New Oracle-Efficient Algorithms for Private Synthetic Data Release

Giuseppe Vietri
University of Minnesota

Grace Tian
Harvard University

Mark Bun
Boston University

Thomas Steinke
IBM Research

Zhiwei Steven Wu
University of Minnesota

Abstract

We present three new algorithms for constructing differentially private synthetic data—a sanitized version of a sensitive dataset that approximately preserves the answers to a large collection of statistical queries. All three algorithms are *oracle-efficient* in the sense that they are computationally efficient given access to an optimization oracle, which can implemented using many existing (non-private) sophisticated optimization tools such as integer program solvers. While the accuracy of the synthetic data is contingent on the oracle's optimization performance, the algorithms satisfy differential privacy even in the worst case. For all three algorithms, we provide theoretical results as well as preliminary empirical evaluation, which shows that the algorithms can efficiently and accurately answer a large collection of queries on the Adult dataset.

1 Introduction

The wide range of personal data collected from individuals has facilitated many studies and data analyses that inform decisions related to science, commerce, and government policy. Since many of these rich datasets also contain highly sensitive personal information, there is a tension between releasing useful information about the population and compromising the privacy of individuals. In this work, we consider the problem of answering a large collection of statistical (or linear) queries subject to the constraint of differential privacy. Formally, we consider a data domain $\mathcal{X} = \{0,1\}^d$ of dimension d and a dataset $D \in \mathcal{X}^n$ consisting of the data of n individuals. Our goal is to approximately answer a large class of statistical queries $\mathcal{Q}$ about D. A statistical query is defined by a predicate $\phi\colon \mathcal{X} \to [0,1]$, and the query $q_\phi : \mathcal{X}^n \to [0,1]$ is given by $q(D) = \frac{1}{n}\sum_{i=1}^n \phi(D_i)$ and an approximate answer $a \in [0,1]$ must satisfy $|a - q(D)| \leq \alpha$ for some accuracy parameter $\alpha > 0$. To preserve privacy we work under the constraint of differential privacy [6]. Answering statistical queries provides the basis for a wide range of data analysis tasks, as many machine learning algorithms can be simulated using statistical queries [17].

An especially compelling way to perform private query release is to release *private synthetic data* — a sanitized version of the dataset that approximates all of the linear queries in the class $\mathcal{Q}$. Notable examples of private synthetic data algorithms are the SmallDB algorithm [1] and the private multiplicative weights (PMW) mechanism [14] (and its more practical variant MWEM (multiplicative weights with exponential mechanism) [13]), which can answer nearly exponentially many queries in the size of the input dataset and also achieve nearly optimal sample complexity [3]. Unfortunately, both algorithms involve maintaining a probability distribution over the data domain $\mathcal{X}$, and hence have running time that is exponential in the dimension d. Moreover, under standard cryptographic assumptions, this running time is necessary in the worst case [21, 22].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

To build more efficient solutions for constructing private synthetic data, we consider *oracle efficient* algorithms that rely on a black-box optimization subroutine. The optimization problem is NP-hard in the worst case. However, we invoke practical optimization heuristics for this subroutine (namely integer program solvers such as CPLEX and Gurobi). These heuristics work well on many real-world instances. Thus the algorithms we present are more practical than the worst-case hardness would suggest is possible. While the efficiency and accuracy of our algorithms are contingent on the solver's performance, differential privacy is guaranteed even if the solver runs forever or fails to optimize correctly.

Overview of our results. To describe our algorithms, we will first revisit a formulation of the query release problem as a zero-sum game between a data player who maintains a distribution $\hat{D}$ over $\mathcal{X}$ and a a query player who selects queries from $\mathcal{Q}$ [15, 10]. Intuitively, the data player aims to approximate the private dataset D with $\hat{D}$, while the query player tries to identify a query which distinguishes between D and $\hat{D}$. Prior work [15, 10] showed that any (approximate) equilibrium for this game gives rise to an accurate synthetic dataset. Moreover, existing algorithms including MWEM can be viewed as performing equilibrium computation for this game through *no-regret dynamics*: in each of a sequence of rounds, one player updates its strategy using a no-regret online learning algorithm, while the other player plays an approximate best response. In MWEM, the data player uses the multiplicative weights (MW) method as the no-regret algorithm, and the query player approximately best responds with the exponential mechanism (see appendix).

Our first two algorithms FEM and sepFEM follow the same pattern of no-regret dynamics in MWEM, but importantly replace the MW method with two different variants of the follow-the-perturbed-leader (FTPL) algorithm [16]—Non-Convex-FTPL [19] and Separator-FTPL [20]—both of which perturb the objective of an optimization problem and solve the perturbed problem via an optimization oracle. The two algorithms we design both satisfy (ε, δ)-differential privacy, and have error rates of $\tilde{O}(d^{3/4}\log^{1/2}|\mathcal{Q}|/(n\varepsilon)^{1/2})$ and $\tilde{O}\left(d^{5/8}\log^{1/4}|\mathcal{Q}|/(n\varepsilon)^{1/2}\right)$. Although the accuracy analysis requires repeated sampling from the FTPL distribution (and thus repeatedly solving perturbed integer programs), our experiments show that the algorithms remain accurate even with a much lower number of samples, which allows much more practical running time.

Our third algorithm takes the *dual* approach of MWEM and improves upon the existing algorithm DualQuery [10]. Unlike MWEM, DualQuery has the query player running MW over the query class $\mathcal{Q}$, which is often significantly smaller than the data domain $\mathcal{X}$, and has the data player playing best response, which can be computed non-privately by solving an integer program. Since the query player's MW distribution is a function of the private data, DualQuery approximates this distribution from a collection of samples drawn from it. Each draw from the MW distribution can be viewed as a single instantiation of the exponential mechanism, which provides a bound on the privacy loss. We improve DualQuery by leveraging the observation that the MW distribution changes slowly between rounds in the no-regret dynamics. Thus can reuse previously drawn queries to approximate the current MW distribution via rejection sampling. By using this technique, our algorithm DQRS (DualQuery with rejection sampling) reduces the number of times we draw new samples from the MW distribution and also the privacy loss, and hence improves the privacy-utility trade-off. We empirically demonstrate that DQRS improves the accuracy guarantee DualQuery. Even though they have worse accuracy performance than FEM and sepFEM, the dual algorithms DualQuery and DQRS run substantially faster, since they make many fewer oracle calls.

Organization. We describe the algorithms FEM and sepFEM in the main body of this extended abstract and defer the details of DQRS to the appendix.

