chosen as in (12), the factorization satisfies $\gamma_2(W) = \|L\|_{2\to\infty}\|R\|_{1\to 2}$.

Proof. The proof is almost identical to the proof of Lemma 5.6. We again use Lemma 5.2 with $S := \frac{1}{|\mathcal{U}|}I$, and it suffices to check that $L^*PL = \frac{c}{|\mathcal{U}|}RR^*$ for some real constant $c \geq 0$. Once again $RR^* = |\mathcal{U}|E$ by the same calculation (despite the definition of τ_a and therefore E being different).

Now we turn to computing L^*PL. For any a and b such that $\tau_a > 0$ and $\tau_b > 0$, we have

$$\begin{aligned}(L^*PL)_{a,b} &= \frac{1}{\sqrt{E_{a,a}E_{b,b}}}(\tilde{U}^*P\tilde{U})_{a,b} \\ &= \frac{1}{\sqrt{E_{a,a}E_{b,b}}} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)\cup\mathrm{supp}(b)}} \frac{p(S)\prod_{j\in S}(\widehat{\phi_j}(b_j)\overline{\widehat{\phi_j}(a_j)})}{|\mathcal{U}_S|^3} \sum_{t\in\mathcal{U}_S} \chi_b(t)\overline{\chi_a(t)} \\ &= \left(\frac{1}{\sqrt{E_{a,a}E_{b,b}}} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)\cup\mathrm{supp}(b)}} \frac{p(S)\prod_{j\in S}(\widehat{\phi_j}(b_j)\overline{\widehat{\phi_j}(a_j)})}{|\mathcal{U}_S|^2}\right) \cdot \mathbb{1}\{a=b\}.\end{aligned}$$

The last equality again follows from (19) after identifying χ_a and χ_b with Fourier characters over $\mathcal{U}_S$. We then have that L^*PL is a diagonal matrix, and the diagonal entries are

$$\begin{aligned}(L^*PL)_{a,a} &= \frac{1}{E_{a,a}} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2} \\ &= \frac{\sum_{b\in\mathcal{U}}\tau_b}{\tau_a} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2} \\ &= \left(\sum_{b\in\mathcal{U}}\tau_b\right)\tau_a = \left(\sum_{b\in\mathcal{U}}\tau_b\right)^2 E_{a,a}.\end{aligned}$$

Thus,

$$L^*PL = \left(\sum_{b\in\mathcal{U}}\tau_b\right)^2 E = \frac{\left(\sum_{b\in\mathcal{U}}\tau_b\right)^2}{|\mathcal{U}|} RR^*,$$

as we needed to show.

Let P^* be the diagonal matrix P but with p^* in place of p. By Lemma 5.3, and the proof so far, it is enough to check that for any $S \in \mathcal{S}$ such that $p^*(S) > 0$, and any $t \in \mathcal{U}_S$, the ℓ_2 norm of the row of L indexed by (S,t) equals $\|L\|_{2\to\infty}$, i.e., is maximal. This is verified analogously to Lemma 5.6, using Lemma 4.3. □

The next theorem is a direct consequence of Lemmas 5.12 and 5.13. It gives optimal factorization norms for a workload of weighted product queries, and shows optimality of Algorithm 3 among all factorization mechanisms.

THEOREM 5.14. *Let $Q^{\phi}_{\mathcal{S}}$ 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 $LR = W$ in* (18) *satisfies*

$$\gamma_F(P^{1/2}W) = \|P^{1/2}L\|_F\|R\|_{1\to 2} = \sum_{\substack{a\in\mathcal{U} \\ \mathrm{supp}(a)\in\mathcal{S}_\downarrow}} \sqrt{\sum_{\substack{S\in\mathcal{S} \\ S\supseteq\mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S}|\widehat{\phi_j}(a_j)|^2}{|\mathcal{U}_S|^2}}, \qquad (23)$$

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 (12)*, the factorization satisfies*

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

5.5 Lower Bounds for Extended Marginals

Our goal in this subsection is to show that the upper bound in Theorem 4.8 is *nearly* optimal among all factorization mechanisms. We do so by proving a lower bound on the trace norm $\|P^{1/2}W\|_{tr}$ of the weighted workload matrix $P^{1/2}W$ of a workload of extended marginals.

In the special case where $d = 1$, and the only attribute is numerical, the workload is just the workload of prefix queries. The corresponding workload matrix is lower triangular, and has 1's on and below the main diagonal. The singular value decomposition (SVD) of this matrix is well known, see [MNT20]. Unfortunately, we are not able to give an explicit description of the singular value decomposition of W when we have a mix of numerical and categorical attributes. A technical hurdle is that the all-ones vector is not a singular vector of the lower triangular matrix mentioned above.

To get around this issue, we instead prove a lower bound on $\|P^{1/2}W\|_{tr}$ using the fact that $\|W\|_{tr} \geq \frac{|\mathrm{tr}(P^{1/2}WY^*)|}{\|Y\|_{op}}$ for any matrix Y of the same dimensions as W (Lemma A.1 in Appendix A). To construct such a "test matrix" Y, we take inspiration from the proof of Theorem 5.14. Note that, if the SVD of $P^{1/2}W$ is $P^{1/2}W = U\Sigma V^*$, then UV^* is an optimal choice of Y, i.e., choosing $Y := UV^*$, we have $\mathrm{tr}(P^{1/2}WY^*) = \|P^{1/2}W\|_{tr}$ and $\|Y\|_{op} = 1$. While we do not know closed forms for the matrices U and V for extended marginals, we "pretend" W is defined by product queries, and construct U and V analogously to the construction of $\tilde{U}$ and $\tilde{V}$ in Section 5.4.

