A. Proof of Theorem 4.1

Proof. We reduce to the (unrelaxed) projection mechanism, which has the following guarantee proven by (Nikolov et al., 2013): for any dataset D consisting of n elements from a *finite* data universe $\mathcal{X}$, and for any set of m statistical queries q, the projection mechanism results in a dataset D' such that: $\sqrt{\frac{1}{m}||q(D') - q(D)||_2^2} \leq \alpha$ for

$$\alpha = O\left(\frac{(\ln(|\mathcal{X}|/\beta)\ln(1/\delta))^{1/4}}{\sqrt{\epsilon n}}\right).$$

Consider a finite data universe $\mathcal{X}^\eta = \{0, \eta, 2\eta, \ldots, 1\}^{d'}$ for some discretization parameter $0 < \eta < 1/k$. Given a dataset $D' \in \mathcal{X}^r$, let $D'_\eta \in \mathcal{X}^\eta$ be the dataset that results from "snapping" each real-valued $x \in D$ to its closest discrete valued point $x_\eta \in \mathcal{X}^r$. Observe that by construction, $||x - x(\eta)||_\infty \leq \eta$, and as a result, for k-way product query q_i, we have $|q_i(D') - q_i(D'_\eta)| \leq O(\eta k)$. Now let $\hat{D}' = \arg\min_{\hat{D}' \in (\mathcal{X}^r)^*} ||a - q(\hat{D}')||$ and $D'' = \arg\min_{D'' \in (\mathcal{X}^\eta)^*} ||a - q(D'')||$. From above, we know that $\sqrt{\frac{1}{m}||q(D'') - q(\hat{D}')||} \leq O(\eta k)$, and hence from an application of the triangle inequality, we have that $\sqrt{\frac{1}{m}||q(D) - q(\hat{D}')||_2^2} \leq O\left(\frac{(\ln(|\mathcal{X}^\eta|/\beta)\ln(1/\delta))^{1/4}}{\sqrt{\epsilon n}} + \eta k\right)$. Finally, for any dataset $\hat{D}' \in (\mathcal{X}^r)^*$, there exists a dataset $D' \in (\mathcal{X}^r)^{n'}$ such that $\sqrt{\frac{1}{m}||q(D') - q(\hat{D}')||_2^2} \leq O(\frac{\sqrt{\log k}}{\sqrt{n'}})$ (This follows from a sampling argument, and is proven formally in (Blum et al., 2008).) Hence, a final application of the triangle inequality yields:

$$\sqrt{\frac{1}{m}||q(D) - q(D')||_2^2} \leq$$

$$O\left(\frac{(\ln(|\mathcal{X}^\eta|/\beta)\ln(1/\delta))^{1/4}}{\sqrt{\epsilon n}} + \eta k + \frac{\sqrt{\log k}}{\sqrt{n'}}\right)$$

Choosing $\eta = \frac{\sqrt{\log k}}{k\sqrt{n'}}$ and noting that $|\mathcal{X}^\eta| = (\frac{1}{\eta})^{d'}$ yields the bound in our theorem.

□

B. Proof of Theorem 4.2

Proof. The privacy of Algorithm 2 follows straightforwardly from the tools we introduced in Section 2. First consider the case of $T = 1$. The algorithm makes m calls to the Gaussian mechanism, each of each satisfies ρ/m-zCDP by construction and Lemma 2.10. In combination, this satisfies ρ-zCDP by the composition Lemma (Lemma 2.6). It then makes a call to the relaxed projection algorithm RP, which is a postprocessing of the Gaussian mechanism, and hence does not increase the zCDP parameter, by Lemma 2.7. Hence the algorithm is ρ-zCDP, and by our choice of ρ and Lemma 2.8, satisfies (ϵ, δ) differential privacy.

