Iterative Constructions and Private Data Release

Anupam Gupta[*]   Aaron Roth[†]   Jonathan Ullman[‡]

September 6, 2011

Abstract

In this paper we study the problem of approximately releasing the *cut function* of a graph while preserving differential privacy, and give new algorithms (and new analyses of existing algorithms) in both the interactive and non-interactive settings.

Our algorithms in the interactive setting are achieved by revisiting the problem of releasing differentially private, approximate answers to a large number of queries on a database. We show that several algorithms for this problem fall into the same basic framework, and are based on the existence of objects which we call *iterative database construction* algorithms. We give a new generic framework in which new (efficient) IDC algorithms give rise to new (efficient) interactive private query release mechanisms. Our modular analysis simplifies and tightens the analysis of previous algorithms, leading to improved bounds. We then give a new IDC algorithm (and therefore a new private, interactive query release mechanism) based on the Frieze/Kannan low-rank matrix decomposition. This new release mechanism gives an improvement on prior work in a range of parameters where the size of the database is comparable to the size of the data universe (such as releasing all cut queries on dense graphs).

We also give a non-interactive algorithm for efficiently releasing private *synthetic data* for graph cuts with error $O(|V|^{1.5})$. Our algorithm is based on randomized response and a non-private implementation of the SDP-based, constant-factor approximation algorithm for cut-norm due to Alon and Naor. Finally, we give a reduction based on the IDC framework showing that an efficient, private algorithm for computing sufficiently accurate rank-1 matrix approximations would lead to an improved efficient algorithm for releasing private synthetic data for graph cuts. We leave finding such an algorithm as our main open problem.

1   Introduction

Consider a graph representing the online communications between a set of individuals: each vertex represents a user, and an edge between two users indicates that they have corresponded by email. It might be extremely useful to allow data analysts access to this graph in order to mine it for statistical information. However, the graph is also composed of sensitive information, and we cannot allow our released information to reveal much about the existence of specific edges. Thus we would like a way to analyze the structure of this graph while protecting the privacy of individual edges. Specifically we would like to be able to provide a promise of *differential privacy* [DMNS06]

(defined in Section 2), which, roughly, requires that our algorithms be randomized, and induce nearly the same distribution over outcomes when given two data sets (e.g. graphs) which differ in only a single point (e.g. an edge).

One natural objective is to provide private access to the *cut function* of this graph. That is, to provide a privacy preserving way for a data analyst to specify any two (of the exponentially many) subsets of individuals, and to discover (up to some error) the number of email correspondences that have passed between these two groups. There are two ways we might try to achieve this goal: We could give an *interactive* solution where we give the analyst private oracle access to the cut function. Here the user can write down any sequence of cut queries and the oracle will respond with private, approximate answers. We may also try for a stronger, *non-interactive* solution, in which we release a private *synthetic dataset*; a new, private graph that approximately preserves the cut function of the original graph.

The case of answering cut queries on a graph is just one instance of the more general problem of query release for exponentially sized families of *linear queries* on a data set. Although this problem has been extensively studied in the differential privacy literature, we observe that no previously known efficient solution is suitable for the case of releasing all cut queries on graphs. In this paper we provide solutions to this problem in both the interactive and non-interactive settings.

We give a generic framework that converts objects that we call *iterative database construction (IDC)* algorithms into private query release mechanisms in both the interactive and non-interactive settings. This framework generalizes the median mechanism [RR10], the online multiplicative weights mechanism [HR10], and the offline multiplicative weights mechanism [GHRU11, HLM11]. Our framework gives a simple, modular analysis of all of these mechanisms, which lead to tighter bounds in the interactive setting than those given in [RR10] and [HR10]. These improved bounds are crucial to our objective of giving non-trivial approximations to all possible cut queries. We also instantiate this framework with a new IDC algorithm for arbitrary linear queries that is based on the Frieze/Kannan low-rank matrix decomposition [FK99a] and is tailored to releasing cut queries. This algorithm leads to a new online query release mechanism for linear queries that gives a better approximation in settings (such as we would encounter trying to answer all cut queries on a dense graph) where the database size is comparable to the size of the data universe. We summarize our bounds in Table 1.

We also give a new algorithm (building on techniques for constructing private synthetic data in [BCD$^+$07, DNR$^+$09]) in the non-interactive setting that efficiently generates private synthetic graphs that approximately preserve the cut function. Finally, we use our IDC framework to show that an efficient, private algorithm for the problem of privately computing good rank-1 approximations to symmetric matrices would automatically yield efficient private algorithms for releasing synthetic graphs with improved approximation guarantees.

1.1 Our Results and Techniques

Our main conceptual contribution is to define the abstraction of *iterative database construction* algorithms (Section 3) and to show that an efficient IDC for any class of queries $\mathcal{Q}$ automatically yields an efficient private data release mechanism for $\mathcal{Q}$ in both the interactive and non-interactive settings. Informally, IDCs construct a data structure that can be used to answer all the queries in $\mathcal{Q}$ by iteratively improving a hypothesis data structure. Moreover, they update the hypothesis when given a query witnessing a significant difference between the hypothesis data structure and the underlying database.

In hindsight, this framework generalizes the median mechanism [RR10] and the online multi-

	Previous Bounds	This Paper	
		General Bounds	Cut Queries
Median Mechanism[a] [RR10]	$\frac{n^{2/3}(\log k)(\log|\mathcal{X}|)^{1/3}}{\epsilon^{1/3}}$ [RR10]	$\frac{n^{1/2}(\log k)^{3/4}(\log|\mathcal{X}|)^{1/4}}{\epsilon^{1/2}}$	$\frac{|E|^{1/2}|V|^{3/4}(\log|V|)^{1/4}}{\epsilon^{1/2}}$
Online MW [HR10]	$\frac{n^{1/2}(\log k)(\log|\mathcal{X}|)^{1/4}}{\epsilon}$ [HR10]	$\frac{n^{1/2}(\log k)^{1/2}(\log|\mathcal{X}|)^{1/4}}{\epsilon^{1/2}}$	$\frac{|E|^{1/2}|V|^{1/2}(\log|V|)^{1/4}}{\epsilon^{1/2}}$
Frieze/Kannan IDC	New in this paper	$\frac{n_2^{1/4}(\log k)^{1/2}|\mathcal{X}|^{1/4}}{\epsilon^{1/2}}$ [b]	$\frac{|E|^{1/4}|V|}{\epsilon^{1/2}}$
K-Norm Mechanism [HT10]	$\frac{\sqrt{k}}{\epsilon}\left(\log\left(\frac{|\mathcal{X}|}{k}\right)\right)^{1/2}$[HT10][c]	Not in the IDC Framework	Not Applicable

Table 1: Comparison of accuracy bounds for linear queries. The bounds in the first column are prior to this work, the second column are what we achieve in this work, and the last column are the new bounds instantiated for releasing all cut queries. The bounds listed here are approximate and hide the dependence on certain parameters, such as δ and β. n denotes database size, k denotes the total number of queries answered, and $\mathcal{X}$ represents the data universe. For a graph $G = (V, E)$, $n = n_2 = |E|$, $|\mathcal{X}| = \binom{|V|}{2}$, and for all cut queries, $k = 2^{2|V|}$. Previous efficient results do not achieve non-trivial ($\leq |E|$) error, while all of the new bounds do for sufficiently dense graphs.

---

[a]The bounds listed here are for linear queries. The Median Mechanism more generally works for any set of low sensitivity queries $\mathcal{Q}$ that have an α-net of size $N_\alpha(\mathcal{Q})$. We improve the bound from the solution to $\alpha = \frac{\log(N_\alpha(\mathcal{Q}))\log^2(\mathcal{Q})}{\epsilon}$ to the solution to $\alpha = \frac{\sqrt{\log N_\alpha(\mathcal{Q})}\log k}{\epsilon}$.

[b]Here we use $n_2 = \|\mathcal{D}\|_2^2$, in contrast to other known IDCs, whose error is in terms of $n = \|\mathcal{D}\|_1$. Note that $n \leq n_2 \leq n^2$.

[c]For $k \leq |\mathcal{X}|/2$. This is an approximate bound on *average* per-query error. All other algorithms listed bound worst-case per-query error.

plicative weights mechanism [HR10]. It also generalizes the offline multiplicative weights mechanism [GHRU11, HLM11]. All of these mechanisms can be seen to use IDCs of the sort we define in this work. (In Appendix A we show how these algorithms fall into the IDC framework.)

Our generalization and abstraction also allows for a simple, modular analysis of mechanisms based on IDCs. Using this analysis, we are able to show improved bounds on the accuracy of both the median mechanism and multiplicative weights mechanism. These improved bounds are critical to our application to releasing all cut queries. For these parameters, the previous bounds would not guarantee error that is $\leq |E|$, meaning that the error may be larger than the largest cut in the graph. Of course, we can privately guarantee error $\leq |E|$ simply by releasing the answer 0 for every cut query. Our new analysis shows that these mechanism are capable of answering all cut queries with error $o(|E|)$ for sufficiently dense graphs.

We also define a new IDC based on the Frieze/Kannan low-rank matrix decomposition [FK99a], which yields a private interactive mechanism for releasing linear queries. Our new mechanism outperforms previously known techniques when the size of the database is comparable to the size of the data universe, as is the case on a dense graph.

We then consider the problem of efficiently releasing private synthetic data for the class of cut queries. We show that a technique based on randomized response efficiently yields a private data structure (but not a synthetic databse) capable of answering any cut query on a graph with $|V|$ vertices up to maximum error $O(|V|^{1.5})$. We then show how to use this data structure to efficiently construct a synthetic database with only a constant factor blowup in our error. Our algorithm is based on a technique for constructing synthetic data in [BCD⁺07, DNR⁺09]. Their observation is that, for linear queries, the set of accurate synthetic databases is described by a (large) set of linear constraints. In the case of cut queries, we are able to use a constant-factor approximation to

the cut-norm due to Alon and Naor [AN06] as the separation oracle to find a feasible solution (and thus a synthetic database) efficiently. Finally, we show how the existence of an efficient private algorithm for finding good low-rank approximations to matrices would imply the existence of an improved algorithm for privately releasing synthetic data for cut queries, using our IDC framework.

1.2 Related Work

Differential privacy was introduced in a series of papers [BDMN05, CDM$^+$05, DMNS06] in the last decade, and has become a standard solution concept for statistical database privacy. The first mechanism for simultaneously releasing the answers to exponentially large classes of statistical queries was given in [BLR08]. They showed that the existence of small nets for a class of queries $\mathcal{Q}$ automatically yields a (computationally inefficient) non-interactive, private algorithm for releasing answers to all the queries in $\mathcal{Q}$ with low error. Subsequent improvements were given by Dwork et al. [DNR$^+$09, DRV10].

Roth and Roughgarden [RR10] showed that large classes of queries could also be released with low error in the *interactive* setting, in which queries may arrive online, and the mechanism must provide answers before knowing which queries will arrive in the future. Subsequently, Hardt and Rothblum [HR10] gave improved bounds for the online query release problem based on the multiplicative weights algorithm. In hindsight, both of these algorithms follow the same basic framework, which is to use an IDC.

Gupta et al. [GHRU11] gave a *non-interactive* data release mechanism based on the multiplicative weights algorithm and an arbitrary agnostic learner for a class of queries. An instantiation of this algorithm (the *offline* multiplicative weights algorithm) using the generic agnostic learner of Kasiviswanathan et al. [KLN$^+$08] (who use the exponential mechanism of [MT07]) was implemented and experimentally evaluated on the task of releasing small conjunctions to low error on real data by Hardt, Ligett, and McSherry [HLM11]. This algorithm gives bounds comparable to those given in this paper, but it does not work in the interactive setting, and is not computationally efficient for settings in which the number of queries is exponentially larger than the database size (as is the case with graph cuts). We note in Section 7 that this generic algorithm can also be instantiated with any iterative database construction algorithm.