2 Query Release Game

The goal in this paper is to privately solve the query release problem which goes as follows: Given a class of queries $\mathcal{Q}$ over a database D, we want to output a differentially private synthetic dataset $\widehat{D}$ such that for any query $q \in \mathcal{Q}$ we have low error:

$$\text{error}(\widehat{D}) = |q(D) - q(\widehat{D})| \leq \alpha$$

We model the query release problem as a zero-sum game between a data-player and a query player. The data player has action set equal to the data universe $\mathcal{X}$ and the query player has action set equal

to the query class $\mathcal{Q}$. The players' payoff for actions $x \in \mathcal{X}$ and $q \in \mathcal{Q}$ is given by:

$$A(x, q) := q(D) - q(x) \tag{1}$$

The data player wants minimizes the payoff $A(x, q)$ while the query player maximizes it. We make the assumption that $\mathcal{Q}$ is closed under negation. That is, for every query $q \in \mathcal{Q}$ there is a *negated query* $\bar{q} \in \mathcal{Q}$ where $\bar{q}(D) = 1 - q(D)$. If $\mathcal{Q}$ is not closed under negation, we can simply add negated queries to $\mathcal{Q}$. By the results of [15, 10] show that if $(\widehat{D}, \widehat{Q}) \in \Delta(\mathcal{X}) \times \Delta(\mathcal{Q})$ forms an α-approximate equilibrium of the game, then $\hat{D}$ is also 2α-accurate—that is for all $q \in \mathcal{Q}$, error$(\widehat{D}) = |q(D) - q(\widehat{D})| \leq 2\alpha$.

To compute such an equilibrium privately, we will utlitize no-regret online learning algorithm. The regret of an online learning algorithm is defined as follows.

Definition 2.1 (Regret)**.** Let $\mathcal{L}$ be an online learning algorithm with action domain $\mathcal{A}$. Every round t, $\mathcal{L}$ chooses an action $a_t \in \mathcal{A}$ and suffers a loss $\ell_t(a_t)$. For any sequence of T losses $\ell_1, \ldots, \ell_T$, the regret of $\mathcal{A}$, is given by

$$R_{\mathcal{L}}(T) = \sum_{t=1}^{T} \ell(a_t) - \min_{a} \sum_{t=1}^{T} \ell(a)$$

In this work, we take advantage of the oracle-efficient algorithms Non-Convex-FTPL from [12, 19] and Separator-FTPL from [20] with bounded expected regret $R_{\text{NC}}(T)$ and $R_{\text{Sep}}(T)$ respectively. We will defer their formal descriptons and regret guarantees to the appendix.

3 Oracle-Efficient Algorithms: FEM and sepFEM

Private No-Regret Dynamics. We first outline a general framework for solving the query release problem using no-regret dynamics, and view our two algorithms FEM and sepFEM as intantiations of this framework. The query player plays exponential mechanism which is an optimal private mechanism with known regret bounds and the data players plays a non-private no-regret algorithm. Suppose that the both query and data player play the zero-sum game as described for T rounds and incur regret equal to $R_q(T)$ and $R_d(T)$ respectively. Then by the seminal result of [9], the average of the data player actions constitudes a synthetic dataset with error bounded by $O\left(R_q(T) + R_d(T)\right)$.

Algorithm 1: General Framework of Private No-Regret Dynamic

Parameters: Target sampling error γ, Target failure probability β
Input: A dataset D, Queryset $\mathcal{Q}$, No-regret algorithm $\mathcal{A}$, Number of rounds T
$\varepsilon_0 = \frac{\varepsilon}{\sqrt{2T \log(1/\delta)}}$;
for $t \leftarrow 1$ **to** T **do**
  Generate $\widehat{D^t}$ with no regret algorithm $\mathcal{A}$;
  Construct $(1/n)$-sensitive function $S_t(D, q) = q(D) - q(\widehat{D^t})$;
  Sample: $q_t \sim \mathcal{M}_E(D, S_t, \mathcal{Q} \cup \bar{\mathcal{Q}}, \varepsilon_0)$;
  Algorithm $\mathcal{A}$ incurs loss $A(\widehat{D}, q_t)$;
end
Output: $\cup_t \widehat{D^t}$

We provide the privacy guarantee that works for any no-regret algorithm $\mathcal{A}$.

Theorem 1 (Privacy)**.** *Let $0 < \delta < 1$. For any no-regret algorithm $\mathcal{A}$, Algorithm 1 is (ε, δ)-differentially private.*

But the accuracy depends on the choice of $\mathcal{A}$. We now plug in two different no-regret algorithms into 1 and prove the corresponding accuracy theorems.

Instantiations with FTPL algoritms. We present a general theorem for the accuracy bound of algorithm 1 given access to an algorithm $\mathcal{A}$ with sublinear regret.

Theorem 2 (Accuracy). *Suppose that the data player in algorithm 1 generates the sequence* $\widehat{D} = (x_1, \ldots, x_{Ts})$ *by sampling from algorithm* $\mathcal{A}$ *with expected regret* $R_{\mathcal{A}}(T)$ *with respect to minimizing payoff function* $A : \mathcal{X} \times \mathcal{Q} \to [0, 1]$ *from equation 1. Let the regret of the query player be at most* $R_{EM}(T)$ *with probability* $1/2\beta$. *Then the error on* $\widehat{D}$ *is bounded by,*

$$\sup_{q \in \mathcal{Q}} |q(D) - q(\widehat{D})| \leq \frac{R_{\mathcal{A}}(T) + R_{EM}(T)}{T} + \gamma$$

We can get different algorithms for solving the query release problem by selecting the no-regret strategy played by the data player. The next two corollaries provide the accuracy bound for algorithm 1 when the data player plays according to strategy Non-Convex-FTPL and Separator-FTPL .