Now consider the case of $T > 1$. Each iteration of the inner loop makes one call to report noisy max, and one call to the Gaussian mechanism. By construction and by Lemmas 2.10 and 2.12, each of these calls satisfies $\frac{\rho}{2TK}$-zCDP, and together by the composition Lemma 2.6, satisfy $\frac{\rho}{TK}$-zCDP. The algorithm then makes a call to the relaxed projection algorithm RP, which is a post-processing of the composition of the Gaussian mechanism with report noisy max, and so does not increase the zCDP parameter by Lemma 2.7. The inner loop runs $T \cdot K$ times, and so the entire algorithm satisfies ρ-zCDP by the composition Lemma 2.6. By our choice of ρ and Lemma 2.8, our algorithm satisfies (ϵ, δ) differential privacy as desired. □

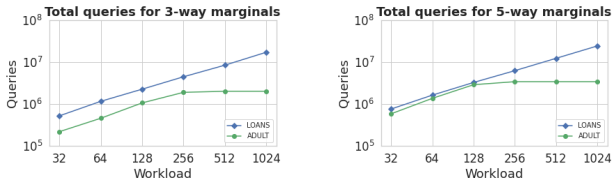
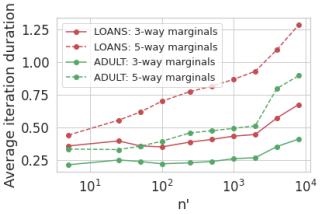


Figure 7. Total number of queries consistent with the selected random 3-way and 5-way marginals on ADULT and LOANS datasets. Y-axis in log scale.

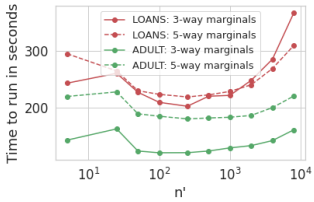
C. Additional Plots

Figure 7 provides the correspondence between the workload size and the number of marginal queries preserved in our experiments. Note that LOANS is a higher dimensional dataset, and so the number of queries continues to increase with the workload, whereas for large enough workloads, we saturate all available queries on ADULT.

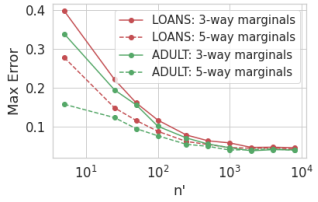
Figure 8 documents our investigation of the run-time and accuracy of our algorithm as a function of the synthetic dataset size n'. n' is a hyperparameter that we can use to trade of the representation ability of our synthetic data (larger n' allows the synthetic data to represent richer sets of answer vectors) with optimization cost. In Figure 8 we plot a) the run-time per iteration, b) the total run-time (over all iterations), and c) the error on several datasets and workloads, all as a function of n'. We find that although (as expected) the run-time per iteration is monotonically increasing in n', the overall run-time is not — it grows for sufficiently large n', but also grows for n' that is very small. This seems to be because as our optimization problem becomes sufficiently under-parameterized, the optimization becomes more difficult, and thus our algorithm needs to run for more iterations before convergence. We find that $n' = 1000$ is generally a good choice across datasets and query workloads, simultaneously achieving near minimal error and run-time. Hence



(a) Per-iteration run-time as a function of n'



(b) Total run-time as a function of n'



(c) Error as a function of n'

Figure 8. Run time and error as a function of the synthetic dataset size n'. At $n' = 1000$, both total run-time and overall error are near optimal across all settings.

we use $n' = 1000$ for all of our other experiments.

D. Linear Threshold Functions

In the body of the paper, we focused on *marginal* queries because of their centrality in the differential privacy literature. But our techniques easily extend to other classes of statistical queries — all that is required is that we can write python code to evaluate (a differentiable surrogate for) queries in our class. Here we do this for a natural class of linear threshold functions: t-out-of-k threshold functions.