Before presenting our lower bound, let us illustrate this idea with the case of $d = 1$ and a single numerical attribute, i.e., a workload of prefix queries on a universe of size m. Then the workload matrix W is an $m \times m$ Toeplitz lower triangular matrix, with 1's on and below the main diagonal. If this matrix were circulant, then its left and right singular vectors would be proportional (up to a phase shift) to the Fourier basis vectors $v_a := \frac{1}{\sqrt{m}}(1, \omega_m^a, \omega_m^{2a}, \ldots, \omega_m^{(m-1)a})$, $a \in \{0, \ldots, m-1\}$. While W is not circulant, we can still use the Fourier basis vectors to construct a matrix Y such that $\mathrm{tr}(WY^*) \approx \|W\|_{tr}$. In particular, it is not hard to calculate that

$$v_a^* W v_a = \begin{cases} \frac{m+1}{2} & a = 0 \\ \frac{1}{1-\omega_m^{-a}} & a \neq 0 \end{cases}.$$

Using this observation, we can define an $m \times m$ matrix V which columns the Fourier basis vectors $v_0, \ldots v_{m-1}$, and an $m \times m$ matrix $U := VE$, where E is a diagonal matrix with entries

$$E_{a,a} := \begin{cases} 1 & a = 0 \\ \frac{1-\omega_m^a}{|1-\omega_m^{-a}|} & a \neq 0 \end{cases}.$$

This matrix is defined exactly so that $\overline{E_{a,a}} v_a^* W v_a = |v_a^* W v_a|$. Then we define the matrix $Y := UV^* = VEV^*$, which clearly satisfies that $\|Y\|_{op} = 1$. We have

$$\begin{aligned}\|W\|_{tr} \geq |\mathrm{tr}(WY^*)| &= |\mathrm{tr}(WVE^*V^*)| \\ &= |\mathrm{tr}(E^*V^*WV)| = \left|\sum_{a=0}^{m-1} \overline{E_{a,a}}\, v_a^* W v_a\right| = \frac{m+1}{2} + \sum_{a=1}^{m-1} \frac{1}{|1-\omega_m^{-a}|}.\end{aligned}$$

The right hand side is $\frac{m+1}{2} + \sum_{a=1}^{m-1} \frac{1}{\sin\left(\frac{\pi a}{m}\right)}$, which can be shown to be within an additive constant of $\|W\|_{tr}$ (see, again, [MNT20]).

We now extend this approach to arbitrary workloads of extended marginals. Let us first recall the notation for extended marginals from Section 4.2. The universe is $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_d$, and the d attributes are partitioned into C and N, where C are the categorical attributes, and N are the numerical ones. We consider the workload $Q_{\mathcal{S}}^{\text{pref}}$ of extended marginal queries defined by a family of subsets $\mathcal{S}$ of $[d] = C \cup N$. The corresponding workload matrix W has rows indexed by pairs (S,t) for $S \in \mathcal{S}$ and $t \in \mathcal{U}_S$, and columns indexed by $\mathcal{U}$. The entries of the workload matrix are $W_{(S,t),x} = q_{S,t}(x)$, with $q_{S,t}(x)$ as defined in (13). We also fix a weight functions $p : \mathcal{S} \to \mathbb{R}_{\geq 0}$, and define the corresponding diagonal matrix P indexed by pairs (S,t) for $S \in \mathcal{S}$, and $t \in \mathcal{U}_S$, with entries $P_{(S,t),(S,t)} := \frac{p(S)}{|\mathcal{U}_S|}$. We focus on lower bounds for factorization mechanisms for releasing $Q_{\mathcal{S}}^{\text{pref}}$, but an analogous argument would apply to $Q_{\mathcal{S}}^{\text{pr-suf}}$.

We define a test matrix Y such that $\|Y\|_{op} = 1$, and compute $\mathrm{tr}(P^{1/2}WY^*)$, which is a lower bound on $\|P^{1/2}W\|_{tr}$. First we define a few auxiliary matrices. Let $A \subseteq \mathcal{U}$ be the set of all $a \in \mathcal{U}$ such that $\mathrm{supp}(a) \in \mathcal{S}_{p,\downarrow}$, i.e., such that $\mathrm{supp}(a)$ is contained in some $S \in \mathcal{S}$ with $p(S) > 0$. As in our other lower bound arguments, let $\tilde{V}$ be a complex matrix with columns indexed by A and rows indexed by $\mathcal{U}$, and with entries $\tilde{V}_{x,a} := \chi_a(x)$. We also define another complex matrix $\tilde{U}$, with rows indexed by pairs (S,t) for $S \in \mathcal{S}$, and $t \in \mathcal{U}_S$, and columns indexed by A. Let us define the functions $f_j : \mathcal{U}_j \to \mathbb{C}$ for every $j \in N$ by

$$f_j(a) := \begin{cases} \frac{m_j+1}{2} & a = 0 \\ \frac{1}{1-\omega_{m_j}^{-a}} & a \neq 0 \end{cases}.$$

We define the entries of $\tilde{U}$ as

$$\tilde{U}_{(S,t),a} := \begin{cases} \frac{1}{|\mathcal{U}_S|} \left(\prod_{j \in S \cap N} f_j(a_j)\right) \chi_a(t) & \mathrm{supp}(a) \subseteq S \\ 0 & \text{otherwise} \end{cases}.$$

This choice of $\tilde{U}$ is analogous to the definition in Section 5.4. In particular, if we had a workload of product queries on $\mathcal{U}$ for which $\phi_j(z) := \mathbb{1}\{z = 0\}$ for $j \in C$, and for $j \in N$ the function ϕ_j were such that $\widehat{\phi_j}(a) = f_j(a)$, then this would be the same $\tilde{U}$. While this workload of product queries is not same as the workload of extended marginals, it turns out that the same definition of $\tilde{U}$ works for our purposes.

Next we normalize $\tilde{U}$ and $\tilde{V}$ so that they have orthonormal columns. We define $V := \frac{1}{\sqrt{|\mathcal{U}|}}\tilde{V}$. We also define

$$\kappa_a := \sqrt{\sum_{\substack{S\in\mathcal{S}\\ t\in\mathcal{U}_S}} \frac{p(S)}{|\mathcal{U}_S|}|\tilde{U}_{(S,t),a}|^2} = \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S\cap N}|f_j(a_j)|^2}{|\mathcal{U}_S|^2}}.$$

We then define $U_{(S,t),a} := \frac{\tilde{U}_{(S,t),a}}{\kappa_a}$. It is now easy to check that V and $P^{1/2}U$ have orthonormal columns. Our test matrix is $Y := P^{1/2}UV^*$. Our lower bound, derived from computing $\mathrm{tr}(P^{1/2}WY^*)$, is given by the following theorem.

