Resource-Efficient Methods for Data Analysis

BY

Duan Tu
B.S., University of Florida, 2020

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois Chicago, 2026

Chicago, Illinois

Defense Committee:
Lev Reyzin, Chair, Advisor
Vishesh Jain
Dhruv Mubayi
György Turán
Ren Wang, Illinois Institute of Technology

Accessibility Statement

An accessible EPUB version of this document can be obtained by contacting the author of the thesis.

Dedication

Dedicated to my parents.

Acknowledgements

I feel extremely fortunate to have had Lev Reyzin as my advisor. Throughout our time working together, he guided me with great patience as I learned new knowledge and developed the skills to think about and solve problems independently. I appreciate him deeply for being a principled and honest mentor. His advice on career and life in general has helped me immensely as I navigate the confusing process of figuring out what kind of young professional and adult I want to become. I want to thank him for the genuine care he has shown me, and for always being a trustworthy source of support and inspiration.

I want to thank Vishesh Jain and Ren Wang for generously welcoming me to join them in exploring research topics that I was not so familiar with. I feel lucky to have had the opportunity to work alongside Vishesh and observe how he thinks about math in such a fast-paced and highly effective way; and I feel honored to have been trusted and encouraged by Ren to take on the leading role in forming a new research project from scratch. Vishesh and Ren have very different mentoring styles, both of which are also different from Lev's. The experience of working with the three of them gave me a well-rounded perspective on how to approach and solve problems.

I am also grateful for several other professors and mentors. I had the opportunity to take classes and read with Dhruv Mubayi, György Turán, Will Perkins, and Marcus Michelen during the first few years of grad school, which was an academically overwhelming period for me. There seemed to be endless math to study and I was unsure how to find my footing. The experience of learning with them not only strengthened my mathematical skills, but also taught me how to approach a daunting problem by taking a few initial stabs at it and gradually finding a thread of ideas to guide my exploration. Likewise, working under Sietse Braakman, Hong Zhang, and Hong Zhang during summer internships exposed me to fields that were foreign to me. My internship mentors helped me gain a clearer understanding of how applied work is carried out outside of academia and taught me valuable skills in their respective disciplines. I am very grateful for their guidance and support.

I am very lucky to have an exceptionally strong support system surrounding me throughout grad school and beyond. Faculty and staff members at the MSCS department have offered me help during every step of my grad school journey. In particular, I want to thank Maureen Madden, without whose support and encouragement my grad school experience would have been far less smooth. I am also grateful for the service of the UIC Counseling Center; it is a privilege to have access to free mental health support through the university. I appreciate my therapist for the professional care and sincere compassion she has offered me. In terms of my peers, the MCS group at UIC and the graduate student body in general are a community of joy and friendship. I enjoyed all the math discussions and casual hangouts with them. I want to give a special shoutout to Nicholas Spanier for helping me get into running. My quality of life has significantly improved because of it. I also want to mention several other friends I met during grad school. I find our friendship interesting in that, on the one hand,

we are very similar because we speak the same mother tongue and carry many of the same life experiences and struggles, but on the other hand we are so different in personality and the work we do that I sometimes zone out (willingly) when they start to talk about math among themselves. I want to thank Ping Wan, Sixuan Lou, and Yeqin Liu for being dining companions that I can always count on. I want to thank Jiamin Li for not being bothered by me napping in our office, and for the late-starting but long-standing fun conversations. I want to thank Zhehao Li for all the exciting adventures, and for helping me feel at home in this city. I want to thank Yijia Chen for being my emergency contact, for living with me for four years, and for always being there to give each other a hand or a hug whenever one of us needed one, just like family would. Furthermore, I want to thank several people I met before my time at UIC. I want to thank Sam Wallace for the mutual support we shared throughout the difficult time of COVID and the confusing transition to young adulthood. I want to thank Cameron Rodríguez for being a wonderful friend who shares all of my interests, and for always being a thoughtful and insightful listener whenever I needed to spiral or cry. I want to thank Zijia Liu for growing up alongside me ever since we both left home, for arguing with me, and for always being there for me.