Hardt and Talwar [HT10] consider the setting where the number of queries is smaller than the universe size. When the number of queries is comparable to the universe size (i.e. $|\mathcal{Q}| = \Omega(|\mathcal{X}|)$), their K-Norm mechanism gives *average error* that is smaller than the worst-case error promised by the online multiplicative weights mechanism when the database size is $n \geq \tilde{O}\left(\frac{|\mathcal{X}|}{\lceil \log |\mathcal{X}| \rceil}\right)$. This is the same range of parameters for which the Frieze/Kannan IDC algorithm improves on the online-multiplicative weights, and in this range of parameters, it achieves roughly the same error as the K-norm mechanism. In general, the bounds for the two mechanisms are incomparable: e.g., [HT10] have a better, logarithmic dependence on $|\mathcal{X}|$, compared to the polynomial dependence for the Frieze/Kannan IDC. On the other hand, the Frieze/Kannan IDC (and all algorithms in the IDC framework) have some advantages. Specifically, the bounds are for worst-case error, rather than average-case error; hold unconditionally, while the accuracy of the K-norm mechanism relies on the truth of the hyperplane conjecture; apply even when the number of queries is larger than the universe size; and typically have running time linear in $|\mathcal{X}|$, rather than poly($|\mathcal{X}|$).

The Frieze-Kannan low-rank approximation (or the weak regularity lemma) shows that every matrix can be approximated by a sum of few cut matrices [FK99a, FK99b]: this fact has many important algorithmic applications. We also use the fact that the proof extends to more general settings, as was noted by [TTV09].

2 Preliminaries

In this paper, we study datasets $\mathcal{D}$ that consist of collections of n elements from some universe $\mathcal{X}$. We can also write $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ when it is convenient to represent $\mathcal{D}$ as a histogram over $\mathcal{X}$. We say that two databases $\mathcal{D}, \mathcal{D}'$ are adjacent if they differ in only a single element. As histograms, they are adjacent if $\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1$. We will require that our algorithms satisfy *differential privacy*:

Definition 2.1 (Differential Privacy). A randomized algorithm $M : \mathbb{N}^{|\mathcal{X}|} \to R$ (for any abstract range R) satisfies (ϵ, δ)-differential privacy if for all adjacent databases $\mathcal{D}$ and $\mathcal{D}'$, and for all events $S \subseteq R$:
$$\Pr[M(\mathcal{D}) \in S] \leq \exp(\epsilon)\Pr[M(\mathcal{D}') \in S] + \delta$$

We will generally think of ϵ as being a small constant, and δ as being negligibly small – i.e. smaller than any inverse polynomial function of n.

We note that when we will discuss interactive mechanisms, we must view the output of a mechanism as a *transcript* of an interaction between an adaptive adversary who supplies questions about the database based on previous outcomes of the mechanism, and the mechanism itself. For clarity, in this paper we will elide specifics about the model of adaptive private composition. For a detailed treatment of this issue, see [DRV10].

A useful distribution is the *Laplace* distribution.

Definition 2.2 (The Laplace Distribution). The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function: $\mathrm{Lap}(x|b) = \frac{1}{2b}\exp(-\frac{|x|}{b})$. We will sometimes write $\mathrm{Lap}(b)$ to denote the Laplace distribution with scale b, and will sometimes abuse notation and write $\mathrm{Lap}(b)$ simply to denote a random variable $X \sim \mathrm{Lap}(b)$.

A fundamental result in data privacy is that perturbing low sensitivity queries with Laplace noise preserves $(\epsilon, 0)$-differential privacy.

Theorem 2.3 ([DMNS06]). *Suppose $Q : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is a function such that for all adjacent databases $\mathcal{D}$ and $\mathcal{D}'$, $\|Q(\mathcal{D}) - Q(\mathcal{D}')\|_1 \leq 1$. Then the procedure which on input $\mathcal{D}$ releases $Q(\mathcal{D}) + (X_1, \ldots, X_k)$, where each X_i is an independent draw from a $\mathrm{Lap}(1/\epsilon)$ distribution, preserves $(\epsilon, 0)$-differential privacy.*

It will be useful to understand how privacy parameters for individual steps of an algorithm compose into privacy guarantees for the entire algorithm. The following useful theorem is due to Dwork, Rothblum, and Vadhan:

Theorem 2.4 ([DRV10]). *Let $0 \leq \epsilon \leq 1$ be a parameter. Let P, Q be probability measures supported on a set $\mathcal{S}$ such that $\max_{s \in S} |\log(P(s)/Q(s))| \leq \epsilon$. Then $\mathbb{E}_P[\log(P(s)/Q(s))] \leq 2\epsilon^2$.*

We are interested in privately releasing accurate answers to large collections of queries. Queries are functions $Q : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}$, and we denote collections of queries by $\mathcal{Q}$. We write $k = |\mathcal{Q}|$ to denote the cardinality of the set of queries.

A common type of queries are *linear queries*. A linear query Q has a representation as a vector $[0, 1]^{|\mathcal{X}|}$, and can be evaluated on a database by taking the dot product between the query and the histogram representation of the database: $Q(\mathcal{D}) = Q \cdot \mathcal{D}$.

Definition 2.5 (Accuracy). Let $\mathcal{Q}$ be a set of queries. A mechanism $M : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}$ is (α, β)-*accurate for* $\mathcal{Q}$ if there exists a function $\mathrm{Eval} : \mathcal{Q} \times \mathcal{R} \to \mathbb{R}$ s.t. for every database $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$, with probability at least $1 - \beta$ over the coins of M, $M(\mathcal{D})$ outputs $r \in \mathcal{R}$ such that $\max_{Q \in \mathcal{Q}} |Q(\mathcal{D}) - \mathrm{Eval}(Q, r)| \leq \alpha$. We will abuse notation and write $Q(r) = \mathrm{Eval}(Q, r)$.

We say that an algorithm *M releases synthetic data* (as is the case for our new IDC, as well as the multiplicative weights IDC [HR10]) if $\mathcal{R} = \mathbb{N}^{|\mathcal{X}|}$ In this case, $M(\mathcal{D}) = \mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}$ and $\text{Eval}(\mathcal{D}', Q) = Q(\mathcal{D}')$. We say that a synthetic data release algorithm is *efficient* if it runs in time polynomial in $n = \|\mathcal{D}\|_1$, the size of the data set. Note that if $n \ll |\mathcal{X}|$, efficient algorithms will have to input and output concise representations of the dataset (i.e., as collections of items from the universe) instead of using the histogram representation. Nevertheless, it will be convenient to think of datasets as histograms.

We say an algorithm efficiently releases k queries from a class $\mathcal{Q}$ in the interactive setting if on an arbitrary, adaptively chosen stream of queries $Q_1, \ldots, Q_k$, it outputs answers $a_1, \ldots, a_k$. The algorithm must output each a_i after receiving query Q_i but before receiving Q_{i+1}, and is only allowed $\text{poly}(n)$ run time per query. We are typically interested in the case when k can be exponentially large in n. Note that as far as computational efficiency is concerned, releasing synthetic data for a class of queries k is at least as difficult as releasing queries from k in the interactive setting, since we can use the synthetic data to answer queries interactively.

Graphs and Cuts. When we consider datasets that represent graphs $G = (V, E)$, we think of the database as being the edge set $\mathcal{D}_G = E$, and the data-universe being the collection of all possible edges in the complete graph: $|\mathcal{X}| = \binom{|V|}{2}$. That is, we consider the vertex set to be common among all graphs, which differ only in their edge sets. One example we care about is approximating the cut function of a private graph G.

For any real-valued matrix $A \in \mathbb{R}^{m \times m'}$, for $S \subseteq [m]$ and $T \subseteq [m']$, we define $A(S, T) := \sum_{s \in S, t \in T} A_{st}$. The *cut norm of the matrix A* is now defined as $\|A\|_C := \max_{S \subseteq [m], T \subseteq [m']} |A(S, T)|$. A graph G can be represented as its adjacency matrix $A_G \in \{0, 1\}^{|V| \times |V|}$. In this paper, a *cut in a graph G* is defined by any two subsets of vertices $S, T \subseteq V$. We write the value of an S, T cut in G as $G(S, T) := A_G(S, T)$, where A_G is the adjacency matrix of G. Similarly, we extend the definition of *cut norm* to n vertex graphs naturally by defining $\|G\|_C := \|A_G\|_C = \max_{S, T \subseteq V} |G(S, T)|$ and $\|G - H\|_C := \|A_G - A_H\|_C$. The class of cut queries $\mathcal{Q}_{\text{Cut}} = \{Q_{S,T} : S, T \subseteq V\}$, where $Q_{S,T}(G) = A_G(S, T)$. Note that cut queries are an example of a class of linear queries, because we can represent them as a vector in which $Q_{S,T}[i, j] = 1$ if $i \in S, j \in T$ and 0 otherwise, and evaluate $Q_{S,T}(G) = \sum_{i,j \in V} Q_{S,T}[i, j] \cdot A_G[i, j]$.

Note that as linear queries, we can write cut queries as the outer product of two vectors: $Q_{S,T} = \chi_S \cdot \chi_T^T$, where $\chi_S, \chi_T \in \{0, 1\}^{|V|}$ are the characteristic vectors of the sets S and T respectively. Let us define a more general class of *rank-1* queries on graphs to be a subset of all linear queries:

$$\mathcal{Q}_{r1} = \{Q \in [0, 1]^{|V| \times |V|} \text{ such that } Q = u \cdot v^T \text{ for some vectors } u, v \in [0, 1]^{|V|}\}$$

A rank-1 query is a linear query and can be evaluated

$$Q_{u,v}(G) = \sum_{i,j \in V} Q[i, j] A_G[i, j] = \sum_{i,j \in V} u[i] v[j] A_G[i, j]$$

Of course the set of rank-1 queries includes the set of cut queries, and any mechanism that is accurate with respect to rank-1 queries is also accurate with respect to cut queries.

3 Iterative Database Constructions

In this section we define the abstraction of *iterative database constructions* that includes our new Frieze/Kannan construction and several existing algorithm [RR10, HR10] as a special case. Roughly,

each of these mechanisms works by maintaining a sequence of data structures $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots$ that give increasingly good approximations to the input database $\mathcal{D}$ (in a sense that depends on the IDC). Moreover, these mechanisms produce the next data structure in the sequence by considering only one query Q that *distinguishes* the real database in the sense that $Q(\mathcal{D}^{(t)})$ differs significantly from $Q(\mathcal{D})$.

Syntactically, we will consider functions of the form $\mathbf{U} : \mathcal{R}_{\mathbf{U}} \times \mathcal{Q} \times \mathbb{R} \to \mathcal{R}_{\mathbf{U}}$. The inputs to $\mathbf{U}$ are a data structure in $\mathcal{R}_{\mathbf{U}}$, which represents the current data structure $\mathcal{D}^{(t)}$; a query Q, which represents the distinguishing query, and may be restricted to a certain set $\mathcal{Q}$; and also a real number. which estimates $Q(\mathcal{D})$. Formally, we define a *database update sequence*, to capture the sequence of inputs to $\mathbf{U}$ used to generate the database sequence $\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots$.

Definition 3.1 (Database Update Sequence). Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be any database and let $\left\{ (\mathcal{D}^{(t)}, Q^{(t)}, \widehat{A}^{(t)}) \right\}_{t=1,\ldots,C} \in (\mathcal{R}_{\mathbf{U}} \times \mathcal{Q} \times \mathbb{R})^C$ be a sequence of tuples. We say the sequence is an $(\mathbf{U}, \mathcal{D}, \mathcal{Q}, \alpha, C)$-*database update sequence* if it satisfies the following properties:

1. $\mathcal{D}^{(1)} = \mathcal{D}(\emptyset, \cdot, \cdot)$,
2. for every $t = 1, 2, \ldots, C$, $\left| Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^{(t)}) \right| \geq \alpha$,
3. for every $t = 1, 2, \ldots, C$, $\left| Q^{(t)}(\mathcal{D}) - \widehat{A}^{(t)} \right| < \alpha$,
4. and for every $t = 1, 2, \ldots, C - 1$, $\mathcal{D}^{(t+1)} = \mathbf{U}(\mathcal{D}^{(t)}, Q^{(t)}, \widehat{A}^{(t)})$.