Theorem 5.15. *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 $\zeta(m) := \frac{1}{m}\sum_{\ell=1}^{m-1}\frac{1}{\sin\left(\frac{\pi\ell}{m}\right)}$.

Before we prove the theorem, let us see how it implies that our factorizations for extended marginals are nearly optimal. Notice that, since $1+\frac{1}{m_j} > 1$ for every j, the lower bound in Theorem 5.15 is at least as strong as

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

Recall also that $\gamma_2(W) \geq \gamma_F(P^{1/2}W)$ for any weight function p and the corresponding diagonal matrix P.

On the other hand, we can use Lemma 5.12 and the calculations in the proof of Theorem 4.8 to get the upper bound

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

Moreover, there is a weight function p^* for which

$$\gamma_2(W) \leq \sum_{\substack{R\subseteq C\\ O\subseteq N}} \left(\prod_{j\in R}(m_j-1)\right)\cdot\left(\prod_{j\in O}\eta(m_j)\right)\sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq R\cup O}} \frac{p^*(S)}{|\mathcal{U}_{S\cap C}|^2\cdot 4^{|S\cap N|}}}.$$

The upper and lower bounds differ only in that $\eta(m) := \frac{1}{m}\sum_{\ell=1}^{m}\frac{1}{\sin\left(\frac{\pi(2\ell-1)}{2m}\right)}$ is replaced by $\zeta(m) := \frac{1}{m}\sum_{a=1}^{m-1}\frac{1}{\sin\left(\frac{\pi a}{m}\right)} = \frac{1}{m}\sum_{\ell=1}^{m-1}\frac{1}{\sin\left(\frac{\pi\cdot 2\ell}{2m}\right)}$. These two functions are always within an additive constant of each other, and are both asymptotically equal to $\frac{1}{\pi}\ln(m)+O(1)$. See

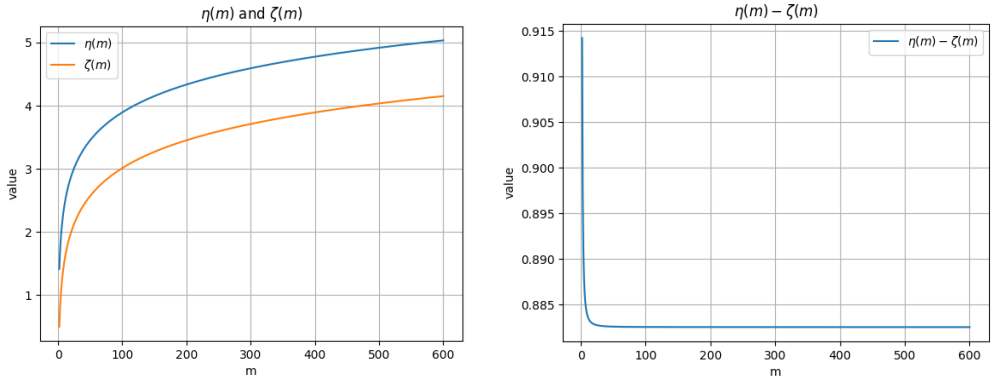

Fig. 2. The functions $\eta(m)$ and $\zeta(m)$, as well as their difference $\eta(m) - \zeta(m)$.

Figure 2. This means that the upper and lower bounds on $\gamma_F(P^{1/2}W)$ and $\gamma_2(W)$ are equal up to a multiplicative factor of $1 - o(1)$ as $\min_{j\in N} m_j \to \infty$ with $\max_{S\in\mathcal{S}}|S|$ fixed. This regime of large domain size for the numerical attributes is arguably the most natural one.

Proof of Theorem 5.15. We define $Y := P^{1/2}UV^*$ with U and V as defined above. First we claim that $\|P^{1/2}U\|_{op} = \|V\|_{op} = 1$, and, therefore, $\|Y\|_{op} \le 1$. We have

$$(V^*V)_{a,b} = \frac{1}{|\mathcal{U}|}(\tilde{V}^*\tilde{V})_{a,b} = \frac{1}{|\mathcal{U}|}\sum_{x\in\mathcal{U}} \chi_b(x)\overline{\chi_a(x)} = \mathbb{1}\{a = b\},$$

by (19). Therefore, V^*V is the identity, and all singular values of V are 1, which implies $\|V\|_{op} \le 1$. For $P^{1/2}U$, we have

$$\begin{aligned}(U^*PU)_{a,b} &= \frac{1}{\kappa_a\kappa_b}(\tilde{U}^*P\tilde{U})_{a,b} \\ &= \frac{1}{\kappa_a\kappa_b} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)\cup\mathrm{supp}(b)}} \frac{p(S)\prod_{j\in S\cap N}(f_j(b_j)\overline{f_j(a_j)})}{|\mathcal{U}_S|^3} \sum_{t\in\mathcal{U}_S} \chi_b(t)\overline{\chi_a(t)} \\ &= \left(\frac{1}{\kappa_a\kappa_b} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)\cup\mathrm{supp}(b)}} \frac{p(S)\prod_{j\in S\cap N}(f_j(b_j)\overline{f_j(a_j)})}{|\mathcal{U}_S|^2}\right) \cdot \mathbb{1}\{a = b\}.\end{aligned}$$

Therefore, U^*PU is a diagonal matrix, with diagonal entries

$$(U^*PU)_{a,a} = \frac{1}{\kappa_a^2} \sum_{\substack{S\in\mathcal{S} \\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)\prod_{j\in S\cap N}|f_j(a_j)|^2}{|\mathcal{U}_S|^2} = 1.$$

This shows that $U^*PU = I$, and $\|P^{1/2}U\|_{op} = 1$. Therefore, $\|Y\|_{op} \le \|P^{1/2}U\|_{op}\|V^*\|_{op} = 1$, and, by Lemma A.1, $|\mathrm{tr}(P^{1/2}WY^*)| \le \|P^{1/2}W\|_{tr}$. We proceed to compute $\mathrm{tr}(P^{1/2}WY^*)$. We will need fact show that $\mathrm{tr}(P^{1/2}WY^*)$ is real and non-negative, so the absolute value will be unnecessary.