Last and certainly not least, I want to thank my parents. I cannot imagine being the person I am today without your never-ending, unconditional love and support. I love you both very much, and I hope I get to spend more time with you.

DT

Contribution of Authors

Chapter 2 represents the manuscript *On Lower Bounds For Local Versions of Metric Embeddings* by Vishesh Jain and Duan Tu [**31**]. All components, including the literature review, formulation of definitions and theorems, and writing, were completed jointly with my coauthor.

Chapter 3 represents the paper *On Sample Reuse Methods for Answering k-wise Statistical Queries* [**37**] by Lev Reyzin and Duan Tu. All components, including the literature review, formulation of definitions and theorems, development of algorithms, and writing, were completed jointly with my coauthor.

Chapter 4 represents the manuscript *An Efficient Algorithm for Generating Private Synthetic Data* by Lev Reyzin and Duan Tu. This manuscript is currently under submission. All theoretical components, including the literature review, formulation of definitions and theorems, development of algorithms, and writing, were completed jointly with my coauthor. The experimental work, including designing examples, implementing the code, and generating figures, was done by me.

Contents

List of Figures

List of Tables

List of Abbreviations

JL Lemma	Johnson–Lindenstrauss Lemma
PAC	Probably Approximately Correct
SQ	Statistical Query

Summary

Data that arise in real-life problems are often enormous in volume and possess highly complex internal structure. Modern machine learning algorithms aim to extract useful information from these massive collections of data. However, manipulating such large datasets is expensive in every respect, such as time, storage space, and energy. It is therefore natural to seek methods that reduce sample usage and computational cost when designing algorithms.

There is an inherent trade-off between an algorithm's accuracy and its resource efficiency. Approaches that conserve samples or computational cost typically tolerate a controlled amount of noise, which in turn reduces accuracy. This thesis studies this trade-off from a theoretical standpoint.

Depending on the types of questions one wishes to answer about a dataset, different resource-saving strategies may be appropriate. This thesis examines three such strategies. Chapter 2 studies embedding data from high-dimensional spaces into lower-dimensional ones to reduce representational and computational complexity. Chapter 3 develops techniques for reusing samples when answering statistical queries, thereby lowering the number of samples required. Chapter 4 presents a computationally efficient algorithm for generating synthetic data that can answer statistical queries while preserving privacy.

Across these settings, we quantitatively characterize the extent to which accuracy must be sacrificed in exchange for reductions in sample usage and computational resources.

CHAPTER 1

Introduction

When designing a machine learning algorithm to extract useful information from a dataset, we often strive for *resource-efficiency*: we seek to obtain accurate information while using as few samples and computational operations as possible. However, perfect accuracy and minimal resource usage cannot be achieved simultaneously. Every learning algorithm is subject to an inherent trade-off between the quality of its output and the amount of data and computation it consumes. In order to conserve resources, some loss of accuracy must be tolerated. For example, to reduce the number of required samples, we may reuse data, which introduces risks of overfitting and cross-contamination. When confronted with highly complex data, reducing its dimensionality or disregarding certain features may improve computational tractability but inevitably discards some of the information contained in the original dataset. Similarly, we may limit the required runtime of an algorithm by relaxing stopping criteria or by substituting exact values with estimates at intermediate steps, but this will in return degrade accuracy.

This thesis analyzes the fundamental trade-off between output accuracy and resource efficiency from a theoretical perspective. We study three different strategies to conserve sample and computational resources. In each of the problems studied, we provide explicit quantitative bounds describing how much accuracy must be sacrificed in order to obtain a specified level of savings in sample complexity or computational cost.