Corollary 2.1 (FEM Accuracy). *With probability at least* $1-\beta$, *the algorithm* FEM *finds a synthetic database that answers all queries in* $\mathcal{Q}$ *with additive error*

$$\alpha = \widetilde{O}\left(\frac{d^{3/4}\log^{1/2}|\mathcal{Q}| \cdot \log^{1/2}(1/\delta)\log^{1/2}(1/\beta)}{n^{1/2}\varepsilon^{1/2}}\right)$$

Corollary 2.2 (sepFEM Accuracy). *With probability at least* $1 - \beta$, *algorithm* sepFEM *finds a synthetic database that answers all queries in* $\mathcal{Q}$ *with additive error*

$$\alpha = O\left(\frac{d^{5/8}\log^{1/4}|\mathcal{Q}| \cdot \log^{1/2}(1/\delta)\log^{1/2}(1/\beta)}{n^{1/2}\varepsilon^{1/2}}\right)$$

4 Experiments on the Adult dataset

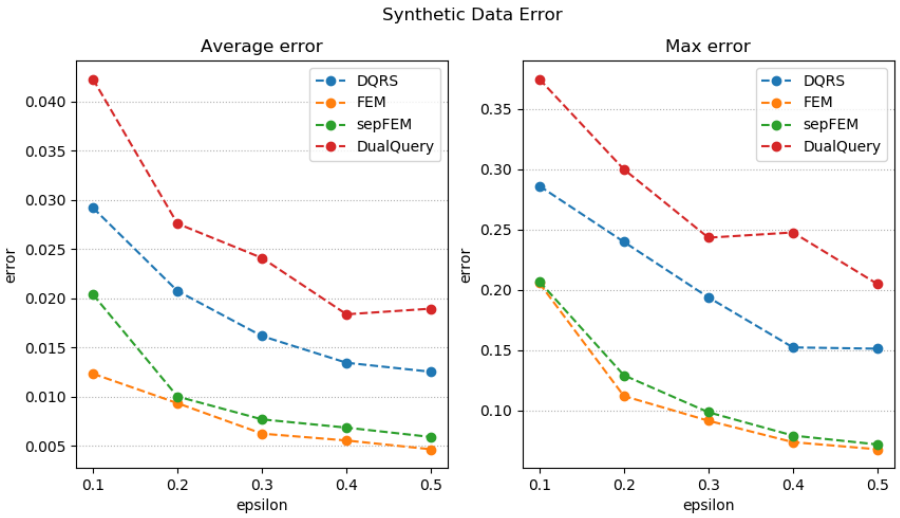

Figure 1: Average error and maximum error on Adult dataset . Comparison for different levels of privacy where the number of queries is 100,000.

We evaluate the algorithms presented in this paper on a $100,000$ 3-way marginal queries over the Adult dataset from the UCI repository [4]. For this experiment we used the first 6 categorical features of Adult and did one-hot encoding of the features resulting in $d = 57$ binary features. We compare the performance of the algorithms presented in this paper againts DualQuery with parameters as in [11]. To measuse the accuracy of the algorithms we used the average error $\left(\frac{1}{|\mathcal{Q}|}\sum_j |q_j(D) - q_j(\widehat{D})|\right)$ and the max error $\left(\max_j |q_j(D) - q_j(\widehat{D})|\right)$. The implementation was writen in python, and we used the Gurobi solver for mixed-integer-programming. We ran the experiments on a machine with 4-core Opteron processor and 192 Gb of ram.

References

[1] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 609–618, New York, NY, USA, 2008. ACM.

[2] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pages 635–658, Berlin, Heidelberg, 2016. Springer.

[3] Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.*, 47(5):1888–1938, 2018.

[4] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[5] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, July 2006.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

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

[8] Y. Freund and R.E. Schapire. Game theory, on-line prediction and boosting. *Conference on Computational Learning Theory (CoLT)*, pages 325–332, 1996.

[9] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996.*, pages 325–332, 1996.

[10] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1170–1178, 2014.

[11] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, pages 1170–1178, 2014.

[12] Alon Gonen and Elad Hazan. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018.

[13] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 2348–2356, 2012.

[14] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 61–70, 2010.

[15] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 341–350, 2013.

[16] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

[17] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.

[18] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–103, 2007.

[19] Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *CoRR*, abs/1903.08110, 2019.

[20] Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E. Schapire. Efficient algorithms for adversarial contextual learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2159–2168. JMLR.org, 2016.

[21] Jonathan Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. *SIAM J. Comput.*, 45(2):473–496, 2016.

[22] Jonathan Ullman and Salil P. Vadhan. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 400–416, 2011.

A Differential Privacy Tools

Definition A.1. A mechanism $M : \mathcal{X} \to R$ satisfies (ε, δ)-differential privacy if for every $S \subset R$ and for all neighboring datasets $D, D' \in \mathcal{X}$, the following holds:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta$$

Definition A.2 (Sensitivity)**.** The sensitivity of a function f with range R is

$$\Delta_f := \max_{D,D':|D \triangle D'|=1, r \in R} |f(D,r) - f(D',r)|.$$

Definition A.3 (Linear threshold functions)**.** A linear threshold function $\phi_p : \mathcal{X} \to \{0,1\}$, is a linear mapping defined by a real valued set P and vector coefficient ϕ. With

$$\phi_p(x) = \begin{cases} 1 & \text{if } \langle \phi, x \rangle \in P \\ 0 & \text{otherwise} \end{cases}$$

Definition A.4 (Statistical linear queries)**.** Given as predicate a linear threshold function ϕ, the linear query $q_\phi : \mathcal{X}^n \to [0,1]$ is defined by

$$q_\phi(D) = \frac{\sum_{x \in D} \phi(x)}{|D|}$$

Theorem 3 (Advanced Composition)**.** *A set of k mechanisms $\mathcal{M}_1, \ldots \mathcal{M}_k$ where each $\mathcal{M}_k$ is ϵ_k -differentially private. Satisfies (ϵ', δ)-differential privacy under k-fold adaptive composition for:*

$$\epsilon' = \sum_{i=1}^{k} \epsilon_i (e^{\epsilon_i} - 1) + \sqrt{\left(\sum_{i=1}^{k} \epsilon_i^2 \right) \ln\left(\frac{1}{\delta}\right) \frac{1}{2}}$$

Proof. For some mechanism $\mathcal{M}$ denote the privacy loss for all neighboring dataset D, D' and all outcomes $r \in \mathcal{R}$ as:

$$\mathcal{L}_{\mathcal{M}}(r) = \ln\left(\frac{\Pr[\mathcal{M}(D)] = r)}{\Pr[\mathcal{M}(D')] = r)} \right)$$

And let the privacy loss for running k mechanism be:

$$\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)}(r) = \sum_{i=1}^{k} \mathcal{L}_{\mathcal{M}_i}(r)$$

Since each $\mathcal{M}_i$ is ϵ_i-differentially private we have for all r that $\mathcal{L}_{\mathcal{M}_i}(r) \leq \epsilon_i$ and, using lemma (ref here), $\mathbb{E}[\mathcal{L}_{\mathcal{M}_i}(r)] \leq \epsilon_i(e_i^{\epsilon} - 1)$. The privacy loss is a function that depends on the randomized output of the private mechanisms r_i. Now we apply Azuma's inequality in Lemma **??**. Let $r_1, \ldots, r_{i-1}$ be the output of mechanisms $\mathcal{M}_1, \ldots \mathcal{M}_{i-1}$ then for all $a, a' \in \mathcal{R}$

$$\left| \mathbb{E}\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)} | r_1, \ldots, r_{i-1}, r = a \right] - \mathbb{E}\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)} | r_1, \ldots, r_{i-1}, r = a' \right] \right| \leq \epsilon_i$$