Note that

$$\mathrm{tr}(P^{1/2}WY^*) = \mathrm{tr}(P^{1/2}WVU^*P^{1/2}) = \mathrm{tr}(U^*PWV).$$

Let us first compute WV. For each $S \in \mathcal{S}$, $t \in \mathcal{U}_S$, and each $a \in A$, we have

$$\begin{aligned}(WV)_{(S,t),a} &= \frac{1}{\sqrt{|\mathcal{U}|}} \sum_{x \in \mathcal{U}} q_{S,t}(x)\chi_a(x) \\ &= \frac{1}{\sqrt{|\mathcal{U}|}} \sum_{x \in \mathcal{U}} \left(\prod_{j \in S \cap C} \mathbb{1}\{x_j = t_j\} \right) \left(\prod_{j \in S \cap N} \mathbb{1}\{x_j \leq t_j\} \right) \chi_a(x) \\ &= \frac{1}{\sqrt{|\mathcal{U}|}} \left(\prod_{j \in S \cap C} \omega_{m_j}^{a_j \cdot t_j} \right) \left(\prod_{j \in S \cap N} \left(\sum_{x_j=0}^{t_j} \omega_{m_j}^{a_j \cdot x_j} \right) \right) \left(\prod_{j \notin S} \left(\sum_{x_j=0}^{m_j-1} \omega_{m_j}^{a_j \cdot x_j} \right) \right).\end{aligned}$$

The final product on the right hand side is 0 unless $a_j = 0$ for all $j \notin S$. Therefore, $(WV)_{(S,t),a} = 0$ unless $\mathrm{supp}(a) \subseteq S$. Defining

$$f_{j,t}(a) := \sum_{x=0}^{t} \omega_{m_j}^{a \cdot x} = \begin{cases} t+1 & a = 0 \\ \frac{1-\omega_{m_j}^{a(t+1)}}{1-\omega_{m_j}^{a}} & a \neq 0 \end{cases},$$

we have that, if $\mathrm{supp}(a) \subseteq S$, then

$$\begin{aligned}(WV)_{(S,t),a} &= \frac{1}{\sqrt{|\mathcal{U}|}} \left(\prod_{j \in S \cap C} \omega_{m_j}^{a_j \cdot t_j} \right) \left(\prod_{j \in S \cap N} f_{j,t_j}(a_j) \right) \prod_{j \notin S} m_j \\ &= \frac{|\mathcal{U}_{[d] \setminus S}|}{\sqrt{|\mathcal{U}|}} \left(\prod_{j \in S \cap C} \omega_{m_j}^{a_j \cdot t_j} \right) \left(\prod_{j \in S \cap N} f_{j,t_j}(a_j) \right) \\ &= \frac{\sqrt{|\mathcal{U}|}}{|\mathcal{U}_S|} \left(\prod_{j \in S \cap C} \omega_{m_j}^{a_j \cdot t_j} \right) \left(\prod_{j \in S \cap N} f_{j,t_j}(a_j) \right).\end{aligned}$$

Next we fix $a \in A$, and proceed to compute $(U^*PWV)_{a,a}$. For any $S \in \mathcal{S}$ such that $\mathrm{supp}(a) \subseteq S$, we have

$$\begin{aligned}\sum_{t \in \mathcal{U}_S} \overline{U_{(S,t),a}} (WV)_{(S,t),a} &= \frac{\sqrt{|\mathcal{U}|}}{\kappa_a |\mathcal{U}_S|^2} \sum_{t \in \mathcal{U}_S} \left(\prod_{j \in S \cap C} \omega_{m_j}^{a_j \cdot t_j} \right) \left(\prod_{j \in S \cap N} (f_{j,t_j}(a_j) \overline{f_j(a_j)}) \right) \overline{\chi_a(t)} \\ &= \frac{\sqrt{|\mathcal{U}|}}{\kappa_a |\mathcal{U}_S|^2} \sum_{t \in \mathcal{U}_S} \left(\prod_{j \in S \cap N} (f_{j,t_j}(a_j) \omega_{m_j}^{-a_j \cdot t_j} \overline{f_j(a_j)}) \right) \\ &= \frac{\sqrt{|\mathcal{U}|} \cdot |\mathcal{U}_{S \cap C}|}{\kappa_a |\mathcal{U}_S|^2} \sum_{t \in \mathcal{U}_{S \cap N}} \left(\prod_{j \in S \cap N} (f_{j,t_j}(a_j) \omega_{m_j}^{-a_j \cdot t_j} \overline{f_j(a_j)}) \right) \\ &= \frac{\sqrt{|\mathcal{U}|} |\cdot |\mathcal{U}_{S \cap C}|}{\kappa_a |\mathcal{U}_S|^2} \prod_{j \in S \cap N} \left(\overline{f_j(a_j)} \sum_{t_j=0}^{m_j-1} f_{j,t_j}(a_j) \omega^{-a_j \cdot t_j} \right). \qquad (24)\end{aligned}$$

We now claim that $\sum_{t_j=0}^{m_j-1} f_{j,t_j}(a_j)\omega^{-a_j\cdot t_j} = m_j f_j(a_j)$. If $a_j = 0$, then $\omega_{m_j}^{a_j\cdot t_j} = 1$, and we have

$$\sum_{t_j=0}^{m_j-1} f_{j,t_j}(0) = \sum_{\ell=1}^{m_j} \ell = \frac{m_j(m_j+1)}{2} = m_j f_j(0).$$