We note that for all of the iterative database constructions we consider, the approximate answer $\widehat{A}^{(t)}$ is used only to determine the *sign* of $Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^{(t)})$, which is the motivation for requiring that $\widehat{A}^{(t)}$ have error smaller than α. The main measure of efficiency we're interested in from an iterative database construction is the maximum number of updates we need to perform before the database $\mathcal{D}^{(t)}$ approximates $\mathcal{D}$ well with respect to the queries in $\mathcal{Q}$. To this end we define an iterative database construction as follows:

Definition 3.2 (Iterative Database Construction). Let $\mathbf{U} : \mathcal{R}_{\mathbf{U}} \times \mathcal{Q} \times \mathbb{R} \to \mathcal{R}_{\mathbf{U}}$ be an update rule and let $B : \mathbb{R} \to \mathbb{R}$ be a function. We say $\mathbf{U}$ is a $B(\alpha)$-*iterative database construction for query class* $\mathcal{Q}$ if for every database $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$, every $(\mathbf{U}, \mathcal{D}, \mathcal{Q}, \alpha, C)$-database update sequence satisfies $C \leq B(\alpha)$.

Note that the definition of an $B(\alpha)$-iterative database construction implies that if $\mathbf{U}$ is a $B(\alpha)$-iterative database construction, then given any maximal $(\mathbf{U}, \mathcal{D}, \mathcal{Q}, \alpha, C)$-database update sequence, the final database $\mathcal{D}^{(C)}$ must satisfy $\max_{Q \in \mathcal{Q}} \left| Q(\mathcal{D}) - Q(\mathcal{D}^{(C)}) \right| \leq \alpha$ or else there would exist another query satisfying property 2 of Definition 3.1, and thus there would exist a $(\mathbf{U}, \mathcal{D}, \mathcal{Q}, \alpha, C+1)$-database update sequence, contradicting maximality.

4 Query Release from Iterative Database Construction

In this section we describe an interactive algorithm for releasing linear queries using an arbitrary iterative database construction.

4.1 Privacy Analysis

Theorem 4.1. *Algorithm 1 is (ϵ, δ)-differentially private.*

Algorithm 1 Online Query Release Mechanism

$\mathcal{M}^{\mathbf{U}}(\mathcal{D}, \epsilon, \delta, \alpha, \beta, k)$:

Input: A database $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$, a parameter $\alpha \in \mathbb{R}$, parameters $\epsilon, \delta, \beta \in [0,1]$, and the number of queries $k \in \mathbb{N}$. Oracle access to $\mathbf{U}$, a $B = B(\alpha)$-iterative database construction for $\mathcal{Q}$.

Parameters:

$$\sigma = \sigma(\alpha) := \frac{1000\sqrt{B(\alpha)} \cdot \log(4/\delta)}{\epsilon} \qquad T = T(\alpha) := 4\sigma(\alpha) \cdot \log(2k/\beta).$$

Set $\mathcal{D}^{(1)} := \mathbf{U}(\emptyset, \cdot, \cdot)$, $C = 0$.

For: $t = 1, 2, \ldots, k$

   1. Receive a query $Q^{(t)} \in \mathcal{Q}$ and compute

$$Z^{(t)} \sim \mathrm{Lap}(\sigma) \qquad A^{(t)} = Q^{(t)}(\mathcal{D}) \qquad \widehat{A}^{(t)} = Q^{(t)}(\mathcal{D}) + Z^{(t)} \qquad \Lambda^{(t)} = Q^{(t)}(\mathcal{D}^{(t)})$$

   2. **If:** $|\widehat{A}^{(t)} - \Lambda^{(t)}| \leq T$ **then:** output $\Lambda^{(t)}$ and set $\mathcal{D}^{(t+1)} = \mathcal{D}^{(t)}$

      Else: output $\widehat{A}^{(t)}$, set $\mathcal{D}^{(t+1)} = \mathbf{U}\left(\mathcal{D}^{(t)}, Q^{(t)}, \widehat{A}^{(t)}\right)$, and set $C = C + 1$.

   3. **If:** $C = B(\alpha)$ **then:** terminate.

Proof. Our privacy analysis follows the approach of [HR10]. Intuitively, we will consider each round of the mechanism individually, conditioned on the previous rounds and classify each round by the amount of "information leaked" from the database. We will use this classification, as well as Azuma's Inequality to bound the total amount of information leaked.

Consider the vector

$$\mathbf{v} = \left\{\mathbf{v}^{(t)}\right\} = \begin{cases} \widehat{A}^{(t)} & \text{if } t \text{ was an update round,} \\ \perp & \text{otherwise.} \end{cases}$$

Observe that $\mathbf{v}$ and the list of queries $Q^{(1)}, \ldots Q^{(k)}$[1] are sufficient to reconstruct the internal state of the mechanism, and thus its output, in each round. Therefore it will be sufficient to demonstrate that a mechanism that releases $\mathbf{v}$ is (ϵ, δ)-differentially private.

Fix any two adjacent databases $\mathcal{D}_1$ and $\mathcal{D}_2$, and let V_1 and V_2 denote the distributions on the vectors $\mathbf{v}$ when $\mathcal{D}_1$ and, $\mathcal{D}_2$ are the input database, respectively. Also fix a vector $\mathbf{v} \in (\mathbb{R} \cup \perp)^k$. We will use $\mathbf{v}^{(<t)}$ to denote the first t entries of the vector $\mathbf{v}$. We will analyze the following *privacy loss function* for each possible output vector $\mathbf{v}$

$$\Psi(\mathbf{v}) = \log\left(\frac{V_1(\mathbf{v})}{V_2(\mathbf{v})}\right) = \sum_{t=1}^{k} \log\left(\frac{V_1(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}\right)$$

In each round $t = 1, 2, \ldots, k$, we define three ranges for the value of the *noise* $Z^{(t)}$ that will describe whether or not we were "never", "sometimes", or "always" going to do an update in round t. Specifically, let $R^{(t)} = Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^{(t-1)})$. Note that $R^{(t)} = A^{(t)} - \Lambda^{(t)}$ and that

---

[1]We treat all the parameters of the mechanism, $\alpha, \beta, \epsilon, \delta, k$ as well as the query sequence $Q^{(1)}, \ldots, Q^{(k)}$ as public information.

$R^{(t)} + Z^{(t)} = \widehat{A}^{(t)} - \Lambda^{(t)}$. Now let

$$E_1^{(t)} = (-T - R^{(t)} + \sigma, T - R^{(t)} - \sigma)$$
$$E_2^{(t)} = [-T - R^{(t)} - \sigma, -T - R^{(t)} + \sigma] \cup [T - R^{(t)} - \sigma, T - R^{(t)} + \sigma]$$
$$E_3^{(t)} = (-\infty, -T - R^{(t)} - \sigma) \cup (T - R^{(t)} + \sigma, \infty)$$

Intuitively, the event $E_1^{(t)}$ corresponds to values of the noise where $\widehat{A}^{(t)} - \Lambda^{(t)}$ is sufficiently small that switching databases could not cause an update. In these rounds, $\mathbf{v}^{(t)} = \perp$ with probability 1 under both V_1 and V_2, so there is no privacy loss. The event $E_3^{(t)}$ corresponds to values of the noise where $\widehat{A}^{(t)} - \Lambda^{(t)}$ is sufficiently large that switching databases could not prevent an update. These rounds do leak information about the database, but the update will increment C, and thus there can only be $B(\alpha)$ such rounds. The event $E_2^{(t)}$ are the problematic rounds. In these rounds we may not update and increment C, thus in principle there may be an arbitrary number of these rounds. However, $\widehat{A}^{(t)} - \Lambda^{(t)}$ may be close enough to the update threshold that switching from $\mathcal{D}_1$ to $\mathcal{D}_2$ would cause an update. Thus these rounds may incur privacy loss. The remainder of the analysis relies on showing that there are not too many such rounds.

Now we make the following claims about the privacy loss in each type of round, based on the properties of the Laplace distribution and the way in which we defined the events $E_1^{(t)}, E_2^{(t)}, E_3^{(t)}$.

Claim 4.2. *For every $u \in \mathbb{R} \cup \perp$ and every $t = 1, 2, \ldots, k$*

$$\log \left(\frac{V_1 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \in E_1^{(t)}, \mathbf{v}^{(<t)} \right)}{V_2 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \in E_1^{(t)}, \mathbf{v}^{(<t)} \right)} \right) = 0$$

Proof. Note that under both conditional measures, the probability of $\mathbf{v}^{(t)} = \perp$ is 1. □

Claim 4.3. *For every $u \in \mathbb{R} \cup \perp$ and every $t = 1, 2, \ldots, k$*

$$\left| \log \left(\frac{V_1 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)}{V_2 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)} \right) \right| \leq 11\epsilon_0 = \frac{11\epsilon}{1000 \cdot \sqrt{B} \cdot \log(4/\delta)}$$

The proof of this claim requires a straightforward analysis of the event $u = \perp$ under both conditional measures. To not interrupt the flow of the larger proof, we defer the details until later. The next claim states that the expected privacy loss is considerably smaller than the worst-case privacy loss.

Claim 4.4. *For every $u \in \mathbb{R} \cup \perp$ and every $t = 1, 2, \ldots, k$*

$$\mathbb{E} \left[\log \left(\frac{V_1 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)}{V_2 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)} \right) \right] \leq 242\epsilon_0^2 = \frac{121\epsilon^2}{500000 \cdot B \cdot \log^2(4/\delta)}$$

as long as $11\epsilon_0 \leq 1$.

Proof. This claim follows from Claim 4.3 and Theorem 2.4. □

In light of the previous claims, we want to bound the number of rounds in which $E_1^{(t)}$ does not occur. Let $H = \left| \left\{ t \mid Z^{(t)} \notin E_1^{(t)} \right\} \right|$.

9

Claim 4.5. *For every $t = 1, 2, \ldots, k$*

$$\Pr\left[Z^{(t)} \in E_3^{(t)} \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)}\right] \geq 1/8.$$

Proof.

$$\Pr\left[Z^{(t)} \in E_3^{(t)} \mid Z^{(t)} \in E_3^{(t)} \cup E_2^{(t)}, \mathbf{v}^{(<t)}\right] = \frac{\Pr\left[Z^{(t)} \in E_3^{(t)}, \mathbf{v}^{(<t)}\right]}{\Pr\left[Z^{(t)} \in E_3^{(t)} \cup E_2^{(t)}, \mathbf{v}^{(<t)}\right]}$$

$$= \frac{\int_{T-R(t)+\sigma}^{\infty} \exp(-z/\sigma)dz}{\int_{T-R(t)-\sigma}^{\infty} \exp(-z/\sigma)dz} = \frac{\exp(-(T-R^{(t)}+\sigma)/\sigma)}{\exp(-(T-R^{(t)}-\sigma)/\sigma)}$$

$$= \exp(-2) \geq 1/8$$

$\square$

Claim 4.6. *With probability $1 - \delta/2$, $|H| \leq 16B\log(4/\delta)$.*

Proof. Claim 4.5 implies that $\mathbb{E}[|H|] \leq 8B$. Note that conditioned on the events of the previous rounds, the events $Z^{(t)} \in E_3^{(t)}$ and $Z^{(t)} \in E_2^{(t)} \cup E_3^{(t)}$ depend only on the coin tosses used to generate $Z^{(t)}$, which are independent of all of the other rounds. Thus we can show that the random variable $|H|$ is dominated by a related random variable in which we do the following: In every round t with $Z^{(t)} \in E_2^{(t)} \cup E_3^{(t)}$, flip a coin $c^{(t)}$ such that $\Pr\left[c^{(t)} = 1\right] = 1/8$. Let H' be defined identically to H but in the process where we terminate the algorithm only when $\sum_{t=1}^{r} c^{(t)} = B$, rather than the actual termination condition $C = B$. Since, by Lemma 4.5, we know that the probability $Z^{(t)} \in E_3^{(t)}$ conditioned on $Z^{(t)} \in E_2^{(t)} \cup E_3^{(t)}$ is at least $1/8$, we can couple these processes to ensure that $c^{(t)} = 1 \implies Z^{(t)} \in E_3^{(t)}$. Thus, our new process will terminate no sooner than the actual algorithm for every choice of random coins, and $|H'|$ dominates $|H|$ in CDF.

Now it suffices to show that $|H'| \leq 2\mathbb{E}[|H'|]\log(4/\delta) \leq 16B\log(4/\delta)$ with probability at least $1 - \delta/2$. By a Chernoff bound[2], the probability that $\sum_{t \in H'} c^{(t)} \geq (1/16\log(4/\delta))|H'|$ is at least $1 - \delta/2$. Thus with probability at least $1 - \delta/2$ we have $|H'| \leq 16B\log(4/\delta)$. $\square$

We now give a high-probability bound on the total privacy loss, conditioned on the event that $|H| \leq 16B \cdot \log(4/\delta)$.