Then by Azuma's:

$$\Pr\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)}(r_1, \ldots, r_k) \geq \mathbb{E}\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)} \right] + t \right] \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^{k} \epsilon_i^2} \right)$$

Letting $\mathbb{E}\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)} \right] = \sum_{i=1}^{k} \epsilon_i (e^{\epsilon_i} - 1)$ and $t = \sqrt{\left(\sum_{i=1}^{k} \epsilon_i^2 \right) \ln\left(\frac{1}{\delta}\right) \frac{1}{2}}$

$$\Pr\left[\mathcal{L}_{\left(\cup_i^k \mathcal{M}_i\right)}(r_1, \ldots, r_k) \geq \sum_{i=1}^{k} \epsilon_i (e^{\epsilon_i} - 1) + \sqrt{\left(\sum_{i=1}^{k} \epsilon_i^2 \right) \ln\left(\frac{1}{\delta}\right) \frac{1}{2}} \right] \leq \delta$$

□

In this work we make the assumption that we have access to an optimization oracle such that, given a set of statistical linear queries, it outputs a data record from $\mathcal{X}$ such that it approximatelly maximizes then answer score on the query set.

Definition A.5 (Approximate Linear Optimization Oracle)**.** Given as input a set of n statistical linear queries $\{q_i\}$ and a d-dimentional vector σ, an α-approximate linear optimization oracle outputs

$$\hat{x} \in \arg\min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^{n} q_i(x) - \langle x, \sigma \rangle \right\}$$

Definition A.6 (Exponential Mechanism [18])**.** Given some database D, arbitrary range $\mathcal{R}$, and score function S, the exponential mechanism $\mathcal{M}_E(D, S, \mathcal{R}, \varepsilon)$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{\varepsilon S(D,r)}{2\Delta_S}\right)$, where Δ_S is the sensitivity of S.

Lemma 4 (Theorem 3.12 in [7])**.** *The exponential mechanism $\mathcal{M}_E(x, S, \mathcal{R})$ is $(\varepsilon, 0)$-differentially private.*

Theorem 5 (Exponential Mechanism Error - Corollary 3.12 in [7])**.** . *Fixing a database x, let $OPT_s(x)$ denote the max score function s. Then, with probability $1 - \beta$ the error is bounded by:*

$$OPT_s(x) - s(\mathcal{M}_E(x, u, \mathcal{R})) \leq \frac{2GS_s}{\varepsilon} \left(\ln |\mathcal{R}|/\beta\right)$$

B Missing Proof in Section 2

B.1 Equilbrium of the Game

Here we show that solving the query release game reduces to solving for the equilibrium of the zero sum game defined above. We make use von Neumann's minimax theorem which states that any zero-sum game has a unique value. Let $\Delta(\mathcal{X})$ and $\Delta(\mathcal{Q})$ be the set of probability distributions over $\mathcal{X}$ and $\mathcal{Q}$. According to von Neumann's minimax theorem, if each player plays from a probability distribution over their actions ($u \in \Delta(\mathcal{X})$ is the data player's action and $w \in \Delta(\mathcal{Q})$ is the query player's action), then:

$$\min_{u \in \Delta(\mathcal{X})} \max_{w \in \Delta(\mathcal{Q})} A(u, w) = \max_{w \in \Delta(\mathcal{Q})} \min_{u \in \Delta(\mathcal{X})} A(u, w) = v_A$$

where

$$A(u, w) := \mathbb{E}_{x \sim u, q \sim w} A(x, q)$$

is the expected payoff and v_A is the value of the game.

Definition B.1 (α-approximate equilibrium)**.** Let $\alpha > 0$. Let A be the payoffs for a two player, zero-sum game with action sets $\mathcal{X}, \mathcal{Q}$. Then a pair of strategies $u^* \in \Delta(\mathcal{X})$ and $w^* \in \Delta(\mathcal{Q})$ form an α-approximate mixed Nash equilibrium if

$$A(u^*, w) \leq v_a + \alpha \text{ and } A(u, w^*) \geq v_A - \alpha$$

for all strategies $u \in \Delta(\mathcal{X})$ and $w \in \Delta(\mathcal{Q})$. Note that $A(u^*, w)$ denotes the expected payoff for the best (minimizing) data response u^* and $A(u, w^*)$ denotes the expected payoff for best (maximizing) query response w^*.

Finally, the the following theorem shows that the α-approximate nash equilibrium is a solution for the query release game:

Theorem 6. *If (u^*, w^*) is the α-approximate equilibrium of a zero sum game between a data-player with action set $\mathcal{X}$ and a query player with actions set $\mathcal{Q} \cup \bar{\mathcal{Q}}$ then for every $q \in \mathcal{Q} \cup \bar{\mathcal{Q}}$, we have*

$$|q(u^*) - q(D)| \leq \alpha$$

Proof of Lemma 6.

Proof. We first show that the value of the game v_A is zero. Let $\widehat{D} \in \Delta(\mathcal{X})$ be the normalized histogram distribution of D. We consider the following cases:

1. Any query w and any data strategy over $\widehat{D}$ has zero payoff:

$$A(\widehat{D}, w) = \mathbb{E}_{x\sim\hat{D},q\sim w}[q(D) - q(X)] = 0.$$

Since the query player wants to maximize the payoff, it follows that the $v_A \leq 0$:

$$v_A = \max_{w\in\Delta(\mathcal{Q})}\left[\min_{u\in\Delta(\mathcal{X})} A(u,w)\right] \leq \max_{w\in\Delta(\mathcal{Q})} A(\widehat{D},w) = \max_{w\in\Delta(\mathcal{Q})} 0 = 0$$

2. For any data strategy u, the payoff of the query q is equal to the negation of the payoff of the negated query $\bar{q}$, i.e. $A(u,q) = -A(u,\bar{q})$:

$$A(u,\bar{q}) = \mathbb{E}_{x\sim u}[\bar{q}(D) - \bar{q}(x)] = \mathbb{E}_{x\sim u}[\bar{q}(x) - \bar{q}(D)] = -A(u,\bar{q}).$$