If $a_j \neq 0$, then

$$\sum_{t_j=0}^{m_j-1} f_{j,t_j}(a_j)\omega^{-a_j\cdot t_j} = \sum_{t_j=0}^{m_j-1} \frac{\omega_{m_j}^{-a_j\cdot t_j} - \omega_{m_j}^{a_j}}{1-\omega_{m_j}^{a_j}} = -\frac{m_j\omega_{m_j}^{a_j}}{1-\omega_{m_j}^{a_j}} = \frac{m_j}{1-\omega_{m_j}^{-a_j}} = m_j f_j(a_j),$$

where we used that $\sum_{t_j=0}^{m_j-1} \omega_{m_j}^{-a_j\cdot t_j} = 0$ whenever $a_j \neq 0$. Plugging back into (24), we have

$$\sum_{t\in\mathcal{U}_S} \overline{U_{(S,t),a}}(WV)_{(S,t),a} = \frac{\sqrt{|\mathcal{U}|}\cdot|\mathcal{U}_{S\cap C}|\cdot|\mathcal{U}_{S\cap N}|}{\kappa_a|\mathcal{U}_S|^2} \prod_{j\in S\cap N} |f_j(a_j)|^2 = \frac{\sqrt{|\mathcal{U}|}}{\kappa_a|\mathcal{U}_S|} \prod_{j\in S\cap N} |f_j(a_j)|^2.$$

Adding up these equalities over $S \in \mathcal{S}$, and using that $U_{(S,t),a} = (WV)_{(S,t),a} = 0$ whenever $\mathrm{supp}(a) \not\subseteq S$, we have

$$\begin{aligned}
(U^*PWV)_{a,a} &= \sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}} \left(\frac{p(S)}{|\mathcal{U}_S|} \sum_{t\in\mathcal{U}_S} \overline{U_{(S,t),a}}(WV)_{(S,t),a} \right) \\
&= \frac{\sqrt{|\mathcal{U}|}}{\kappa_a} \sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2} \prod_{j\in S\cap N} |f_j(a_j)|^2 \\
&= \sqrt{|\mathcal{U}|} \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2} \prod_{j\in S\cap N} |f_j(a_j)|^2}.
\end{aligned}$$

Summing over $a \in A$, we finally get

$$\begin{aligned}
\gamma_F(P^{1/2}W) \geq \frac{1}{\sqrt{|\mathcal{U}|}}\|P^{1/2}W\|_{tr} &\geq \frac{1}{\sqrt{|\mathcal{U}|}}|\mathrm{tr}(P^{1/2}WY^*)| \\
&= \frac{1}{\sqrt{|\mathcal{U}|}}|\mathrm{tr}(U^*PWV)| \\
&= \sum_{a\in A} \sqrt{\sum_{\substack{S\in\mathcal{S}\\ S\supseteq\mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2} \prod_{j\in S\cap N} |f_j(a_j)|^2}, \qquad (25)
\end{aligned}$$

where the first two inequalities are by Lemma 5.2 and Lemma A.1, respectively.

The final step in the proof is to manipulate (25) to put it in more explicit form. To this end, we first compute $|f_j(a_j)|$

$$|f_j(a_j)| = \begin{cases} \frac{m_j+1}{2} & a_j = 0 \\ \frac{1}{2\sin\left(\frac{\pi a_j}{m_j}\right)} & a_j \neq 0, \end{cases}$$

where we use that $|1-\omega_{m_j}^{-a_j}|^2 = 2\left(1-\cos\left(\frac{2\pi a_j}{m_j}\right)\right) = 4\sin\left(\frac{\pi a_j}{m_j}\right)^2$. Therefore, for any a with $R := \mathrm{supp}(a)\cap C$ and $O := \mathrm{supp}(a)\cap N$, we have

$$\sum_{\substack{S\in\mathcal{S}\\ S\supseteq \mathrm{supp}(a)}} \frac{p(S)}{|\mathcal{U}_S|^2}\prod_{j\in S\cap N}|f_j(a_j)|^2 = \sum_{\substack{S\in\mathcal{S}\\ S\supseteq R\cup O}} \frac{p(S)}{|\mathcal{U}_S|^2}\left(\prod_{j\in O}\frac{1}{4\sin\left(\frac{\pi a_j}{m_j}\right)^2}\right)\left(\prod_{j\in (S\cap N)\setminus O}\frac{(m_j+1)^2}{4}\right)$$

$$= \left(\prod_{j\in O}\frac{1}{m_j^2\sin\left(\frac{\pi a_j}{m_j}\right)^2}\right)\sum_{\substack{S\in\mathcal{S}\\ S\supseteq R\cup O}}\frac{p(S)}{|\mathcal{U}_{S\cap C}|^2\cdot 4^{|S\cap N|}}\prod_{j\in (S\cap N)\setminus O}\left(1+\frac{1}{m_j}\right)^2. \tag{26}$$

Moreover, for any $R\subseteq C$ and $O\subseteq N$, summing over a such that $\mathrm{supp}(a)\cap C = R$ and $\mathrm{supp}(a)\cap N = O$, we get

$$\sum_{\substack{a\in A\\ \mathrm{supp}(a)\cap C=R\\ \mathrm{supp}(a)\cap N=O}}\prod_{j\in O}\frac{1}{m_j\sin\left(\frac{\pi a_j}{m_j}\right)} = \left(\prod_{j\in R}(m_j-1)\right)\prod_{j\in O}\left(\frac{1}{m_j}\sum_{a_j=1}^{m_j-1}\frac{1}{\sin\left(\frac{\pi a_j}{m_j}\right)}\right). \tag{27}$$

Combining (25), (26), and (27), and recalling $\zeta(m) := \frac{1}{m}\sum_{a=1}^{m-1}\frac{1}{\sin\left(\frac{\pi a}{m}\right)}$ we get

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

as we needed to show. □

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.

References

[AAC+22] 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. The 2020 census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, 2, 2022.