Claim 4.7. *If $|H| \leq 16B \cdot \log(4/\delta)$ then*

$$\Pr\left[|\Psi(\mathbf{v})| > \epsilon\right] \leq \delta/2.$$

---

[2]A form of the Chernoff bound states that for independent $\{0,1\}$-random variables $X_1, \ldots, X_n$, with $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$, $\Pr\left[X < (1-\gamma)\mu\right] < \exp(-\mu\gamma^2/2)$. From this we deduce that for $\mu \geq 2\log(4/\gamma)$, $\Pr\left[X < \mu/\log(1/\gamma)\right] < \gamma$.

Proof. The expected total privacy loss is

$$\mathbb{E}\left[\Psi(\mathbf{v})\right] = \mathbb{E}\left[\sum_{t=1}^{k} \log\left(\frac{V_1(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}\right)\right] = \sum_{t=1}^{k}\mathbb{E}\left[\log\left(\frac{V_1(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}\right)\right]$$

$$\leq \sum_{t=1}^{k}\Pr\left[E_1^{(t)} \mid \mathbf{v}^{(<t)}\right] \cdot \mathbb{E}\left[\log\left(\frac{V_1(\mathbf{v}^{(t)}|E_1^{(t)}, \mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|E_1^{(t)}, \mathbf{v}^{(<t)})}\right)\right]$$

$$+ \Pr\left[\neg E_1^{(t)} \mid \mathbf{v}^{(<t)}\right] \cdot \mathbb{E}\left[\log\left(\frac{V_1(\mathbf{v}^{(t)}|\neg E_1^{(t)}, \mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|\neg E_1^{(t)}, \mathbf{v}^{(<t)})}\right)\right] \tag{1}$$

$$\leq \sum_{t=1}^{k}\Pr\left[\neg E_1^{(t)} \mid \mathbf{v}^{(<t)}\right] \cdot 242\epsilon_0^2$$

$$\leq 1936 B\epsilon_0^2 = \frac{191\epsilon^2}{62500 \cdot \log^2(4/\delta)} \leq \frac{\epsilon}{2}$$

where (1) follows from the convexity of relative entropy, and the final inequality follows from Claim 4.5 and the fact that $\mathbb{E}[|H|] \leq 8$.

Conditioning on the coins of the mechanism we have

$$\Psi(\mathbf{v}) = \sum_{t \in H} \log\left(\frac{V_1(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}{V_2(\mathbf{v}^{(t)}|\mathbf{v}^{(<t)})}\right)$$

and, by Claim 4.3, each term in the sum is at most ϵ_0 in absolute value. Thus we can apply Azuma's Inequality[3] to $\Psi(\mathbf{v})$ to show

$$\Pr\left[|\Psi(\mathbf{v})| > \epsilon\right] \leq \Pr\left[|\Psi(\mathbf{v}) - \mathbb{E}\left[\Psi(\mathbf{v})\right]| > \epsilon/2\right]$$

$$\leq 2\exp\left(-\frac{\epsilon^2}{2|H|\epsilon_0^2}\right)$$

If we condition on the event that $|H| \leq 16B\log(4/\delta)$ then we have

$$\Pr\left[|\Psi(\mathbf{v})| > \epsilon\right] \leq 2\exp\left(-\log(4/\delta)\right) \leq \delta/2,$$

which proves the claim. □

Claims 4.6 and 4.7 suffice to prove the Theorem. □

We now give a proof of Claim 4.3.

Claim 4.8 (Claim 4.3, restated)**.** *For every $u \in \mathbb{R} \cup \perp$ and every $t = 1, 2, \ldots, k$*

$$\left|\log\left(\frac{V_1\left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)}\right)}{V_2\left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)}\right)}\right)\right| \leq 11\epsilon_0 = \frac{11\epsilon}{1000 \cdot \sqrt{B} \cdot \log(4/\delta)}$$

---

[3]Azuma's Inequality states that for a sequence of random variables $X_0, X_1, \ldots, X_n$, s.t. $|X_i - X_{i-1}| \leq \eta$ for $i = 1, 2, \ldots, n$, $\Pr\left[|X_n - X_0| > \gamma\right] \leq 2\exp(-\gamma^2/n\eta^2)$

Proof of Claim 4.3. First we will bound the left-hand-side in the case of $u \in \mathbb{R}$.

$$\left| \log \left(\frac{V_1 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)}{V_2 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)} \right) \right| = \left| \log \left(\frac{\exp(-|u - Q^{(t)}(\mathcal{D}_1)|/\sigma)}{\exp(-|u - Q^{(t)}(\mathcal{D}_2)|/\sigma)} \right) \right|$$

$$\leq \log \left(\exp(1/\sigma) \right) = 1/\sigma \qquad (2)$$

$$= \epsilon_0$$

where inequality (2) follows because the sensitivity of $Q^{(t)}$ is bounded above by 1. Now we will consider the case of $u = \perp$.

$$\left| \log \left(\frac{V_1 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)}{V_2 \left(\mathbf{v}^{(t)} = u \mid Z^{(t)} \notin E_1^{(t)}, \mathbf{v}^{(<t)} \right)} \right) \right|$$

$$= \left| \log \left(\frac{\int_{-R(t)-T}^{-R(t)-T+\sigma} \exp(-|u|/\sigma)du + \int_{-R(t)+T-\sigma}^{-R(t)+T} \exp(-|u|/\sigma)du}{\int_{-R(t)-T+1}^{-R(t)-T+\sigma} \exp(-|u|/\sigma)du + \int_{-R(t)+T-\sigma}^{-R(t)+T-1} \exp(-|u|/\sigma)du} \right) \right|$$

$$\leq \left| \log \left(\frac{\int_{-R(t)-T}^{-R(t)-T+\sigma} \exp(-|u|/\sigma)du}{\int_{-R(t)-T+1}^{-R(t)-T+\sigma} \exp(-|u|/\sigma)du} + \frac{\int_{-R(t)+T-\sigma}^{-R(t)+T} \exp(-|u|/\sigma)du}{\int_{-R(t)+T-\sigma}^{-R(t)+T-1} \exp(-|u|/\sigma)du} \right) \right|$$

$$= \left| \log \left(2 + \frac{\int_{-R(t)-T}^{-R(t)-T+1} \exp(-|u|/\sigma)du}{\int_{-R(t)-T+1}^{-R(t)-T+\sigma} \exp(-|u|/\sigma)du} + \frac{\int_{-R(t)+T-1}^{-R(t)+T} \exp(-|u|/\sigma)du}{\int_{-R(t)+T-\sigma}^{-R(t)+T-1} \exp(-|u|/\sigma)du} \right) \right|$$

$$\leq \left| \log \left(2 \left(1 + \frac{e}{\sigma - 1} \right) \right) \right|$$

$$\leq \frac{2e}{\sigma - 1} \leq \frac{4e}{\sigma} \leq 11\epsilon_0$$

$\square$

4.2 Utility Analysis

Theorem 4.9. *Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be any database. And $\mathbf{U}$ be a $B(\alpha)$-iterative database construction for query class $\mathcal{Q}$. Then for any $\beta, \epsilon, \delta > 0$, Algorithm 1 is $\left(\frac{5T(\alpha)}{4}, \beta \right)$-accurate for $\mathcal{Q}$, as long as $T(\alpha) \in [4\alpha/3, 2\alpha]$.*

Proof. Roughly, the argument is as follows: Assume we did not add any noise to the queries. Then we would answer each query with the true answer $A^{(t)}$ or with $\Lambda^{(t)}$ if $\Lambda^{(t)}$ is sufficiently close to $A^{(t)}$. Thus the only reason the mechanism would fail to be accurate is if it performs too many updates and has to terminate due to the condition $C = B$. But since we only invoke $\mathbf{U}$ when we find a query such that $|Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^{(t)})|$ is large, we are actually generating a database update sequence, which cannot be too long if $\mathbf{U}$ is an efficient iterative database construction. To formalize this intuition we have to consider the effect of the noise on this process and show that with high probability the noise remains in a small enough range that this intuition is indeed correct.

Fix any $\alpha, T(\alpha)$, such that $T(\alpha) \in [4\alpha/3, 2\alpha]$. For brevity, we use T to denote $T(\alpha)$. First, we observe that, with probability $1 - \beta$,

$$\max_{t=1,2,\ldots,k} |Z^{(t)}| \leq T/4.$$

Indeed, by a direct calculation:

$$\Pr\left[\max_{t=1,2,\dots k}|Z^{(t)}|>T/4\right]\le k\cdot\Pr\left[|Z^{(1)}|>T/4\right]$$

$$\le 2k\exp\left(-T/4\sigma\right)$$

$$= 2k\exp\left(-\log(2k/\beta)\right)\le\beta$$

For the rest of the proof we will condition on this event and show that for every $t=1,2,\dots,k$,

$$\left|Q^{(t)}(\mathcal{D}^{(t)})-Q^{(t)}(\mathcal{D})\right|\le 5T/4$$

Assuming the algorithm has not yet terminated, in step 3 we answer each query with either $\Lambda^{(t)}$ s.t.

$$T\ge|\widehat{A}^{(t)}-\Lambda^{(t)}|=|Q^{(t)}(\mathcal{D})+Z^{(t)}-\Lambda^{(t)}|$$

$$\ge|Q^{(t)}(\mathcal{D})-\Lambda^{(t)}|-|Z^{(t)}|\ge|Q^{(t)}(\mathcal{D})-\Lambda^{(t)}|-T/4,$$

(in which case the error is at most $5T/4$); or else we answer directly with $\widehat{A}^{(t)}$, in which case

$$|Q^{(t)}(\mathcal{D})-\widehat{A}^{(t)}|=|Z^{(t)}|\le T/4\le 5T/4.$$

Now it suffices to show that Algorithm 1 does not prematurely terminate (due to the condition $C=B$) before answering every query, and in particular that the sequence of invocations of $\mathbf{U}$ form an $(\mathbf{U},\mathcal{D},\mathcal{Q},\alpha,C)$-database update sequence. Indeed, if this were the case, then we'd be assured (by Definition 3.2) that after B invocations of $\mathbf{U}$, the resulting database $\mathcal{D}^*$ would be $(\alpha,\mathcal{Q})$-accurate. So in every subsequent round we'd have

$$|\widehat{A}^{(t)}-\Lambda^{(t)}|=|Q^{(t)}(\mathcal{D})+Z^{(t)}-Q^{(t)}(\mathcal{D}^{(t)})|\le|Q^{(t)}(\mathcal{D})-Q^{(t)}(\mathcal{D}^{(t)})|+|Z^{(t)}|\le\alpha+T/4\le T,$$

and we'd never make the $B+1^{st}$ update. So to complete the proof, we show that we satisfy the properties in Definition 3.1. Firstly, in every round in which we invoke $\mathbf{U}$,

$$T<|\widehat{A}^{(t)}-\Lambda^{(t)}|\le|Q^{(t)}(\mathcal{D})-Q^{(t)}(\mathcal{D}^{(t)})|+|Z^{(t)}|\le|Q^{(t)}(\mathcal{D})-Q^{(t)}(\mathcal{D}^{(t)})|+T/4$$

$$\implies|Q^{(t)}(\mathcal{D})-Q^{(t)}(\mathcal{D}^{(t)})|>3T/4\ge\alpha$$

so that the update sequence satisfies property 2 of Definition 3.1. Secondly, we have already seen that in every round

$$|\widehat{A}^{(t)}-Q^{(t)}(\mathcal{D})|\le T/4\le\alpha/2,$$

so that the update sequence satisfies property 3 of Definition 3.1. Properties 1 and 4 of Definition 3.1 follow by the construction of Algorithm 1. This completes the proof. $\qquad\square$

In order to get the best accuracy parameters, one can just solve for the equation $\alpha=3T(\alpha)/4$; substituting for $T(\cdot)$, this is the same as solving the following equation for α:

$$\alpha=\frac{3000\sqrt{B(\alpha)}\log(4/\delta)\log(k/\beta)}{\epsilon}.\tag{3}$$

Corollary 4.10. *The Multiplicative Weights mechanism is (ϵ, δ)-differentially private and (α, β) accurate for:*

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

The Median Mechanism is (ϵ, δ)-differentially private and (α, β) accurate for:

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

Proof. The multiplicative weights and median mechanism subroutines are given in Appendix A. By Theorem A.4, the multiplicative weights subroutine is a $B(\alpha)$-IDC for $B(\alpha) = 4n^2 \log |\mathcal{X}|/\alpha^2$. By Theorem A.2, the median mechanism subroutine is a $B(\alpha)$-IDC for $B(\alpha) = n^2 \log k \log |\mathcal{X}|/\alpha^2$. The bounds then follow simply by solving for α in the expression $\alpha = \frac{3000\sqrt{B(\alpha)}\log(4/\delta)\log(k/\beta)}{\epsilon}$. $\quad\square$

Remark 4.11. *We note that for the setting in which the database represents the edge set of a graph $G = (V, E)$, and the class of queries we are interested in is the set of all cut queries, this corresponds to an error bound of $\tilde{O}(\sqrt{|E||V|}\log(V)^{1/4}/\sqrt{\epsilon})$.*

5   An Iterative Database Construction Based on Frieze/Kannan

In this section we describe and analyze an iterative database construction based on the Frieze/Kannan "cut decomposition" [FK99a]. Although the style of analysis we use was originally applied specifically to cuts in [FK99a], we use a generalization of their argument to arbitrary linear queries. To our knowledge, such a generalization was first observed in [TTV09].