Chapter 2 tackles the problem of representing high-dimensional data in lower-dimensional spaces. In many real-world data-analysis tasks, each data point may contain dozens or even hundreds of features, requiring extremely high-dimensional representations. For instance, modeling the daily average temperature in Chicago may require incorporating factors such as wind speed, humidity, weather conditions, and many more. Using all of these features

directly in a learning algorithm imposes significant computational and memory burdens. It is therefore desirable to compress the data into lower-dimensional representations that remain faithful to the essential structure of the original points while being far more efficient to store and manipulate. In doing so, we mitigate the *curse of dimensionality*, which refers to various counterintuitive phenomena that arise only in high-dimensional settings and can severely hurt algorithmic performance. By reducing the dimensionality, we also substantially reduce both the memory and the computational resources required by downstream algorithms. Chapter 2 considers a particular setting of this problem, where we embed points from a high-dimensional space to a lower-dimensional one while preserving, up to a small distortion, only a *subset* of the pairwise distances specified by a bounded degree graph G. We provide a general reduction showing that, in many cases, this is no easier than embedding the points while approximately preserving *all* pairwise distances.

Chapter 3 studies how to reuse samples when answering *statistical queries* about a dataset. A statistical query is a function that measures a statistical property of the data; for example, given the dataset of male students at UIC, the average height is a valid statistical query. A common approach to answering such queries is to draw samples from the dataset and estimate the answer using only those samples, since working with the entire dataset may be computationally cumbersome. This naturally raises the question of whether we can reduce the sampling cost by reusing data points across multiple samples. Chapter 3 explores this possibility. In particular, we focus on *k-wise statistical queries*, which are queries evaluated on samples consisting of k data points. For example, given a dataset of all bus stops in Chicago, the average distance between two bus stops is a 2-wise statistical query, since each sample consists of a pair of bus stops. Chapter 3 examines two methods for reusing data points when constructing k-wise samples. For method (1), we show a trade-off between k, the arity of the query, and M, the total number of queries to be answered. For method (2), we show that it performs no worse than the baseline method (which does not reuse data points), and in fact may achieve lower variance in the resulting estimates.

Switching from conserving sample usage to reducing computational costs, Chapter 4 proposes an efficient algorithm for generating *synthetic data* that can be used to answer statistical queries while preserving the privacy of the original dataset. Protecting the privacy of individuals who contribute data is an important ethical requirement in modern data analysis. While aiming to obtain accurate answers to statistical queries, we must ensure that no participant's sensitive information can be leaked or inferred. For example, if we wish to study the correlation between household income and GPA among UIC students, we aim to understand this correlation for the *population as a whole* without revealing the income or GPA of any *specific student*. One resource-efficient approach to achieving this privacy goal is to release a synthetic dataset that approximates the answers produced by the real dataset but consists entirely of artificial data points. Once such a synthetic dataset is released, any downstream analysis automatically inherits its privacy guarantees, eliminating the need for repeated privacy-preserving computations on the true data and thereby simplifying the overall computational pipeline.

From an algorithmic standpoint, prior work leaves an under-explored territory between existing approaches. On one end, generating a synthetic dataset that is accurate for an *arbitrary* query class is computationally intractable in the worst case [**40, 41**]. On the other end, existing polynomial-time algorithms apply only to restricted families of queries, such as counting queries [**24**]. More general methods that handle broad classes of queries typically rely on oracle efficiency, meaning they assume that NP-hard optimization subroutines can be solved efficiently in practice [**44, 17**]. Chapter 4 fills this gap. Our algorithm avoids oracle assumptions by imposing mild analytical conditions on the query class, namely twice continuous differentiability, and under these conditions achieves running time polynomial in the input size for a broad family of statistical queries.

Each of the subsequent chapters is self-contained and focuses on its own specific problem. The remainder of this chapter provides the necessary background and technical tools that will be used throughout the thesis.

1. Dimension Reduction

Dimension reduction is a cornerstone of modern data analysis and theoretical computer science, providing methods to represent high-dimensional data in lower-dimensional spaces while preserving essential geometric properties. In particular, we are interested in preserving pairwise distances between data points in the space.