[ALNP24] Martin Aumüller, Christian Janos Lebeda, Boel Nelson, and Rasmus Pagh. PLAN: variance-aware private mean estimation. *Proc. Priv. Enhancing Technol.*, 2024(3):606–625, 2024.

[BCD+07] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In Leonid Libkin, editor, *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 273–282. ACM, 2007.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, November 2015.

[BUV14] Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 1–10. ACM, 2014.

[CKS22] Clement Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Journal of Privacy and Confidentiality*, 12(1), Jul. 2022.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[CMRT23] Christopher A. Choquette-Choo, Hugh Brendan McMahan, J. Keith Rush, and Abhradeep Guha Thakurta. Multi-epoch matrix factorization mechanisms for private machine learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 5924–5963. PMLR, 2023.

[CT65] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[DGK+23] Matthew Dawson, Badih Ghazi, Pritish Kamath, Kapil Kumar, Ravi Kumar, Bo Luan, Pasin Manurangsi, Nishanth Mundru, Harikesh Nair, Adam Sealfon, and Shengyu Zhu. Optimizing hierarchical queries for the attribution reporting API. In *AdKDD@KDD*, volume 3556 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023.

[DJY+24] Yuval Dagan, Michael I. Jordan, Xuelin Yang, Lydia Zakynthinou, and Nikita Zhivotovskiy. Dimension-free private mean estimation for anisotropic distributions. In *NeurIPS*, 2024.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 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*, pages 265–284. Springer, 2006.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210. ACM, 2003.

[DN04] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 528–544. Springer, 2004.

[DNT15] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discret. Comput. Geom.*, 53(3):650–673, 2015.

[DRS22] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022.

[ENU20] Alexander Edmonds, Aleksandar Nikolov, and Jonathan R. Ullman. The power of factorization mechanisms in local and central differential privacy. In *STOC*, pages 425–438. ACM, 2020.

[GAM19] Simson L. Garfinkel, John M. Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Commun. ACM*, 62(3):46–53, 2019.

[Haa85] Uffe Haagerup. Injectivity and decomposition of completely bounded maps. In *Operator algebras and their connections with topology and ergodic theory (Buşteni, 1983)*, volume 1132 of *Lecture Notes in Math.*, pages 170–222. Springer, Berlin, 1985.

[HKU25] Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. Normalized square root: Sharper matrix factorization bounds for differentially private continual counting. *CoRR*, abs/2509.14334, 2025.

[HRMS10] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1):1021–1032, 2010.

[HU25] Monika Henzinger and Jalaj Upadhyay. Improved differentially private continual observation using group algebra. In *SODA*, pages 2951–2970. SIAM, 2025.

[HUU23] Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. 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*, pages 5003–5039. SIAM, 2023.

[JAS20] JASON. Formal privacy methods for the 2020 census. Technical Report JSR-19-2F, U.S. Census Bureau, 2020. https://www.census.gov/programs-surveys/decennial-census/decade/2020/planning-management/plan/planning-docs/privacy-methods-2020-census.html [Accessed 14-July-2025].

[KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam D. Smith, and Jonathan R. Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 775–784. ACM, 2010.

[Leb23] Christian Janos Lebeda. *Differentially Private Release of Sparse and Skewed Data*. PhD thesis, IT University of Copenhagen, November 2023.

[Leb25] Christian Janos Lebeda. Better gaussian mechanism using correlated noise. In *2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 119–133, 2025.

[LM13] Chao Li and Gerome Miklau. Optimal error of query sets under the differentially-private matrix mechanism. In *ICDT*, pages 272–283. ACM, 2013.

[LM15] Chao Li and Gerome Miklau. Lower bounds on the error of query sets under the differentially-private matrix mechanism. *Theory Comput. Syst.*, 57(4):1159–1201, 2015.

[LMH+15] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.

[LSŠ08] Troy Lee, Adi Shraibman, and Robert Špalek. A direct product theorem for discrepancy. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 71–80. IEEE Computer Society, 2008.

[LUZ24] Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. Optimality of matrix mechanism on ℓ_p^p-metric. *CoRR*, abs/2406.02140, 2024.

[MMHM23] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Journal of Privacy and Confidentiality*, 13(1), Aug. 2023.

[MMS21] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3), Dec. 2021.

[MNT20] Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. Factorization norms and hereditary discrepancy. *International Mathematics Research Notices*, 2020(3):751–780, 2020.

[MSM19] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 4435–4444. PMLR, 2019.

[NT24] Aleksandar Nikolov and Haohua Tang. General gaussian noise mechanisms and their optimality for unbiased mean estimation. In *ITCS*, volume 287 of *LIPIcs*, pages 85:1–85:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[NTZ16] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: The small database and approximate cases. *SIAM J. Comput.*, 45(2):575–616, 2016.

[PUC+25] 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. Correlated noise mechanisms for differentially private learning. *CoRR*, abs/2506.08201, 2025.

[TUV12] Justin Thaler, Jonathan R. Ullman, and Salil P. Vadhan. Faster algorithms for privately releasing marginals. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 810–821. Springer, 2012.

[Uni21] United States Census Bureau. 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, 2021. [Accessed 14-July-2025].

[UV10] Jonathan R. Ullman and Salil P. Vadhan. Pcps and the hardness of generating synthetic data. *Electron. Colloquium Comput. Complex.*, TR10-017, 2010.

[XHT+25] Yingtai Xiao, Guanlin He, Levent Toksoz, Zeyu Ding, Danfeng Zhang, and Daniel Kifer. Residualplanner+: a scalable matrix mechanism for marginals and beyond. *arXiv preprint arXiv:2305.08175*, 2025.

[XHZK23] Yingtai Xiao, Guanlin He, Danfeng Zhang, and Daniel Kifer. An optimal and scalable matrix mechanism for noisy marginals under convex loss functions. *Advances in Neural Information Processing Systems*, 36:20495–20539, 2023.

[ZWL+21] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. Privsyn: Differentially private data synthesis. In *USENIX Security Symposium*, pages 929–946. USENIX Association, 2021.