Algorithm 2 The Frieze/Kannan-based IDC

$\mathbf{U}_\alpha^{FK}(\mathcal{D}, Q, \widehat{A})$**:**

   If: $\mathcal{D} = \emptyset$ **then:** output $\mathcal{D}' = \emptyset$

   Else if: $Q(\mathcal{D}) - \widehat{A} > 0$ **then:** output $\mathcal{D}' = \mathcal{D} - \frac{\alpha}{|\mathcal{X}|} \cdot Q$

   Else if: $Q(\mathcal{D}) - \widehat{A} < 0$ **then:** output $\mathcal{D}' = \mathcal{D} + \frac{\alpha}{|\mathcal{X}|} \cdot Q$

Note that the sum in Algorithm 2 denotes vector addition.

Theorem 5.1. *Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be a dataset. For any $\alpha > 0$, $\mathbf{U}_\alpha^{FK}$ is a $B(\alpha)$-iterative database construction for a class of linear queries $\mathcal{Q}$, where $B(\alpha) = \frac{\|\mathcal{D}\|_2^2|\mathcal{X}|}{\alpha^2}$.*

Proof. Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be any database and let

$$\left\{(\mathcal{D}^{(t)}, Q^{(t)}, \widehat{A}^{(t)})\right\}_{t=1,\dots,C}$$

be $(\mathbf{U}_\alpha^{FK}, \mathcal{D}, \mathcal{Q}, \alpha, B)$-database update sequence (Definition 3.1). We want to show that $C \leq \|\mathcal{D}\|_2^2|\mathcal{X}|/\alpha^2$. Specifically, that after $\|\mathcal{D}\|_2^2|\mathcal{X}|/\alpha^2$ invocations of $\mathbf{U}_\alpha^{FK}$, the database $\mathcal{D}^{(\|\mathcal{D}\|_2^2|\mathcal{X}|/\alpha^2)}$ is $(\alpha, \mathcal{Q})$-accurate for $\mathcal{D}$, and thus there cannot be a sequence of longer than $\|\mathcal{D}\|_2^2|\mathcal{X}|/\alpha^2$ queries that satisfy property 2 of Definition 3.1.

In order to formalize this intuition, we use a potential argument as in [FK99a] to show that for every $t = 1, 2, \ldots, B$, $\mathcal{D}^{(t+1)}$ is significantly closer to $\mathcal{D}$ than $\mathcal{D}^{(t)}$. Specifically, our potential function is the L_2^2 norm of the database $\mathcal{D} - \mathcal{D}^{(t)}$, defined as

$$\|\mathcal{D}\|_2^2 = \sum_{i \in \mathcal{X}} \mathcal{D}(i)^2.$$

Observe that $\|\mathcal{D} - \mathcal{D}^{(1)}\|_2^2 = \|\mathcal{D}\|_2^2$, and $\|\mathcal{D}\|_2^2 \geq 0$. Thus it suffices to show that in every step, the potential decreases by $\alpha^2/|\mathcal{X}|$. We analyze the case where $|Q^{(t)}(\mathcal{D}^{(t)})| < \widehat{A}^{(t)}$, the analysis is the opposite case will be similar. Let $R^{(t)} = \mathcal{D}^{(t)} - \mathcal{D}$. Observe that in this case we have

$$|Q^{(t)}(R^{(t)})| \geq \alpha$$

and

$$Q^{(t)}(R^{(t)}) \geq \widehat{A}^{(t)} - Q^{(t)}(\mathcal{D}^{(t)}) - \alpha/2 > -\alpha/2.$$

Thus we must have

$$Q^{(t)}(R^{(t)}) \geq \alpha.$$

Now we can analyze the drop in potential.

$$
\begin{aligned}
\|R^{(t)}\|_2^2 - \|R^{(t+1)}\|_2^2 &= \|R^{(t)}\|_2^2 - \|R^{(t)} - (\alpha/|\mathcal{X}|) \cdot Q^{(t)}\|_2^2 \\
&= \sum_{i \in \mathcal{X}} R^{(t-1)}(i)^2 - \left(R^{(t)}(i,j) - (\alpha/|\mathcal{X}|) \cdot Q^{(t)}(i) \right)^2 \\
&= \sum_{i \in \mathcal{X}} \left(\frac{2\alpha}{|\mathcal{X}|} \cdot R^{(t)}(i) Q^{(t)}(i) - \frac{\alpha^2}{|\mathcal{X}|^2} Q^{(t)}(i)^2 \right) \\
&= \frac{2\alpha}{|\mathcal{X}|} Q^{(t)}(R^{(t)}) - \frac{\alpha^2}{|\mathcal{X}|^2} \sum_{i \in \mathcal{X}} Q^{(t)}(i)^2 \\
&\geq \frac{2\alpha}{|\mathcal{X}|} Q^{(t)}(R^{(t)}) - \frac{\alpha^2}{|\mathcal{X}|^2} |\mathcal{X}| \\
&\geq \frac{2\alpha^2}{|\mathcal{X}|} - \frac{\alpha^2}{|\mathcal{X}|} = \frac{\alpha^2}{|\mathcal{X}|}
\end{aligned}
$$

This bounds the number of steps by $\|\mathcal{D}\|_2^2 |\mathcal{X}|/\alpha^2$, and completes the proof. $\qquad \square$

Corollary 5.2. *Algorithm 1, instantiated with $\mathbf{U}_\gamma^{FK}$ for $\gamma = O\left(\epsilon^{-1/2} n_2^{1/4} |\mathcal{X}|^{1/4} \sqrt{\log(k/\beta)} \right)$ is (ϵ, δ)-differentially private and an (α, β)-accurate interactive release mechanism for query set $\mathcal{Q}$ with $\alpha = O\left(\frac{n_2^{1/4} |\mathcal{X}|^{1/4} \sqrt{\log(k/\beta) \log(1/\delta)}}{\sqrt{\epsilon}} \right)$ where $n_2 = \|\mathcal{D}\|_2^2$. Note that for databases that are subsets of the data universe (rather than multisets), $n_2 = n$.*

Remark 5.3. *We note that for the setting in which the database represents the edge set of a graph $G = (V, E)$, and the class of queries we are interested in is the set of all cut queries, this bounds corresponds to $\tilde{O}(|V||E|^{1/4}/\sqrt{\epsilon})$. This is an improvement on the bound given by the multiplicative weights IDC for dense graphs: when $|E| \geq \Omega(|V|^2/\log|V|)$.*

6 Results for Synthetic Data

In this section, we consider the more demanding task of efficiently releasing *synthetic data* for the class of cut queries on graphs. The task at hand here is to actually generate another graph that approximates the private graph with respect to cuts. Such a graph can then simply be released to data analysts, who can examine it at their leisure. This is preferable to the interactive setting, in which an actual graph is never produced, and a central stateful API must be maintained to handle queries as they come in from data analysts. Our algorithm is simple, and is based on releasing a noisy histogram. Note that for a graph, $|\mathcal{X}| = \binom{|V|}{2}$, and $\mathcal{D} = E$, so as long as $|E| = \Omega(|V|)$, the universe is at most a polynomial in the database size. (Moreover, it is easy to show that there does not exist any $(\epsilon, 0)$-private mechanism that has error $o(|V|)$, so the only interesting cases are when $|E| = \Omega(|V|)$.)

Consider a database whose elements are drawn from $\mathcal{X}$; we represent this as a vector (histogram) $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$. Let $\widehat{\mathcal{D}} = \mathcal{D} + (Y_1, \ldots, Y_{|\mathcal{X}|})$ be a "noisy" database, where each $Y_i \sim \text{Lap}(1/\varepsilon)$ is an independent draw from the Laplace distribution. Note that by Theorem 2.3, the procedure which on input $\mathcal{D}$ releases the noisy database $\widehat{\mathcal{D}}$ preserves $(\epsilon, 0)$-differential privacy. This follows because the histogram vector can be viewed as simply the evaluation of the identity query $Q : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{N}^{|\mathcal{X}|}$, which can be easily seen to be 1-sensitive. At this stage, we could release $\widehat{\mathcal{D}}$ and be satisfied that we have designed a private algorithm. There are two issues: first, we must analyze the utility guarantees that $\widehat{\mathcal{D}}$ has with respect to our query set $\mathcal{Q}$. Second, $\widehat{\mathcal{D}}$ is not quite synthetic data. It will be a vector with possibly negative entries, and so does not represent a histogram. Interpreted as a graph, it will be a weighted graph with negative edge weights. This may not be satisfactory, so we must do a little more work.

The utility guarantee of this procedure over the collections $\mathcal{Q}$ of linear queries is also not difficult; i.e., each query $Q \in \mathcal{Q}$ is a vector in $[0, 1]^{|\mathcal{X}|}$, and on any database $\mathcal{D}$ evaluates to $Q(\mathcal{D}) = \langle Q, \mathcal{D} \rangle$.

Lemma 6.1. *Suppose that $\mathcal{Q} \subseteq [0, 1]^{|\mathcal{X}|}$ is some collection of linear queries. For the case $|\mathcal{Q}| \leq (\beta/2)\, 2^{|\mathcal{X}|/6}$, it holds that with probability at least $1 - \beta$,*

$$|Q(\mathcal{D}) - Q(\widehat{\mathcal{D}})| \leq \varepsilon^{-1}\sqrt{6|\mathcal{X}|\log(|\mathcal{Q}|/\beta)}$$

for every query $Q \in \mathcal{Q}$. For general $\mathcal{Q}$, the error bound is $O(\varepsilon^{-1}\sqrt{|\mathcal{X}|\log(|\mathcal{X}|/\beta)\log(|\mathcal{Q}|/\beta)})$.

Proof. Note that $(Q(\widehat{\mathcal{D}}) - Q(\mathcal{D})) = \langle Q, \mathcal{D} - \widehat{\mathcal{D}} \rangle \sim \sum_i q_i Y_i$, where each random variable $Y_i \sim \text{Lap}(1/\varepsilon)$, and $q_i \in [0, 1]$ is the ith coordinate of query Q. By a tail bound for sums of Laplace random variables (Theorem 6.5), we know that

$$\Pr[|\textstyle\sum_{i=1}^{|\mathcal{X}|} q_i Y_i| \geq \alpha] \leq 2\exp(-\alpha^2 \varepsilon^2 / 6|\mathcal{X}|),$$

as long as $\alpha \leq |\mathcal{X}|/\varepsilon$. If we set $\alpha = \varepsilon^{-1}\sqrt{12|\mathcal{X}|\log(2|\mathcal{Q}|/\beta)}$ the probability bound on the right hand side at most β, and the condition $\alpha \leq |\mathcal{X}|/\varepsilon$ translates to $|\mathcal{Q}| \leq (\beta/2)\, 2^{|\mathcal{X}|/6}$.