In Chapter 2, we say that an embedding f from a metric space (X, d_X) to a metric space (Y, d_Y) has *distortion* at most $D \geq 1$ if there exists $r > 0$ such that

$$r \cdot d_X(x, y) \leq d_Y(f(x), f(y)) \leq D \cdot r \cdot d_X(x, y) \qquad \forall x, y \in X.$$

The Johnson-Lindenstrauss (JL) lemma [**32**] is a seminal result, establishing that any set of n points in Euclidean space can be embedded into $\ell_2^{O(\log n/\varepsilon^2)}$ (the Euclidean space in $O(\log n/\varepsilon^2)$ dimensions) while preserving all pairwise Euclidean distances up to a $(1+\varepsilon)$ distortion. It was shown by Larsen and Nelson that this bound is tight up to constant factors for $\varepsilon \in (n^{-0.499}, 1)$ (see [**34**, Theorem 2] for a more general statement).

Lemma 1.1 (Johnson–Lindenstrauss Lemma [**32**]). *For any $0 < \varepsilon < \frac{1}{2}$ and any integer $m > 4$, let $k = \frac{20 \log m}{\varepsilon^2}$. Then for any set V of m points in $\mathbb{R}^N$, there exists a map $f : \mathbb{R}^N \to \mathbb{R}^k$ such that for all $\mathbf{u}, \mathbf{v} \in V$,*

$$(1-\varepsilon)\|\mathbf{u}-\mathbf{v}\|^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq (1+\varepsilon)\|\mathbf{u}-\mathbf{v}\|^2.$$

Another seminal result in the theory of metric embeddings is Bourgain's embedding theorem [**12**], which asserts that every metric space on n points can be embedded into Euclidean space with distortion $O(\log n)$; this is known to be tight up to constant factors (see, e.g., [**35**]).

Theorem 1.2 (Bourgain's Theorem[**12**]). *Every n-point metric space (V, d_V) can be embedded in ℓ_2 with distortion at most $O(\log n)$.*

2. PAC Learning and Statistical Query Learning

Machine learning is the study of algorithms that learn from data to make accurate predictions or decisions on new, unseen inputs. Formally, let $\mathcal{X}$ be a set of *examples* (or input space) and $\mathcal{Y}$ be a set of possible labels. Let $\mathcal{C}$ be the *concept class*, which is a collection of target functions $c : \mathcal{X} \to \mathcal{Y}$. A learning algorithm $\mathcal{A}$ receives a sample of labeled examples

$$S = \{(x_1, c(x_1)), \ldots, (x_n, c(x_n))\},$$

drawn *independent and identically distributed* (i.i.d.) from an unknown distribution $\mathcal{D}$ over $\mathcal{X}$, and outputs a hypothesis $h \in \mathcal{H}$. The goal of learning is to produce a hypothesis whose *generalization error*

$$\mathrm{err}_{\mathcal{D}}(h) = \Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)]$$

is small; that is, h should perform well not only on the observed sample but also on fresh examples drawn from the same distribution.

The *Probably Approximately Correct* (PAC) learning framework introduced by [**42**] formalizes this objective and is the most foundamental definition for learnability.

Definition 1.3 (PAC Learning, [**42**]). A concept class $\mathcal{C}$ is said to be *PAC-learnable* if there exists an algorithm $\mathcal{A}$ and a polynomial function $\mathrm{poly}(\cdot)$ such that for all $\varepsilon, \delta > 0$, all distributions $\mathcal{D}$ over $\mathcal{X}$, and all concepts $c \in \mathcal{C}$, if the sample size satisfies $n \geq \mathrm{poly}\big(1/\varepsilon, 1/\delta, m, \mathrm{size}(c)\big)$, the hypothesis h_S output by $\mathcal{A}$ on a sample $S \sim \mathcal{D}^n$ satisfies

$$\Pr_{S \sim \mathcal{D}^n}\left[\mathrm{err}_{\mathcal{D}}(h_S) \leq \varepsilon\right] \geq 1 - \delta.$$

Moreover, if $\mathcal{A}$ runs in time polynomial in $1/\varepsilon$, $1/\delta$, m, and $\mathrm{size}(c)$, then $\mathcal{C}$ is said to be *efficiently PAC-learnable*.