Because the query strategy places equal weight on q and $\bar{q}$, the expected payoff $A(u,q)$ is zero. Since the data player minimizes the payoff, $v_a \geq 0$:

$$v_A = \min_{u\in\Delta(\mathcal{X})}\left[\max_{w\in\Delta(\mathcal{Q})} A(u,w)\right] \geq \min_{u\in\Delta(\mathcal{X})} A(u,q) = \min_{u\in\Delta(\mathcal{X})} 0 = 0$$

Thus $v_A = 0$ as desired. Let (u^*, w^*) by the α-approximate equilibrium. From the left inequality in Definition B.1, we know that when the data player plays u^* and the query player plays q, the expected payoff is $A(u^*, w) = q(D) - q(u^*) \leq \alpha$. Likewise, if the data player plays u^* and the query player plays the negated query $\bar{q}$, the expected payoff is $A(u^*,\bar{q}) \leq \alpha$ and $A(u^*,\bar{q}) = \bar{q}(D) - \bar{q}(u^*) = 1 - q(D) - (1 - q(u^*)) = -q(D) + q(u^*)$. Thus, we have $q(D) - q(u^*) \geq -\alpha$ Therefore, we have $|q(u^*) - q(D)| \leq \alpha$ as desired, so the setup of the α-approximate equilibrium game satisfies the query release problem. Since the approximate data equilibrium u^* is also the synthetic data, we just need to find u^*, which will then give us a query answer $q(u^*)$ that is accurate and private. □

The next theorem by Freund and Schapire shows that in a zero-sum game, if the two players maintains a distribution using a no-regret algorithm, then both of the players' actions converge to an α-approximate equilibrium.

Theorem 7 (Freund and Schapire[8])**.** *Let $\alpha > 0$, and let $A(i,j) \in [-1,1]^{m\times n}$ be the payoff matrix for a zero-sum game. Suppose the first player uses a no-regret strategy over their actions to play distributions $p^1, \ldots, p^T$ and obtains average regret $R_1(T)$, while the second player plays approximate best responses $x^1, \ldots, x^T$ with regret $R_2(T)$. If T is selected such that $R_1(T) = R_2(T)$, then the empirical distributions*

$$\frac{1}{T}\sum_{t=1}^{T} p^t \text{ and } \frac{1}{T}\sum_{t=1}^{T} x^t$$

form an $(R_1(T) + R_2(T))$-approximate Nash equilibrium.

From theorem 7 and 6, it follows that having the data player and the query player playing algorithms with known regret bounds leads to a solution to the query release game. In the next section we introduce a general framework for analysing private no-regret dynamics.[5].

C Missing Proof in Section 3

Proof of Theorem 1.

Proof. By the advanced composition theorem, running a composition of k ε_0-private mechanisms is (ε,δ)-private for

$$\varepsilon = \sqrt{2k\log(1/\delta)}\varepsilon_0 + k\varepsilon_0(\exp(\varepsilon_0) - 1)$$

The privacy cost of each round is ε_0 and there are $k = T$ rounds. We plug in ε_0 and k accordingly into the advanced composition theorem to get

$$\varepsilon = \sqrt{2T\log(1/\delta)}\varepsilon_0 + T\varepsilon_0(\exp(\varepsilon_0) - 1).$$

Since $\exp(\varepsilon_0) \leq 1 + 2\varepsilon_0$ for $0 < \varepsilon_0 < 1$, we can simplify the expression to get $\epsilon \leq \sqrt{2T\log(1/\delta)}\varepsilon_0 + 2T{\varepsilon_0}^2$. Solving for ε_0 we find that

$$\begin{aligned}
\varepsilon_0 &= \frac{\sqrt{2T\log(1/\delta) + 8T\varepsilon} - \sqrt{2T\log(1/\delta)}}{4T} \\
&= \frac{1}{4T} \cdot \frac{8T\varepsilon}{\sqrt{2T\log(1/\delta) + 8T\varepsilon} + \sqrt{2T\log(1/\delta)}} \\
&= \frac{2\varepsilon}{\sqrt{2T\log(1/\delta)}\left(1 + \sqrt{1 + \frac{4\varepsilon}{\log(1/\delta)}}\right)} \\
&= \frac{\varepsilon}{\frac{\sqrt{2}}{2}\left(1 + \sqrt{1 + \frac{4\varepsilon}{\log(1/\delta)}}\right)\sqrt{T\log(1/\delta)}} \\
&< \frac{\varepsilon}{\frac{\sqrt{2}}{2}(1+1)\sqrt{T\log(1/\delta)}}
\end{aligned}$$

Thus we get ε_0 as desired. □

Next we describe the details of algorithm FEM and the accuracy guarantee proof.

Algorithm 2: FEM

Input: A dataset D, A queryset $\mathcal{Q}$

$s = \frac{8\log(4T/\beta)}{\alpha^2}$;

$\eta = \sqrt{\frac{1+\log d}{125dT}}$;

Let $\sigma_j^t \sim Exp(\eta)$;

Sample s data values $\{x_j^t\}$ according to $\arg\max_x \left\{\sum_{i=1}^{t-1} q_i(x) - x\sigma_j^t\right\}$;

Output: $\widehat{D^t} := \{x_1^t, x_2^t, \ldots, x_s^t\}$

The accuracy proof proceeds in two steps. First we show that the sample distribution $\widehat{D^t}$ played by the data player is close the true distribution D^t and we show that the query player plays an approximate best response. Then, by theorem 7, we show that algorithm 2 finds an approximate equilibrium.

We use Chernoff bound.

Lemma 8 (Chernoff bound [7])**.** *Let $X_1, \ldots, X_m$ be i.i.d random variables such that $0 \leq X_i \leq 1$ for all i. Let $S = \frac{1}{m}\sum_{i=1}^m X_i$ denote their mean and let $\mu = \mathbb{E}[S]$ denote their expected mean. Then,*

$$Pr\left[|S - \mu| > T\right] \leq 2\exp\left(-2mT^2\right)]$$

Lemma 9. *Let $\beta \in (0,1)$ and let D^t be the true distribution over $\mathcal{X}$. Suppose we draw*

$$s = \frac{8\log(4T/\beta)}{\alpha^2}$$

samples $\{x_i^t\}$ from D^t to form $\widehat{D^t}$. Then with probability at least $1 - \beta/2$, we have

$$\left|\frac{1}{s}\sum_{i=1}^{s} q(x_i^t) - q(D^t)\right| < \frac{\alpha}{4}$$

for all $0 \leq t \leq T$

Proof. For any fixed t, note that $\frac{1}{s}\sum_{i=1}^s q(x_i^t)$ is the average of the random variables $q(x_1^t), q(x_2^t), \ldots, q(x_s^t)$. Also $\mathbb{E}[q(x^t)] = q(D^t)$ for all $0 \leq t \leq T$. Thus by the Chernoff bound and our choice of s,

$$\Pr\left[\left|\frac{1}{s}\sum_{i=1}^{s} q(x_i^t) - q(D^t)\right| > \frac{\alpha}{4}\right] \leq 2\exp\left(-s\alpha^2/8\right) = \beta/2T.$$

A union bound over all T rounds gives a total fail probability of at most $\beta/2$ as desired. □

Lemma 10 (Data Player's Regret)**.** *Given that the data player plays the distributions $\widehat{D^1}, \widehat{D^2} \dots \widehat{D^T}$, the data player achieves an average regret bound of*

$$\frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) - \min_{x\in\mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} A(x, q^t) \leq \frac{\alpha}{4} + (125 + 5\sqrt{5})\sqrt{\frac{d^3(1+\log d)}{T}}$$

for all T rounds with probability at least $1 - \beta/2$.