Definition D.1. A t-out-of-k threshold query is defined by a subset $S \subseteq [d]$ of $|S| = k$ features, a particular value for each of the features $y \in \prod_{i \in S} \mathcal{X}_i$, and a threshold $t \leq k$. Given such a pair (S, y, t), the corresponding statistical query $q_{S,y,t}$ is defined as:

$$q_{S,y,t}(x) = \mathbb{1}(\sum_{i \in S} \mathbb{1}(x_i = y_i) \geq t)$$

Observe that for each collection of features S, there are $\prod_{i \in S} |\mathcal{X}_i|$ many t-out-of-k threshold queries for each threshold t.

In words, a t-out-of-k threshold query evaluates to 1 exactly when at least t of the k features indexed by S take the values indicated by y. These generalize the marginal queries that we studied in the body of the paper: A marginal query is simply the special case of a t-out-of-k threshold query for $t = k$.

To use our approach to generate synthetic data for t-out-of-k linear threshold functions, we need an extended differentiable query class for them. It will be convenient to work with the same one-hot-encoding function $h : \mathcal{X} \to \{0,1\}^{d'}$ from the body of the paper, that maps d-dimensional vectors of categorical features to d'-dimensional vectors of binary features. Our statistical queries are then binary functions defined on the hypercube. We can generically find a differentiable surrogate for our query class by polynomial interpolation: in fact for every boolean function that depends on k variables, there always exists a polynomial of degree k that matches the function on boolean variables, but also extends it in a differentiable manner to the reals. t-out-of-k threshold functions are such a class, and so can always be represented by polynomials of degree k.

```
import jax.numpy as np
def threeway_thresholded_marginals(D):
  return (D.shape[0] - np.einsum('ij,ik,il
      ->jkl', 1-D, 1-D, 1-D))/D.shape[0]
```

Figure 9. Python function used to compute (an extended equivalent differentiable query for) 1-out-of-3 linear threshold functions

Lemma D.2. Any boolean class of queries that depends on at most k variables (i.e. a 'k-junta') has an equivalent extended differentiable query that is a polynomial of degree k.

In our experiments we will consider 1-out-of-k queries (equivalently, disjunctions), which have an especially simple extended differentiable representation.

Definition D.3. Given a subset of features $T \subseteq [d']$, the 1-out-of-k polynomial query $q_T : \mathcal{X}^r \to \mathbb{R}$ is defined as: $q_T(x) = 1 - \prod_{i \in T}(1 - x_i)$.

It is easy to see that 1-out-of-k polynomials are extended differentiable queries equivalent to 1-out-of-k threshold queries. They are differentiable because they are polynomials. A 1-out-of-k threshold query corresponding to a set of k binary features T (i.e. the one-hot encoded indices for the categorical feature values y_i) evaluates to 0 exactly when *every* binary feature $x_i \in T$ takes value $x_i = 0$ — i.e. exactly when $\prod_{i \in T}(1 - x_1) = 1$. Our 1-out-of-$k$ polynomials are the negation of this monomial on binary valued inputs.

The code to evaluate such queries is similarly easy to write — see Figure 9.

We repeat our experiments on the Adult and Loans datasets using 1-out-of-3 threshold queries in place of 3-way marginals. All other experimental details remain the same. In Figure 10, we report the results on a workload of size 64, with δ fixed to $1/n^2$, and ϵ ranging from 0.1 to 1.0.

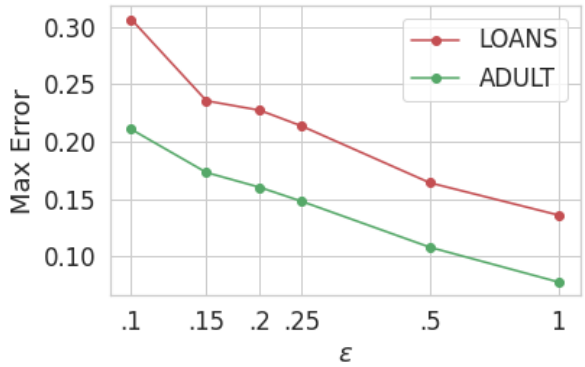


Figure 10. Max error for increasing ϵ of 1-out-of-3 threshold queries with workload 64