A Missing Proofs from Section 5.1

Towards proving Lemmas 5.2 and 5.3, let us first recall a well-known characterization of the trace norm.

Lemma A.1. *For any two matrices X and Y of the same dimensions, we have $|\mathrm{tr}(XY^*)| \leq \|X\|_{tr}\|Y\|_{op}$. Moreover, for any matrix X, there is a matrix Y of the same dimensions with $\|Y\|_{op} = 1$, and such that*

$$\mathrm{tr}(XY^*) = \|X\|_{tr}\|Y\|_{op} = \|X\|_{tr}.$$

Proof. Let $X = U\Sigma V^*$ be the SVD of X, where Σ is an $r \times r$ diagonal matrix with non-negative entries, and U and V are matrices with r orthonormal columns each. Then

$$|\mathrm{tr}(XY^*)| = |\mathrm{tr}(\Sigma(U^*YV)^*)| \leq \sum_{i=1}^{r} \sigma_i |u_i^* Y v_i|,$$

where u_i is the i-th column of U (i.e., the i-th left singular vector of Y), and v_i is the i-th column of V (i.e., the i-th right singular vector of X). We have that

$$|u_i^* Y v_i| \leq \|u_i\|_2 \|Y\|_{op} \|v_i\|_2 = \|Y\|_{op},$$

so $\mathrm{tr}(XY^*) \leq \sum_{i=1}^{r} \sigma_i \|Y\|_{op} = \|X\|_{tr}\|Y\|_{op}$.

For the claim after "moreover", take U and V to be the matrices of left and right singular vectors of X, as above, and define $Y = UV^*$. Then

$$\mathrm{tr}(XY^*) = \mathrm{tr}(U^*XV) = \mathrm{tr}(\Sigma) = \|X\|_{tr},$$

and $\|UV^*\|_{op} = 1$. □

To prove Lemma 5.2, we also need a matrix version of the Cauchy-Schwarz inequality.

Lemma A.2. *For any two matrices X and Y for which the product XY is well defined,*

$$\|XY\|_{tr} \leq \|X\|_F \|Y\|_F,$$

and equality holds if and only if $X^*X = cYY^*$ for some real number $c \geq 0$.

Proof. Let Z be a matrix with $\|Z\|_{op} = 1$, and of the same dimensions as the product XY, so that $\mathrm{tr}(XYZ^*) = \|XY\|_{tr}$. We have

$$\|XY\|_{tr} = |\mathrm{tr}(X(ZY^*)^*)| \leq \|X\|_F \|ZY^*\|_F, \tag{28}$$

where we just used the standard Cauchy-Schwarz inequality, treating X and ZY^* as vectors, and $\mathrm{tr}((X)(ZY^*)^*)$ as the standard inner product between them. Now observe that

$$\|ZY^*\|_F^2 = \mathrm{tr}(Y(Z^*Z)Y^*) \leq \mathrm{tr}(YY^*) = \|Y\|_F^2. \tag{29}$$

This is because, by assumption, $\|Z^*Z\|_{op} = \|Z\|_{op}^2 \leq 1$, so $y_i(Z^*Z)y_i^* \leq y_i y_i^*$ for each row y_i of Y (seen as a row vector). Substituting gives us the required inequality.

If the inequality holds with equality, then (28) and (29) must also hold with equality. For the Cauchy-Schwarz inequality (28) to hold with equality, it must be the case that there is a number $c \in \mathbb{C}$ for which $X = c(ZY^*)$. For (29) to hold with equality, Z^*Z must act as the identity on the row space of Y, i.e., we must have $Z^*ZY^* = Y^*$. Then we have

$$X^*X = |c|^2 YZ^*ZY^* = |c|^2 YY^*,$$

as we needed to show (after renaming $|c|^2$ to c).

Finally, assume that $X^*X = cYY^*$ for some real number $c \geq 0$. If $c = 0$, then clearly $X = 0$ as well, so $\|XY\|_{tr} = 0 = \|X\|_F\|Y\|_F$. Let us assume then that $c > 0$. Then

$$(XY)(XY)^* = XYY^*X^* = \frac{1}{c}(XX^*)^2.$$

Therefore, the singular values of XY are equal to the eigenvalues of the positive semidefinite matrix $\frac{1}{\sqrt{c}}XX^*$, and we have

$$\begin{aligned}\|XY\|_{tr} &= \frac{1}{\sqrt{c}}\,\mathrm{tr}(XX^*) = \sqrt{\mathrm{tr}(XX^*)}\sqrt{\mathrm{tr}((1/c)\,XX^*)} \\ &= \sqrt{\mathrm{tr}(XX^*)}\sqrt{\mathrm{tr}(Y^*Y)} = \|X\|_F\|Y\|_F.\end{aligned}$$

Thus the inequality holds with equality. □

We can now prove Lemma 5.2.

Proof of Lemma 5.2. Let $W = LR$ be a factorization of W. Applying Lemma A.2 to $X := L$ and $Y := RS^{1/2}$, we have

$$\|WS^{1/2}\|_{tr} = \|L(RS^{1/2})\|_{tr} \leq \|L\|_F\|RS^{1/2}\|_F. \tag{30}$$

Let r_j be the j-th column of R. We have

$$\|RS^{1/2}\|_F^2 = \sum_{j=1}^{N} S_{j,j}\|r_j\|_2^2 \leq \left(\sum_{j=1}^{N} S_{j,j}\right) \max_{j=1}^{N}\|r_j\|_2^2 = \|R\|_{1\to 2}^2. \tag{31}$$

The inequalities (30) and (31) imply that $\|WS^{1/2}\|_{tr} \leq \|L\|_F\|R\|_{1\to 2}$ for any L and R such that $W = LR$. Minimizing the right hand side over all such choices of L and R then gives us $\|WS^{1/2}\|_{tr} \leq \gamma_F(W)$.