Proof. We calculate the average error per round of the data and query player to find the value of α, the additive error of each query, with probability at least $1 - \beta$ using Theorem 7.

Because $\widehat{D^t}$ approximates D^t with probability at least $1 - \beta/2$, we calculate the average regret per round when the data player plays with the true distribution D^t. We assume that our oracle is efficient so $\alpha' = 0$. Since the dimension of the perturbation is the dimension of the dataset, $d = \log \mathcal{X}$. We also have $D = 1$ and $L = 1$. From the FTPL regret bound [12], since $\mathbb{E}_\sigma\left[\sum_{t=1}^{T} A(x^t, q^t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} A(x^t, q^t)\right] = \sum_{t=1}^{T} A(D^t, q^t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} A(x, q^t)$, the average regret per round for the data player playing the true distribution D^t has the following bound:

$$\frac{1}{T}\sum_{t=1}^{T} A(D^t, q^t) - \min_{x\in\mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} A(x, q^t) \leq 125\eta d^2 + \frac{(1+\log d)d}{T\eta}$$

.

From Lemma 9, we know that with probability at least $1-\beta/2$, the average error per round of sample distribution $\widehat{D^t}$ from the true distribution D^t is $\alpha/4$. Hence, with probability at least $1 - \beta/2$, the average regret per round for the data player playing the sample distribution $\widehat{D^t}$ is

$$\frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) - \min_{x\in\mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} A(x, q^t) \leq \frac{\alpha}{4} + 125\eta d^2 + \frac{(1+\log d)d}{T\eta}$$

Setting $\eta = \sqrt{\frac{1+\log(d)}{125Td}}$, we have

$$\frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) - \min_{x\in\mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} A(x, q^t) = \frac{\alpha}{4} + (125 + 5\sqrt{5})\sqrt{\frac{d^3(1+\log(d))}{T}}$$

as desired. □

We next calculate the upper bound of the average error per round for the query player playing the exponential mechanism with probability at least $1 - \beta/2$.

Lemma 11 (Query Player's Regret)**.** *The query player achieves an average regret bound of*

$$\max_{q\in\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) - \frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) \leq \frac{2/n}{\varepsilon_0}\ln(2T|\mathcal{Q}|/\beta) = \frac{2\sqrt{2}}{n\varepsilon}\sqrt{T\log(1/\delta)}\ln\left(\frac{2T|\mathcal{Q}|}{\beta}\right)$$

for all T rounds with probability $1 - \beta/2$.

Proof. Since the sensitivity of the query player's score function GS_s is $1/n$, then with probability $1 - \beta/2T$ the error for each is round is at most $\frac{2/n}{\varepsilon}\ln(2T|\mathcal{Q}|/\beta)$ by Theorem 5. Applying union bound over T rounds, with probability $1 - \beta/2$ the query player's average regret for T rounds is

$$\max_{q\in\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) - \frac{1}{T}\sum_{t=1}^{T} A(\widehat{D^t}, q^t) \leq \frac{2/n}{\varepsilon_0}\ln(2T|\mathcal{Q}|/\beta) = \frac{2\sqrt{2}}{n\varepsilon}\sqrt{T\log(1/\delta)}\ln\left(\frac{2T|\mathcal{Q}|}{\beta}\right)$$

□

Our final accuracy guarantee follows.
Proof of Theorem 2.1.

Proof. From Lemma 10 and Lemma 11, let $R_D(T)$ and $R_Q(T)$ be the upper bounds for the average error of the data and query player respectively with probability at least $1-\beta/2$. Then, with probability at least $1-\beta$ due to the union bound over 2 events, α is the average regret for all rounds by Theorem 7:

$$\begin{aligned}\alpha &= R_D(T) + R_Q(T)\\ &= \frac{\alpha}{4} + (125+5\sqrt{5})\sqrt{\frac{d^3(1+\log d)}{T}} + \frac{2\sqrt{2}}{n\varepsilon}\sqrt{T\log(1/\delta)}\ln\left(\frac{2T|\mathcal{Q}|}{\beta}\right)\end{aligned}$$

Setting $T = \frac{125+5\sqrt{5}}{2\sqrt{2}}\frac{n\varepsilon\cdot[d^3(1+\log d)]^{1/2}}{\log\frac{2|\mathcal{Q}|}{\beta}\log^{1/2}(1/\delta)}$, we have

$$\frac{3\alpha}{4} < 2(250\sqrt{2}+10\sqrt{10})^{1/2}\frac{\left[d^3(1+\log d\right]^{1/4}\log^{1/4}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\left[2\log^{1/2}\left(\frac{2|\mathcal{Q}|}{\beta}\right)+\log^{-1/2}\left(\frac{2|\mathcal{Q}|}{\beta}\right)\ln T\right]$$

$$\alpha = O\left(\frac{d^{3/4}\cdot\log^{1/4}(1/\delta)\log^{1/2}\left(\frac{2|\mathcal{Q}|}{\beta}\right)}{n^{1/2}\varepsilon^{1/2}}\cdot\text{polylog}\left(d,n,\varepsilon,\log(1/\delta),\log|\mathcal{Q}|,\log(1/\beta)\right)\right)$$

□

Definition C.1. A set $\text{sep}(\mathcal{Q})$ is a small-separator for queries $\mathcal{Q}$ if for any two distinct records $x, x' \in \mathcal{X}$, there exist $q \in \text{sep}(\mathcal{Q})$ such that $q(x) \neq q(x')$.