The proof for the general case, where we do not assume a bound on the size $|\mathcal{Q}|$, loses an extra factor of $O(\sqrt{\log(|\mathcal{X}|/\beta)})$. Indeed, with probability at least $1 - \beta/2$, each of the absolute values $|Y_i|$'s are at most $L = O(1/\varepsilon)\log(|\mathcal{X}|/\beta)$. Now, conditioning on this event happening, the sum $\sum_{i=1}^{|\mathcal{X}|} q_i Y_i$ behaves like a sum of $|\mathcal{X}|$-many independent $[-L, L]$-bounded random variables with mean 0: in this case, $\Pr[|\sum_i q_i Y_i| > \alpha] \leq 2e^{-\Omega(\alpha^2/(L^2|\mathcal{X}|))} = e^{-\Omega(\alpha^2 \varepsilon^2/(|\mathcal{X}|\log(|\mathcal{X}|/\beta)))}$ by a standard Chernoff bound. Now setting $\alpha = O(\varepsilon^{-1}\sqrt{|\mathcal{X}|\log(|\mathcal{X}|/\beta)\log(|\mathcal{Q}|/\beta)})$ causes this probability to be at most $\beta/2$; by a union bound, the probability of large deviations is at most β. $\quad\square$

In summary, note that the bounds on the error are $\approx \varepsilon^{-1}\sqrt{|\mathcal{X}|\log|\mathcal{Q}|}$, with some correction terms depending on whether the size of the query set is at most $2^{O(|\mathcal{X}|)}$ or larger.

6.1 Randomized Response and Synthetic Data for Cut Queries

For the case of cuts in graph on a vertex set V, the database is a vector in $\{0,1\}^{\binom{|V|}{2}}$, and the noisy database just adds independent $\mathrm{Lap}(1/\varepsilon)$ noise to each bit value. Since the query set $\mathcal{Q}_{cuts}$ has size $2^{2|V|}$, (namely it consists of all (S,T) pairs), we have $|\mathcal{Q}_{cuts}| \ll (\beta/2)2^{|\mathcal{X}|/6}$ for all reasonable β and $|V|$, we can use the randomized response analysis above to get accuracy

$$O\left(\left(\binom{|V|}{2}\log(|\mathcal{Q}_{cuts}|/\beta)\right)^{1/2}/\varepsilon\right) = O((|V|^{3/2} + |V|\log 1/\beta)/\varepsilon)$$

with probability at least $1 - \beta$. In fact, one can give a slightly tighter analysis where the accuracy depends on the size of the sets S, T—by observing that the number of random variables participating in a cut query (S,T) is exactly $|S||T|$, one can show that the accuracy for all cuts is whp $O(\varepsilon^{-1}\sqrt{|V||S||T|})$.

Viewing the noisy database $\widehat{\mathcal{D}}$ as a weighted graph $\widehat{G}$, where the weight of (u,v) is $\mathbf{1}_{(u,v)\in E(G)} + \mathrm{Lap}(1/\varepsilon)$, note that $\widehat{G}$ has negative weight edges and hence cannot be considered synthetic data. We can remedy the situation (using the idea of solving a suitable linear program [BCD$^+$07, DNR$^+$09]):

Lemma 6.2 (Synthetic Data for Cuts). *There is a computationally efficient $(\varepsilon, 0)$-differentially private randomized algorithm that takes a unweighted graph G and outputs a synthetic graph G' such that, with high probability, $\|G - G'\|_C \leq O(|V|^{3/2}/\varepsilon)$—all cuts in G and G' are within $O(|V|^{3/2}/\varepsilon)$ additive error.*

Proof. First we construct the noisy datastructure $\widehat{D}$ by perturbing each entry of $\mathcal{D}$ with independent noise drawn from $\mathrm{Lap}(1/\epsilon)$. All further operations will be conducted on $\widehat{D}$, and so the entire algorithm will be $(\epsilon, 0)$-differentially private. let $z_{i,j}$ denote the i,j'th entry of $\widehat{D}$: i.e. $z_{i,j} = \mathcal{D}[i,j] + \mathrm{Lap}(1/\epsilon)$. Let us condition on the event that for every cut (S,T), the additive error bound is $O(|V|^{3/2}/\varepsilon)$. Now define the following LP:

$$\min \lambda$$
$$\text{such that } \sum_{ij \in S \times T} (x_{ij} - z_{ij}) \leq \lambda \qquad \forall S, T$$
$$\sum_{ij \in S \times T} (z_{ij} - x_{ij}) \leq \lambda \qquad \forall S, T$$
$$x_{ij} \in [0,1].$$

There exists a feasible solution to this LP with $\lambda = O(|V|^{3/2}/\varepsilon)$, since we can just use the original graph to get the solution $x_{ij} = \mathbf{1}_{(ij \in E(G))}$. Now if we solve the LP, and output the optimal feasible solution to the LP, it would be a weighted graph G' such that $\|G - G'\|_C \leq O(|V|^{3/2}/\varepsilon)$.

Since the LP has exponentially many constraints, it remains to show how to solve the LP. Define the matrix A with $A_{ij} = x_{ij} - z_{ij}$, and define $A(S,T) := \sum_{i \in S, j \in T} A_{ij}$, then the separation oracle must find sets S, T such that $|A(S,T)|$ is larger than λ. Equivalently, it suffices to approximately compute the cut norm of the matrix A. There is a constant-factor approximation algorithm of Alon and Naor for the cut norm problem [AN06]; using this we can solve the LP above to within constant factors of optimum. $\qquad \square$

Remark 6.3. *The procedure outlined above results in outputting a weighted graph (with non-negative edge weights) $\mathcal{D}' \in [0,1]^{|V| \times |V|}$. Note that if the original graph was unweighted: $\mathcal{D} \in \{0,1\}^{|V| \times |V|}$ and it is desired to output another unweighted graph, we can simply randomly round*

$\mathcal{D}'$ to an integral solution in the obvious way. This does not incur any asymptotic loss in the stated accuracy bound.

6.2 A Spectral Solution, and Rank-1 Queries

The Alon-Naor algorithm involves solving SDPs which are computationally intensive, but we can avoid that by using the tighter accuracy bound of $O(\varepsilon^{-1}\sqrt{|V||S||T|})$ we proved. Consider the modified LP:

$$\min \lambda$$
$$\text{such that} \sum_{ij \in S \times T} (x_{ij} - z_{ij}) \leq \lambda\sqrt{|S||T|} \qquad \forall S, T$$
$$\sum_{ij \in S \times T} (z_{ij} - x_{ij}) \leq \lambda\sqrt{|S||T|} \qquad \forall S, T$$
$$x_{ij} \in [0, 1].$$

Again, this LP has a feasible solution (whp) with $\lambda = O(|V|^{1/2}/\varepsilon)$. And solving the LP to within a factor of ρ, and outputting a near-optimal feasible solution to the LP would give a synthetic weighted graph G' such that $\|G - G'\|_C \leq O(\rho \cdot \sqrt{|V||S||T|}/\varepsilon) = O(\rho \cdot |V|^{3/2}/\varepsilon)$. To this end, define the *normalized cut norm* as

$$\|A\|_{NC} := \max_{S,T} \frac{|A(S,T)|}{\sqrt{|S||T|}};$$

Now the separation problem is to find S, T approximately maximizing the normalized cut norm. For this we use a theorem of Nikiforov [Nik09] which says that if $\sigma_1(A)$ is the top singular value of A (and $\|A\|_2$ is A's spectral norm) then

$$\|A\|_{NC} \leq \sigma_1(A) = \|A\|_2 \leq \|A\|_{NC} \cdot O(\log |V|). \tag{4}$$

There is also a polynomial-time algorithm that given the top singular value/vector for A, returns a normalized cut (S', T') of value $|A(S', T')|/\sqrt{|S'||T'|} \geq \sigma_1(A)/O(\log |V|) \geq \|A\|_{NC}/O(\log |V|)$. Using this as a separation oracle we can solve the LP to within $\rho = O(\log |V|)$ of the optimum, and hence get an additive error of $O(\varepsilon^{-1} \log |V|\sqrt{|V||S||T|}) = O(\varepsilon^{-1}|V|^{3/2} \log |V|)$.

Remark 6.4. *We note that the theorem of Nikiforov quoted above [Nik09] also implies that the synthetic graph released by our algorithm is useful for the (infinite) set of rank-1 queries $\mathcal{Q}_{r1}$ as well as the set of cut queries, with only an $O(\log |V|)$ factor loss in the additive approximation for each query.*

6.3 A Tail Bound for Laplace Distributions

The following tail bound for Laplace random variables uses standard techniques, we give it here for completeness.

Theorem 6.5. *Suppose $\{Y_i\}_{i=1}^k$ are i.i.d. $\text{Lap}(b)$ random variables, and scalars $q_i \in [0, 1]$. Define $Y := \sum_i q_i Y_i$. Then*

$$\Pr[Y \geq \alpha] \leq \begin{cases} \exp\left(-\frac{\alpha^2}{6kb^2}\right). & \text{if } \alpha \leq kb \\ \exp\left(-\frac{\alpha}{6b}\right). & \text{if } \alpha > kb \end{cases} \tag{5}$$

Proof. Suppose $Y_1, Y_2, \ldots, Y_k$ are i.i.d. Laplace(b) random variables, and $q_1, q_2, \ldots, q_k$'s are scalars in $[0, 1]$. We now give a tail bound for $Y := \sum_i q_i Y_i$. It is useful to recall that the moment generating function for a Laplace random variable $Y \sim \text{Lap}(b)$ is $E[e^{tY}] = 1/(1 - b^2 t^2)$ for $|t| < 1/b$, and also that if $Y \sim \text{Lap}(b)$, then $cY \sim \text{Lap}(cb)$ for $c > 0$. Hence for $t \in [0, 1/b]$, we have

$$\Pr[Y \geq \alpha] = \Pr[e^{tY} \geq e^{t\alpha}] \leq \frac{E[e^{tY}]}{e^{t\alpha}}$$

$$= e^{-t\alpha} \prod_{i=1}^{k} E[e^{tY_i}] = e^{-t\alpha} \prod_i (1 - (q_i bt)^2)^{-1}$$

$$= \exp(-t\alpha - \sum_i \log(1 - (q_i bt)^2))$$

$$= \exp(-t\alpha + \sum_i ((q_i bt)^2 + (q_i bt)^4/2 + (q_i bt)^6/3 + \cdots))$$

where we used the Taylor series expansion (and hence need that $|q_i bt| < 1$). The last expression only worsens as the q_i's increase, so the worst case is when all $q_i = 1$, when we get a bound of

$$\exp(-t\alpha + k((bt)^2 + (bt)^4/2 + (bt)^6/3 + \cdots))$$
$$\leq \exp(-t\alpha + k((bt)^2 + (bt)^4 + (bt)^6 + \cdots))$$
$$\leq \exp\left(-t\alpha + k\frac{(bt)^2}{1 - (bt)^2}\right). \tag{6}$$

Let us set $t := \frac{\alpha}{2kb^2}$. Recall that we needed the condition that $t \in [0, 1/b]$, so let us assume that $\alpha \leq kb$. This implies that $tb = \alpha/2kb \leq 1/2$. Hence, plugging in this setting for t, and noting that $(1 - (tb)^2) \geq 3/4$, we get

$$\Pr[Y \geq \alpha] \leq \exp\left(-\frac{\alpha^2}{2kb^2} + k\frac{\alpha^2}{(3/4)(2kb)^2}\right) = \exp\left(-\frac{\alpha^2}{6kb^2}\right).$$

This completes the proof for the case $\alpha \leq kb$. Now suppose $\alpha > kb$; in that case let us set $t = 1/2b$—substituting this into (6) gives us a tail bound of $\exp(-\alpha/2b + k/3)$. And since $\alpha > kb$, this is bounded by $\exp(-\alpha/6b)$. This proves the theorem. $\square$

7 Towards Improving on Randomized Response for Synthetic Data

In this section, we consider one possible avenue towards giving an efficient algorithm for privately generating synthetic data for graph cuts that improves over randomized response. We first show how generically, any efficient Iterative Database Construction algorithm can be used to give an efficient *offline* algorithm for privately releasing synthetic data when paired with an efficient *distinguisher*. The analysis here follows the analysis of [GHRU11], who analyzed the corresponding algorithm when instantiated with the multiplicative weights algorithm, rather than a generic Iterative Database Construction algorithm.

We will pair an Iterative Database Construction algorithm for a class of queries $\mathcal{C}$ with a corresponding *distinguisher*.