Suppose now that $LR = W$ is some factorization of W. Clearly $\|WS^{1/2}\|_{tr} = \gamma_F(W) = \|L\|_F\|R\|_{1\to 2}$ if and only if both (30) and (31) hold with equality. By Lemma A.2, (30) is tight if and only if $L^*L = cRSR^*$ for some $c \geq 0$. The inequality (31) is clearly tight if and only if $\|r_j\|_2 = \|R\|_{1\to 2}$ whenever $S_{j,j} \neq 0$. □

The proof of Lemma 5.3 is analogous to the proof of Lemma 5.2, but we use Lemma A.2 with $X := P^{1/2}L$ and $Y := RS^{1/2}$ instead.

B Discrepancy with the SVD Lower Bound of McKenna et al.

Theorem 12 in [MMHM23] claims the following formula holds for the singular value lower bound of weighted marginal queries. Here we use the notation from Section 5.2, rather the notation of McKenna et al.

$$\|P^{1/2}W\|_{tr} = \frac{1}{\sqrt{|\mathcal{U}|}} \sum_{R\subseteq[d]} |\mathcal{U}_R| \sqrt{\sum_{T\supseteq R} \frac{p(T)|\mathcal{U}_{[d]\setminus T}|}{|\mathcal{U}_T|}}\,, \tag{32}$$

where we define $|\mathcal{U}_\emptyset| = 1$, and set $p(T) = 0$ if $T \notin \mathcal{S}$. To see that this corresponds to Theorem 12 in [MMHM23], note that the bit vectors $a, b \in \{0,1\}^d$ in their notation correspond to sets R and T whose indicator vectors are, respectively, a and b; $c(\neg a)$ in their notation equals $|\mathcal{U}_R|$, while $c(b) = |\mathcal{U}_{[d]\setminus T}|$, and, with our normalization, $w(b) = \frac{p(R)}{|\mathcal{U}_R|}$.

Let us rewrite (32) to bring it closer to (20). By bringing the $1/\sqrt{|\mathcal{U}|}$ term inside the square root, we get

$$\begin{aligned}\|P^{1/2}W\|_{tr} &= \sum_{R\subseteq[d]} |\mathcal{U}_R| \sqrt{\sum_{T\supseteq R} \frac{p(T)|\mathcal{U}_{[d]\setminus T}|}{|\mathcal{U}_T|\cdot|\mathcal{U}|}} = \sum_{R\subseteq[d]} |\mathcal{U}_R| \sqrt{\sum_{T\supseteq R} \frac{p(T)}{|\mathcal{U}_T|^2}} \\ &= \sum_{R\subseteq[d]} \left(\prod_{j\in R} m_j\right) \sqrt{\sum_{T\supseteq R} \frac{p(T)}{|\mathcal{U}_T|^2}}\,. \end{aligned} \tag{33}$$

Now we can see that the coefficient in front of each square root on the right hand side in (33) is $\prod_{j\in R} m_j$ rather than $\prod_{j\in R}(m_j - 1)$ as in (20). In particular, (33) is always at least as large as (20), and would seem to contradict Lemma 5.4.

The reason for the discrepancy is an error in the proof of Theorem 12 in [MMHM23]. In their Theorem 9, McKenna et al. define, for each set $R \subseteq [d]$, a matrix $V(R)$ (or $V(a)$ in their notation) with $|\mathcal{U}_R|$ rows, where each row is an eigenvector of $W^T P W$ with eigenvalue $\kappa(R) := \sum_{T\supseteq R} \frac{p(T)|\mathcal{U}_{[d]\setminus T}|}{|\mathcal{U}_T|}$. The matrix is given by the formula $V(R) := \prod_{j=1}^d V_j(R)$, where $V_j(R)$ equals the $1 \times m_j$ all-ones matrix if $j \notin R$, and $J - m_j I$ if $j \in R$, for the $m_j \times m_j$ all-ones matrix J, and the $m_j \times m_j$ identity matrix I. From this, they infer that, for any $R \subseteq [d]$, $W^T P W$ has $|\mathcal{U}_R|$ singular values equal to $\sqrt{\kappa(R)}$, and add these singular values with these multiplicities to get (32). This argument, however, overcounts the singular values. In particular, the rows of $V(R)$ are not linearly independent unless $R = \emptyset$, so the eigenspace spanned by $V(R)$ is not necessarily of dimension $|\mathcal{U}_R|$. Indeed, notice that $J - m_j I$ has rank $m_j - 1$, and, therefore, $V(R)$ has rank $\prod_{j\in R}(m_j - 1)$. One needs to also verify that there are no non-trivial linear dependencies between the rows in different $V(R)$ matrices, but this turns out to be a non-issue since $V(R)V(R')^T = 0$ whenever $R \neq R'$. Correcting for the dimension of the rowspan of $V(R)$ in the McKenna et al. proof recovers our formula (20).

Let us consider a small example to illustrate this. Suppose that $d = 2$, $m_1 = m_2 = 2$, $\mathcal{S} = \{\{1\}, \{2\}\}$, and $p(\{1\}) = p(\{2\}) = \frac{1}{2}$. Then

$$W^T P W = \frac{1}{4}\begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}.$$

This matrix has one eigenvalue equal to 1, two eigenvalues equal to $\frac{1}{2}$, and one eigenvalue equal to 0. The eigenvectors are just the columns of $\tilde{V}$:

$$\tilde{V} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Therefore, the SVD lower bound is $\frac{1}{2}\|P^{1/2}W\|_{tr} = \frac{1}{2} + \frac{1}{2\sqrt{2}} + \frac{1}{2\sqrt{2}} + 0 = \frac{1+\sqrt{2}}{2}$. At the same time, (32) and (33) give $\frac{1}{2} + \sqrt{2}$ which is larger than the sensitivity of W, i.e. $\|W\|_{1\to 2} = \sqrt{2}$. The eigenmatrices in Theorem 9 of [MMHM23] are

$$V(\emptyset) = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix};$$

$$V(\{1\}) = \begin{pmatrix} -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix};$$

$$V(\{2\}) = \begin{pmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}.$$

Notice that the two rows of $V(\{1\})$ are colinear, as are the two rows of $V(\{2\})$, which is the cause of the overcounting.