Algorithm 3: sepFEM

Input: A dataset D, the $t-1$ queries $\{q_1,\ldots,q_{t-1}\}$, A queryset $\mathcal{Q}$ with small separator set $\text{sep}(\mathcal{Q})$
Parameters: Laplace noise parameters η, The number of samples s
Let $\tilde{q} \leftarrow \text{sep}(\mathcal{Q})$ and $M \leftarrow |\text{sep}(\mathcal{Q})|$;
for $i \leftarrow 1$ **to** s **do**
  Sample noise vector $\sigma \sim \text{Lap}(\eta)^M$;
  Get $x_i \in \arg\max_x \left\{\sum_{j=1}^{t-1} q_j(x) + \sum_{j=1}^{M}\sigma_j\tilde{q}_j(x)\right\}$;
end
Output: $\widehat{D} = \{x_1, x_2, \ldots, x_s\}$

Proof of Theorem 2.2.

Proof. Setting $\eta = (5/2)^{1/2}d^{1/4}T^{-1/2}$, the data player's average regret for all T rounds is $R_D(T) = 4\sqrt{10}d^{5/4}T^{-1/2}$ and the average regret for the query player is $R_Q(T)$ given by lemma 11. If $T = \frac{2\sqrt{5}d^{5/4}n\varepsilon}{\log^{1/2}(1/\delta)\log\left(\frac{2|\mathcal{Q}|}{\beta}\right)}$, then, by union bound and by Theorem 7, with probability at least $1-\beta$, the accuracy of sepFEM is:

$$\begin{aligned}\alpha &= R_D(T) + R_Q(T)\\ &= 4\sqrt{10}d^{5/4}T^{-1/2} + \frac{2\sqrt{2}}{n\varepsilon}\sqrt{T\log(1/\delta)}\ln\left(\frac{2T|\mathcal{Q}|}{\beta}\right)\end{aligned}$$

Plugging in the value of T,

$$\alpha = \frac{8\sqrt[4]{5}d^{5/8}\log^{1/4}(1/\delta)\log^{1/2}\left(\frac{2T|\mathcal{Q}|}{\beta}\right)}{n^{1/2}\varepsilon^{1/2}}$$

□

D DQRS: DualQuery with Rejection Sampling

In this section, we present an improvement from the DualQuery algorithm [11]. In DualQuery , the query player maintains a distribution over queries using Multiplicative Weights. But the algorithm can't directly release the distribution $\mathcal{Q}^t$ proposed by MW during round t because it depends on the private data. Instead, for each round t, it takes s samples from $\mathcal{Q}^t$ to form an estimate distribution $\widehat{\mathcal{Q}^t}$. The data player then best-responds against $\widehat{\mathcal{Q}^t}$. Our algorithm DQRS improves the sampling step of DualQuery . The basic idea is to apply the rejection sampling technique to generate a sample from $\mathcal{Q}^t$ using samples obtained from the distribution in the previous round, i.e., $\mathcal{Q}^{t-1}$. We show that by taking fewer samples from $\mathcal{Q}^t$ for each round t, we use less of the privacy budget. The result is that the algorithm operates for more iterations and obtains better performance.

Theorem 12. DualQuery *with rejection sampling (Algorithm 4) takes in a private dataset $D \in \mathcal{X}^n$ and makes $T = O\left(\frac{\log|\mathcal{Q}|}{\alpha^2}\right)$ queries to an optimization oracle and outputs a dataset $\tilde{D} = (x^1, \cdots, x^T) \in \mathcal{X}^T$ such that, with probability at least $1-\beta$, for all $q \in \mathcal{Q}$ we have $|q(\tilde{D}) - q(D)| \leq \alpha$. The algorithm is ρ-CDP for*

$$\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|)}{n^2\alpha^5}\right).$$

In contrast, DualQuery (*without* rejection sampling) obtains the same result except with

$$\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|)}{n^2\alpha^7}\right).$$

To obtain (ε, δ)-differential privacy, it suffices to have ρ-CDP for $\rho = \Theta(\varepsilon^2/\log(1/\delta)$. Thus the guarantee of Theorem 12 can be rephrased as the sample complexity bound

$$n = O\left(\frac{\log^{1.5}(|\mathcal{Q}|) \cdot \sqrt{\log(|\mathcal{X}|T/\beta) \cdot \log(1/\delta)}}{\alpha^{2.5}\varepsilon}\right)$$

to obtain α-accurate synthetic data with probability $1-\beta$ under (ε, δ)-differential privacy.

Algorithm 4: Rejection Sampling Dualquery

Parameters: Target accuracy $\alpha \in (0,1)$, target failure probability $\beta \in (0,1)$
Input: dataset D, and linear queries $q_1, \ldots, q_k \in \mathcal{Q}$
Initialize: Let $\mathcal{Q} = \bigcup_{i=1}^k q_i \cup \bar{q}_i$, and Q^1 a uniform distribution on $\mathcal{Q}$
Set $T = \frac{16\log|\mathcal{Q}|}{\alpha^2}$, $\eta = \frac{\alpha}{4}$, $\gamma_t = \frac{1}{2t^{2/3}}$, $s = \frac{48\log(3|\mathcal{X}|T/\beta)}{\alpha^2}$, and $\tilde{s}_t = (2\gamma_t + 4\eta)s$.
Construct sample S_1 of s queries $\{q_i\}$ from $\mathcal{Q}$ according to $\mathcal{Q}^1$;
for $t \leftarrow 1$ **to** T **do**
  Let $\tilde{q} = \frac{1}{s}\sum_{q \in S_t} q$;
  Find x^t with $A_D(x^t, \tilde{q}) \geq \max_x A_D(x, \tilde{q}) - \alpha/4$;
  for $q \in \mathcal{Q}$ **do**
    $\hat{\mathcal{Q}}_q^{t+1} := e^{-\eta-\gamma_t} \cdot \exp\left(-\eta A_D(x^t, q)\right) \mathcal{Q}_q^t$;
  end
  Normalize $\hat{\mathcal{Q}}^{t+1}$ to obtain $\mathcal{Q}^{t+1}$;
  Construct S_{t+1} as follows;
  for $q \in S_t$ **do**
    Add q to S_{t+1} with probability $\hat{\mathcal{Q}}^{t+1}/\mathcal{Q}^t$;
    Add $\tilde{s}_t$ independent fresh samples from Q^{t+1} to S_{t+1}.;
    If $|S_{t+1}| > s$, discard elements at random so that $|S_{t+1}| = s$.;
  end
end

Lemma 13. *The subroutine which accepts q with probability $\hat{Q}_q^{t+1}/Q_q^t = e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q))$ is ε-differentially private for $\varepsilon = \max\{\eta/n, \eta/\gamma_t n\}$.*

Proof. Note that $0 < p := \hat{Q}_q^{t+1}/Q_q^t = e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q)) \leq e^{-\gamma_t} < 1$. In particular, the probability is well-defined.