Definition 7.1 (($F(\epsilon), \gamma$)-Private Distinguisher)**.** Let $\mathcal{Q}$ be a set of queries, let $\gamma \geq 0$ and let $F(\epsilon) : \mathbb{R}^+ \to \mathbb{Z}$ be a function. An algorithm $\text{Distinguish}_\epsilon : \mathbb{N}^{|\mathcal{X}|} \times \mathbb{N}^{|\mathcal{X}|} \to \mathcal{Q}$ is an ($F(\epsilon), \gamma$)-Private

Distinguisher for $\mathcal{Q}$ if for every setting of the privacy parameter ϵ, it is ϵ-differentially private with respect to $\mathcal{D}$ and if for every $\mathcal{D}, \mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|}$ it outputs a $Q^* \in \mathcal{Q}$ such that $|Q^*(\mathcal{D}) - Q^*(\mathcal{D}')| \geq \max_{Q \in \mathcal{Q}} |Q(\mathcal{D}) - Q(\mathcal{D}')| - F(\epsilon)$ with probability at least $1 - \gamma$.

Note that in [GHRU11], we referred to a distinguisher as an agnostic learner. Indeed, a distinguisher is solving the agnostic learning problem for its corresponding set of queries. We here refer to it as a distinguisher to emphasize its applicability beyond the typical realm of learning (e.g. we here hope to apply a distinguisher to a graph cuts problem).

Algorithm 3 The Iterative Construction (IC) Mechanism. It takes as input an $(F(\epsilon), \gamma)$-Private Distinguisher $\text{Distinguish}_\epsilon$ for $\mathcal{Q}$, together with an $B(\alpha)$-iterative database construction algorithm $\mathbf{U}_\alpha$. for $\mathcal{Q}$

$\mathbf{IC}(\mathcal{D}, \epsilon, \delta, \alpha, \text{Distinguish}, \mathbf{U})$:

   Let $D^0 = \mathbf{U}(\emptyset, \cdot, \cdot)$.

   Let $\epsilon_0 = \epsilon_0(\alpha) =\leftarrow \frac{\epsilon}{4\sqrt{B(\alpha)\log(1/\delta)}}$

   for $t = 1$ to $B(\alpha)$ **do**

      Let $Q^{(t)} = \text{Distinguish}_{\epsilon_0}(\mathcal{D}, D^{t-1})$

      Let $\widehat{A}^{(t)} = Q^{(t)}(\mathcal{D}) + \text{Lap}\left(\frac{1}{\epsilon_0}\right)$.

      if $|\widehat{A}^{(t)} - Q^{(t)}(D^{t-1})| < 3\alpha/4$ **then**

         Output $\mathcal{D}' = D^{t-1}$.

      else

         Let $D^t = \mathbf{U}_{\alpha/2}(\mathcal{D}^{t-1}, Q^{(t)}, \widehat{A}^{(t)})$.

      end if

   end for

   Output $\mathcal{D}' = D^{B(\alpha)}$.

What follows is a formal analysis, but the intuition for the mechanism is simple: we simply run the iterative database construction algorithm to construct a hypothesis that approximately matches $\mathcal{D}$ with respect to the queries $\mathcal{C}$. If our distinguisher succeeds in finding a query that has high discrepancy between the hypothesis database and the true database whenever one exists, then our IDC algorithm will output a database that is β-accurate with respect to $\mathcal{C}$. This requires at most T iterations, and so we access the data only $2T$ times using $(\epsilon_0, 0)$-differentially private methods (running the given distinguisher, and then checking its answer with the Laplace mechanism). Privacy will therefore follow from the composition theorem.

Theorem 7.2. *Given parameters $\epsilon, \delta < 1$, The IC mechanism is (ϵ, δ) differentially private.*

Proof. The mechanism accesses the data at most $2B(\alpha)$ times using algorithms that are ϵ_0-differentially private. By Theorem **??**, the mechanism is therefore (ϵ', δ)-differentially private for $\epsilon' = \sqrt{4B(\alpha)\ln(1/\delta)}\epsilon_0 + 2B(\alpha)\epsilon_0(e^{\epsilon_0} - 1)$. Plugging in our choice of ϵ_0 proves the claim. $\square$

Theorem 7.3. *Given an $(F(\epsilon), \gamma)$-private distinguisher and a $B(\alpha)$-IDC, the Iterative Construction mechanism is α, β accurate for:*

$$\alpha \geq \max\left[\frac{16\sqrt{B(\alpha)\log(1/\delta)}\log(2B(\alpha)/\beta)}{\epsilon}, 2F\left(\frac{\epsilon}{4\sqrt{B(\alpha)\log(1/\delta)}}\right)\right]$$

so long as $\gamma \leq \beta/(2B(\alpha))$.

Proof. The analysis is straightforward. First we observe that because the algorithm runs for at most $B(\alpha)$ steps, except with probability at most $\beta/2$, for all t:

$$|\widehat{A}^{(t)} - Q^{(t)}(\mathcal{D})| \leq \frac{1}{\epsilon_0} \log \frac{2B(\alpha)}{\beta} = \frac{4\sqrt{B(\alpha)\log(1/\delta)}}{\epsilon} \log \frac{2B(\alpha)}{\beta} \leq \frac{\alpha}{4}$$

Note that by assumption, $\gamma \leq \beta/(2B(\alpha))$, so we also have that except with probability $\beta/2$,

$$|Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^{t-1})| \geq \max_{Q' \in \mathcal{Q}} |Q'(\mathcal{D}) - Q'(\mathcal{D}^{t-1})| - F(\frac{\epsilon}{4\sqrt{B(\alpha)\log(1/\delta)}}) \geq \max_{Q' \in \mathcal{Q}} |Q'(\mathcal{D}) - Q'(\mathcal{D}^{t-1})| - \frac{\alpha}{2}$$

For the rest of the argument, we will condition on both of these events occurring, which is the case except with probability β. There are two cases. Either a database $\mathcal{D}' = D^{B(\alpha)}$ is output, or database $\mathcal{D}' = D^{t-1}$ for $t \leq B(\alpha)$ is output. First, suppose $\mathcal{D}' = D^{B(\alpha)}$. Since for all t $|\widehat{A}^{(t)} - Q^{(t)}(D^{t-1})| \geq 3\alpha/4$ and by our conditioning, $|\widehat{A}^{(t)} - Q^{(t)}(\mathcal{D})| \leq \frac{\alpha}{4}$, the sequence $(D^t, Q^{(t)}, \widehat{A}^{(t)})$, formed a maximal $(\mathbf{U}_{\alpha/2}, \mathcal{D}, \mathcal{Q}, \alpha/2, B(\alpha))$-Database Update Sequence. Therefore, we have that $\max_{Q \in \mathcal{Q}} |Q(\mathcal{D}) - Q(\mathcal{D}')| \leq \alpha/2$ as desired. Next, suppose $\mathcal{D}' = D^{t-1}$ for $t \leq B(\alpha)$. Then it must have been the case that for some t, $|\widehat{A}^{(t)} - Q^{(t)}(D^{t-1})| < 3\alpha/4$. By our conditioning, in this case it must be that $Q^{(t)}(\mathcal{D}) - Q^{(t)}(D^{t-1}) < \alpha/2$, and that therefore by the properties of an $(F(\epsilon_0), \gamma)$-distinguisher:

$$\max_{Q \in \mathcal{Q}} |Q(\mathcal{D}) - Q(\mathcal{D}')| < \alpha/2 + F(\epsilon_0) \leq \alpha$$

as desired. $\qquad\qquad\square$

Note that the running time of the algorithm is dominated by the running time of the IDC algorithm and of the distinguishing algorithm: efficient IDC algorithms paired with efficient distinguishing algorithms for a class of queries $\mathcal{Q}$ automatically correspond to efficient algorithms for privately releasing synthetic data useful for $\mathcal{Q}$. For the class of graph cut queries, both the multiplicative weights IDC and the Frieze/Kannan IDC are computationally efficient. Therefore, one approach to finding a computationally efficient algorithm for releasing synthetic data useful for cut queries is to find an efficient private distinguishing algorithm for cut queries.

One curious aspect of this approach is that it might in fact be computationally easier to release a *larger* class of queries than cut queries, even though this is a strictly more difficult task from an information theoretic perspective. For example, solving the distinguishing problem for cut queries on graphs $\mathcal{D}$ and $\mathcal{D}'$ is equivalent to finding a pair of sets (S, T) which witness the cut-norm on the graph $\mathcal{D} - \mathcal{D}'$. On the other hand, solving the distinguishing problem for rank-1 queries (which include cut queries, and are a larger class) is equivalent to finding the best rank-1 approximation to the adjacency matrix $\mathcal{D} - \mathcal{D}'$. The former problem is NP-hard, whereas the latter problem can be quickly solved non-privately using the singular value decomposition.

Corollary 7.4. *An efficient $(F(\epsilon), \gamma)$-distinguisher for the class of rank-1 queries for $F(\epsilon) = T/\epsilon$ would yield an (α, β)-accurate mechanism for releasing synthetic data for graph cuts (and all rank-1 queries) for any $\beta \geq \Omega(\exp(-\epsilon T))$ and: $\alpha_{MW} = 2\sqrt[4]{2}\epsilon^{-1/2}\sqrt{Tm}(\log|V|\log(1/\delta))^{1/4}$ using the multiplicative weights IDC, or: $\alpha_{FK} \geq 2\epsilon^{-1/2}(m\log(1/\delta))^{1/4}\sqrt{|V|T}$ using the Frieze/Kannan IDC*

Proof. The Multiplicative Weights mechanism is a $B(\alpha)$-IDC for the class of rank-1 queries on graphs with $m = |E|$ edges and $|V|$ vertices for $B(\alpha) = 2m^2 \log|V|/\alpha^2$. We can set:

$$\alpha \geq F\left(\frac{\epsilon}{4\sqrt{B(\alpha)\log(1/\delta)}}\right) = \frac{4\sqrt{2}Tm\sqrt{\log|V|\log(1/\delta)}}{\alpha\epsilon}$$

which allows us to take

$$\alpha \geq \frac{2\sqrt[4]{2}\sqrt{Tm}\left(\log|V|\log\frac{1}{\delta}\right)^{1/4}}{\sqrt{\epsilon}}$$

The Frieze/Kannan algorithm is a $B(\alpha)$-IDC with $B(\alpha) = m|V|^2/\alpha^2$. We can set:

$$\alpha \geq F\left(\frac{\epsilon}{4\sqrt{B(\alpha)\log(1/\delta)}}\right) = \frac{4T\sqrt{m\log(1/\delta)}|V|}{\alpha\epsilon}$$

which allows us to take:

$$\alpha \geq \frac{2(m\log 1/\delta)^{1/4}\sqrt{|V|\cdot T}}{\sqrt{\epsilon}}$$

$\square$

We remark that for the class of rank-1 queries, an efficient $(F(\epsilon), \gamma)$-distinguisher with $F(\epsilon) = \tilde{O}\left(\frac{|V|}{\epsilon}\right)$ would be sufficient to yield an efficient algorithm for releasing synthetic data useful for cut queries, with guarantees matching those of the best known algorithms for the interactive case, as listed in Table 1. For graphs for which the size of the edge set $m \leq \Omega(n^2)$, this would yield an improvement over our randomized response mechanism, which is the best mechanism currently for privately releasing synthetic data for graph cuts. We note that a distinguisher for rank-1 queries must simply give a good rank-1 approximation to the matrix $\mathcal{D} - \mathcal{D}'$. We further note that in the case of the Frieze/Kannan IDC for graph cuts, $\mathcal{D} - \mathcal{D}'$ is always a symmetric matrix (because both the hypothesis is at every step simply the adjacency matrix for an undirected graph, as of course is the private database), and hence an algorithm for finding accurate rank-1 approximations merely for symmetric matrices would already yield an algorithm for releasing synthetic data for cuts! Unlike classes of queries like conjunctions, for which their are imposing barriers to privately outputting useful synthetic data [UV11, GHRU11], there are as far as we know no such barriers to improving our randomized-response based results for synthetic data for graph cuts. We leave finding such an algorithm, for privately giving low rank approximations to matrices, as an intriguing open problem.

References

[AN06]   Noga Alon and Assaf Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803 (electronic), 2006.

[BCD$^+$07]   B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 273–282. ACM New York, NY, USA, 2007.

[BDMN05]   A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138. ACM New York, NY, USA, 2005.

[BLR08]   A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 609–618. ACM, 2008.