We compute the ratio between the probabilities that q is accepted under executions of the algorithm on neighboring datasets D, D' for fixed choices of the best responses $x^1, \ldots, x^t$. This ratio is given by

$$\frac{p}{p'} = \frac{\hat{Q}_q^{t+1}[D]}{Q_q^t[D]} \cdot \frac{Q_q^t[D']}{\hat{Q}_q^{t+1}[D']} = \frac{\exp(-\eta A_D(x^t, q))}{\exp(-\eta A_{D'}(x^t, q))} \leq e^{\eta/n}.$$

Similarly, we evaluate the ratio of the probabilities that q is *not* accepted under executions of the algorithm on D and D': Since $p' \leq e^{-\gamma_t}$ and $p/p' \geq e^{-\eta/n}$, we have

$$\frac{1-p}{1-p'} = 1 + \frac{1}{1/p' - 1}\left(1 - \frac{p}{p'}\right) \leq 1 + \frac{1 - e^{-\eta/n}}{e^{\gamma_t} - 1} \leq 1 + \frac{\eta/n}{\gamma_t} \leq e^{\eta/\gamma_t n},$$

as required. □

Bad samples also incur privacy loss from sampling from the distribution Q^t. Just as in [11], we use the fact that this step can be viewed as an instantiation of the exponential mechanism with score function $\sum_{i=1}^{t-1}(q(D) - q(x^i))$ to obtain:

Lemma 14. *Sampling from Q^t is ε-differentially private for $\varepsilon = 2\eta(t-1)/n$.*

Proof of Privacy for Theorem 12.

Proof. Each round t incurs privacy loss from s invocations of a $(\eta/\gamma_t n)$-differentially private algorithm (rejection sampling, Lemma 13), and $\tilde{s}_t$ invocations of a $(2\eta(t-1)/n)$-differentially private algorithm (Lemma 14). Since ε-differential privacy implies $\frac{1}{2}\varepsilon^2$-CDP [2], we have (by composition) that round t is ρ_t-CDP for

$$\rho_t = \frac{\eta^2}{2\gamma_t^2 n^2} s + \frac{2\eta^2(t-1)^2}{n^2}\tilde{s}_t = \frac{\eta^2 s}{n^2}\left(\frac{1}{2\gamma_t^2} + 2(t-1)^2 \cdot (2\gamma_t + 4\eta)\right) \leq \frac{\eta^2 s}{n^2}\left(4t^{4/3} + 8\eta t^2\right).$$

Composing over rounds $t = 1 \cdots T$ yields $\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta)\cdot\log^{2+1/3}(|\mathcal{Q}|)}{n^2\alpha^{4+2/3}} + \frac{\log(|\mathcal{X}|T/\beta)\cdot\log^3(|\mathcal{Q}|)}{n^2\alpha^5}\right)$, as required.

□

Accuracy

The accuracy analysis follows that of of DualQuery, together with the following claims showing that the rejection sampling process simulates the collection of independent samples in the DualQuery algorithm.

Lemma 15. *Let P and Q be probability distributions over $\mathcal{Q}$, and let $M \geq \max_{q\in\mathcal{Q}} P_q/Q_q$. Sample an element of $\mathcal{Q}$ as follows. Sample q according to Q, and accept it with probability $P_q/(M \cdot Q_q)$. If q is not accepted, sample q according to P. Then the resulting element is distributed according to P.*

Proof. The total probability of sampling q according to this procedure is given by

$$\begin{aligned} Q_q \cdot \frac{P_q}{M \cdot Q_q} + P_q \cdot \sum_{q'\in\mathcal{Q}} Q_{q'} \cdot \left(1 - \frac{P_{q'}}{M \cdot Q_{q'}}\right) &= P_q \cdot \left(\frac{1}{M} + \sum_{q'\in\mathcal{Q}}\left(Q_{q'} - \frac{P_{q'}}{M}\right)\right) \\ &= P_q \cdot \left(\frac{1}{M} + \left(1 - \frac{1}{M}\right)\right) \\ &= P_q. \end{aligned}$$

□

Lemma 16. *For any given round t, the probability that more than $\tilde{s}_t$ samples are rejected is at most $(e/4)^{\tilde{s}_t} \leq \frac{\beta}{3T}$.*

Proof. The probability that any given sample is rejected is $1 - \hat{Q}_q^{t+1}/Q_q^t = 1 - e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q)) \leq 1 - e^{-2\eta-\gamma_t} \leq 2\eta + \gamma_t = \frac{\tilde{s}_t}{2s}$. (In particular, $\tilde{s}_t$ is at least twice the expected number of rejected samples.) The set of s samples is rejected independently. By a multiplicative Chernoff bound, the probability that more than $\tilde{s}_t$ samples are rejected is at most $(e/4)^{\tilde{s}_t}$. Note that $\tilde{s}_t \geq 4\eta s = \frac{48}{\alpha} \log\left(\frac{3|\mathcal{X}|T}{\beta}\right)$. Thus $(e/4)^{\tilde{s}_t} \leq \left(\frac{\beta}{3|\mathcal{X}|T}\right)^{18/\alpha} \leq \frac{\beta}{3T}$. □

Together Lemmas 15 and 16 show that, with high probability, at each round t, the set S_t is distributed as s independent samples from $\mathcal{Q}^t$. Given this, the rest of the proof follows that of the original DualQuery.

Proof of Accuracy for Theorem 12.

Proof. For each round t, by Hoeffding's bound and Lemma 16 and a union bound over $\mathcal{X}$, with probability at least $1 - \frac{\beta}{T}$, we have

$$\forall x \in \mathcal{X} \qquad \left| \frac{1}{s} \sum_{q \in S_t} q(x) - \mathop{\mathbb{E}}_{q \leftarrow \mathcal{Q}^t} [q(x)] \right| \leq \frac{\alpha}{4}.$$

By a union bound over the T rounds we have that the above holds for all $t \in [T]$ with probability at least $1 - \beta$.

By assumption, in each round t, our oracle returns x^t that is an $\alpha/4$-approximate best response to the uniform distribution over S_t. Thus, with high probability, the sequence $x^1, \cdots, x^T$ are $\alpha/2$-approximate best responses to the distributions $Q^1, \cdots, Q^t$. Since the distributions are generated by multiplicative weights, we have that this is an α-approximate equilibrium. Hence the uniform distribution over $x^1, \cdots, x^T$ is an α-accurate synthetic database for D. □