[CDM$^+$05]   S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 363–385, 2005.

[DMNS06]   C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference TCC*, volume 3876 of *Lecture Notes in Computer Science*, page 265. Springer, 2006.

[DNR$^+$09]   C. Dwork, M. Naor, O. Reingold, G.N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st annual ACM Symposium on the Theory of Computing*, pages 381–390. ACM New York, NY, USA, 2009.

[DRV10]   C. Dwork, G.N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annua IEEEl Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[FK99a]   A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[FK99b]   Alan Frieze and Ravi Kannan. A simple algorithm for constructing Szemerédi's regularity partition. *Electron. J. Combin.*, 6:Research Paper 17, 7 pp. (electronic), 1999.

[GHRU11] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately Releasing Conjunctions and the Statistical Query Barrier. In *Proceedings of the 43rd annual ACM Symposium on the Theory of Computing.* ACM New York, NY, USA, 2011.

[HLM11] M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *Arxiv preprint arXiv:1012.4763*, 2011.

[HR10] M. Hardt and G.N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51st Annua IEEEl Symposium on Foundations of Computer Science*, pages 61–70. IEEE, 2010.

[HT10] M. Hardt and K. Talwar. On the Geometry of Differential Privacy. In *The 42nd ACM Symposium on the Theory of Computing, 2010. STOC'10*, 2010.

[KLN+08] S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What Can We Learn Privately? In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science, 2008. FOCS'08*, pages 531–540, 2008.

[MT07] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science*, 2007.

[Nik09] V. Nikiforov. Cut-norms and spectra of matrices. *Arxiv preprint arXiv:0912.0336*, 2009.

[RR10] A. Roth and T. Roughgarden. Interactive Privacy via the Median Mechanism. In *The 42nd ACM Symposium on the Theory of Computing, 2010. STOC'10*, 2010.

[TTV09] Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Regularity, boosting, and efficiently simulating every high-entropy distribution. In *IEEE Conference on Computational Complexity*, pages 126–136, 2009.

[UV11] Jonathan Ullman and Salil P. Vadhan. PCPs and the hardness of generating private synthetic data. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 400–416. Springer, 2011.

A Other Iterative Database Construction Algorithms

In this section, we demonstrate how the median mechanism and the multiplicative weights mechanism fit into the IDC framework. These mechanisms apply to general classes of linear queries $\mathcal{Q}$.

A.1 The Median Mechanism

In this section, we show how to use the median database subroutine as an Iterative Database Construction.

Definition A.1 (Median Datastructure). A median datastructure $\mathbf{D}$ is a collection of databases $\mathbf{D} \subset \mathbb{N}^{|\mathcal{X}|}$. Any query can be evaluated on a median datastructure as follows: $Q(\mathbf{D}) = \text{Median}(\{Q(\mathcal{D}') : \mathcal{D}' \in \mathbf{D}\})$.

Algorithm 4 The Median Mechanism (MM) Algorithm.

$\mathbf{U}_{k,\alpha}^{MM}(\mathbf{D}^t, Q^{(t)}, \widehat{A}^{(t)})$

    If: $\mathbf{D}^t = \emptyset$ **then:** output $\mathbf{D}^0 = \{\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|} : |\mathcal{D}| = n^2 \log k / \alpha^2\}$

    Else if: $Q^{(t)}(\mathbf{D}^t) - \widehat{A}^{(t)} > 0$ **then:** output $\mathbf{D}' = \mathbf{D}' \setminus \{\mathcal{D} \in \mathbf{D} : Q^{(t)}(\mathcal{D}) \geq Q^{(t)}(\mathbf{D})\}$

    Else if: $Q^{(t)}(\mathbf{D}^t) - \widehat{A}^{(t)} < 0$ **then:** output $\mathbf{D}' = \mathbf{D}' \setminus \{\mathcal{D} \in \mathbf{D} : Q^{(t)}(\mathcal{D}) \leq Q^{(t)}(\mathbf{D})\}$

Theorem A.2. *The Median Mechanism algorithm is a $B(\alpha) = n^2 \log |\mathcal{X}| \log k / \alpha^2$ iterative database construction algorithm for every class of k linear queries $\mathcal{Q}$.*

Proof. Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be any database and let $\left\{ (\mathbf{D}^t, Q^{(t)}, \widehat{A}^{(t)}) \right\}_{t=1,\dots,B}$ be a $(\mathbf{U}_k^{MM}, \mathbf{D}^*, \mathcal{Q}, \alpha, B)$-database update sequence. We want to show that $B(\alpha) \leq n^2 \log |\mathcal{X}| \log k / \alpha^2$. Specifically, that after $n^2 \log |\mathcal{X}| \log k / \alpha^2$ invocations of $\mathbf{U}_{k,\alpha}^{MM}$, the median datastructure $\mathbf{D}^{n^2 \log |\mathcal{X}| \log k / \alpha^2}$ is $(\alpha, \mathcal{Q})$-accurate for $\mathcal{D}$. The argument is simple. First, we have a simple fact from [BLR08]:

Claim A.3. *For any set of k linear queries $\mathcal{Q}$ and any database $\mathcal{D}$ of size n, there is a database $\mathcal{D}'$ of size $|\mathcal{D}'| = n^2 \log k / \alpha^2$ so that $\mathcal{D}'$ is α-accurate for $\mathcal{D}$ with respect to $\mathcal{Q}$.*

From this claim, we have that $|\mathbf{D}^t| \geq 1$ for all t, and so can always be used to evaluate queries. On the other hand, each update step eliminates half of the databases in the median datastructure: $|\mathbf{D}^t| = |\mathbf{D}^{t-1}|/2$. This is because the update step eliminates every database either above or below the median with respect to the last query. Initially $|\mathbf{D}^0| = |\mathcal{X}|^{n^2 \log k / \alpha^2}$, and so there can be at most $B(\alpha) \leq \log n^2 |\mathcal{X}| \log k / \alpha^2$ update steps before we would have $|\mathbf{D}^B| < 1$, a contradiction. $\square$

A.2 The Multiplicative Weights Mechanism

In this section we show how to use the multiplicative weights subroutine as an Iterative Database Construction. The analysis of the multiplicative weights algorithm is not new, and follows [HR10]. It will be convenient to think of our databases in this section as probability distributions, i.e. normalized so that $||\mathcal{D}||_1 = 1$. Note that if we are α/n accurate for the normalized database, we are α-accurate for the un-normalized database with respect to any set of linear queries.

Theorem A.4. *The Multiplicative Weights algorithm is a $B(\alpha) = 4n^2 \log |\mathcal{X}| / \alpha^2$ iterative database construction algorithm for every class of linear queries $\mathcal{Q}$.*

Algorithm 5 The Multiplicative Weights (MW) Algorithm.

$\mathbf{U}_\alpha^{MW}(\mathcal{D}^t, Q^{(t)}, \widehat{A}^{(t)})$:

   Let $\eta \leftarrow \alpha/(2n)$.

   If: $\mathcal{D}^t = \emptyset$ **then:** output $\mathcal{D}' = \mathcal{D} \in \mathbb{R}^{|\mathcal{X}|}$ such that $D_i^0 = 1/|\mathcal{X}|$ for all i.

   if $\widehat{A}^{(t)} < Q^{(t)}(\mathcal{D}^t)$ **then**

     Let $r_t = Q^{(t)}$

   else

     Let $r_t = 1 - Q^{(t)}$

   end if

   Update: For all $i \in [|\mathcal{X}|]$ Let

$$\hat{\mathcal{D}}_i^{t+1} = \exp(-\eta r_t(\mathcal{D}_i^t)) \cdot \mathcal{D}_i^t$$

$$\mathcal{D}_i^{t+1} = \frac{\hat{\mathcal{D}}_i^{t+1}}{\sum_{j=1}^{|\mathcal{X}|} \hat{\mathcal{D}}_j^{t+1}}$$

   Output $\mathcal{D}^{t+1}$.

Proof. Let $\mathcal{D} \in \mathbb{N}^{|\mathcal{X}|}$ be any database and let $\left\{ (\mathcal{D}^{(t)}, Q^{(t)}, \widehat{A}^{(t)}) \right\}_{t=1,\dots,B}$ be a $(\mathbf{U}^{MW}, \mathcal{D}^*, \mathcal{Q}, \alpha, B)$-database update sequence. We want to show that $B(\alpha) \le 4n^2 \log |\mathcal{X}|/\alpha^2$. Specifically, that after $4n^2 \log |\mathcal{X}|/\alpha^2$ invocations of $\mathbf{U}^{MW}$, the database $\mathcal{D}^{(4n^2 \log |\mathcal{X}|/\alpha^2)}$ is $(\alpha, \mathcal{Q})$-accurate for $\mathcal{D}$. First let $\hat{\mathcal{D}} \in \mathbb{R}^{|X|}$ be a normalization of the database $\mathcal{D}$: $\hat{\mathcal{D}}_i = \mathcal{D}_i/\|\mathcal{D}\|_1$. Note that for any linear query, $Q(\mathcal{D}) = n \cdot Q(\hat{\mathcal{D}})$. We define:

$$\Psi_t \overset{\text{def}}{=} D(\hat{\mathcal{D}}||D^t) = \sum_{i=1}^{|\mathcal{X}|} \hat{\mathcal{D}}_i \log \left(\frac{\hat{\mathcal{D}}_i}{D_i^t} \right)$$

We begin with a simple fact:

Claim A.5 ([HR10])**.** *For all t: $\Psi_t \ge 0$, and $\Psi_0 \le \log |\mathcal{X}|$.*

We will argue that in every step for which $|Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^t)| \ge \alpha/n$ the potential drops by at least $\alpha^2/4n$. Because the potential begins at $\log |\mathcal{X}|$, and must always be non-negative, we know that there can be at most $B(\alpha) \le 4n^2 \log |X|/\alpha^2$ steps before the algorithm outputs a database $\mathcal{D}^t$ such that $\max_{Q \in \mathcal{Q}} |Q(\mathcal{D}) - Q(\mathcal{D}^t)| < \alpha/n$, which is exactly the condition that we want.

Lemma A.6 ([HR10])**.**
$$\Psi_t - \Psi_{t+1} \ge \eta \left(r_t(\mathcal{D}^t) - r_t(\mathcal{D}) \right) - \eta^2$$

Proof.

$$\Psi_t - \Psi_{t+1} = \sum_{i=1}^{|\mathcal{X}|} \hat{\mathcal{D}}_i \log\left(\frac{D_i^{t+1}}{D_i^t}\right)$$

$$= -\eta r_t(\mathcal{D}) - \log\left(\sum_{i=1}^{|\mathcal{X}|} \exp(-\eta r_t(x_i)) D_i^t\right)$$

$$\geq -\eta r_t(\mathcal{D}) - \log\left(\sum_{i=1}^{|\mathcal{X}|} D_i^t(1 + \eta^2 - \eta r_t(x_i))\right)$$

$$\geq \eta\left(r_t(\mathcal{D}^t) - r_t(\mathcal{D})\right) - \eta^2$$

$\square$

The rest of the proof now follows easily. By the conditions of an iterative database construction algorithm, $|\widehat{A}^{(t)} - Q^{(t)}(\mathcal{D})| \leq \alpha/(2n)$. Hence, for each t such that $|Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^t)| \geq \alpha/n$, we also have that $Q^{(t)}(\mathcal{D}) > Q^{(t)}(\mathcal{D}_t)$ if and only if $\widehat{A}^{(t)} > Q^{(t)}(\mathcal{D}_t)$. In particular, $r_t = Q^{(t)}$ if $Q^{(t)}(\mathcal{D}^t) - Q^{(t)}(\mathcal{D}) \geq \alpha/n$, and $r_t = 1 - Q^{(t)}$ if $Q^{(t)}(\mathcal{D}) - Q^{(t)}(\mathcal{D}^t) \geq \alpha/n$. Therefore, by Lemma A.6 and the fact that $\eta = \alpha/2n$:

$$\Psi_t - \Psi_{t+1} \geq \frac{\alpha}{2n}\left(r_t(\mathcal{D}^t) - r_t(\mathcal{D})\right) - \frac{\alpha^2}{4n^2} \geq \frac{\alpha}{2n}\left(\frac{\alpha}{n}\right) - \frac{\alpha^2}{4n^2} = \frac{\alpha^2}{4n^2}$$

$\square$