In this definition, m is a number such that the computational cost of representing any $x \in \mathcal{X}$ is at most $O(m)$ and $\mathrm{size}(c)$ denote the computational cost to represent any $c \in \mathcal{C}$.

Statistical query (SQ) learning is an alternative framework for machine learning. Originally introduced by [**33**], SQ learning aims to learn a target function class $\mathcal{C}$ under PAC guarantees

using access to an SQ oracle rather than labeled samples. In this thesis, we take $\mathcal{C}$ to be a class of Boolean functions $c : \mathcal{X} \to \{+1, -1\}$. Definition 1.4 formally introduces an SQ oracle.

DEFINITION 1.4 (SQ Oracle, [**36**]). Let $\mathcal{X}$ be a domain, let $\mathcal{D}$ be a distribution over $\mathcal{X}$, and let $\mathcal{Q}$ be a class of statistical queries $q : \mathcal{X}^n \to \mathbb{R}$. Given a query $q \in \mathcal{Q}$ and a tolerance parameter $\tau > 0$, the *statistical query oracle* $\mathrm{SQ}_{\mathcal{D}}(q, \tau)$ returns a value $v \in \mathbb{R}$ in the interval

$$\Big[\mathbb{E}_{X\sim\mathcal{D}^n}[q(X)] - \tau,\ \mathbb{E}_{X\sim\mathcal{D}^n}[q(X)] + \tau\Big].$$

DEFINITION 1.5 (Efficient SQ Learning, [**36**]). We say an algorithm $\mathcal{A}$ *efficiently SQ-learns* a concept class $\mathcal{C}$ if for every $c \in \mathcal{C}$, every probability distribution $\mathcal{D}$ over $\mathcal{X}$, and every $\varepsilon > 0$, there exists a polynomial $p(\cdot, \cdot, \cdot)$ such that:

(1) $\mathcal{A}$ makes at most $p(1/\varepsilon, n, |\mathcal{C}|)$ calls to the SQ oracle,

(2) the tolerance τ satisfies $1/\tau \leq p(1/\varepsilon, n, |\mathcal{C}|)$,

(3) the queries q are evaluable in time $p(1/\varepsilon, n, |\mathcal{C}|)$,

and $\mathcal{A}$ outputs a hypothesis h satisfying $\mathrm{err}_{\mathcal{D}}(h) \leq \varepsilon$.

Notice that unlike Definition 1.3, this definition contains no failure parameter δ, since the SQ oracle allows no failure probability and is assumed to always return answers within the prescribed tolerance. This makes SQ learning a natural restriction of PAC learning: if a class $\mathcal{C}$ is efficiently SQ-learnable, then it is also efficiently PAC-learnable. Indeed, an SQ oracle can be simulated by taking an empirical average over a sufficiently large sample for each query. Standard concentration bounds, such as Hoeffding's inequality (Theorem A.5), ensure that this simulation succeeds with high probability. Formal statement for the simulation is given in Proposition 3.4.

Next, we introduce the ℓ_1-*sensitivity* of a statistical query in Definition 1.6. The ℓ_1-sensitivity measures the magnitude by which perturbing a single data point can change a query's output in the worst case. This definition is used in both Chapter 3 and Chapter 4.

DEFINITION 1.6 (ℓ_1-sensitivity). Let $q : \mathcal{X}^n \to \mathbb{R}$. The (record-level) ℓ_1-sensitivity of q is

$$\Delta_1(q) = \max_{X \sim X'} |q(X) - q(X')|,$$

where $X \sim X'$ denotes two databases $X, X' \in \mathcal{X}^n$ that differ in exactly one record.

3. Differential Privacy

As large-scale data collection becomes more common and technologies make it easier to congregate information across sources, preserving the privacy of individuals in datasets has become a central problem in data analysis. Consequently, it is important to develop formal frameworks to reason about privacy and algorithms that can be provably meet these guarantees. *Differential privacy* provides exactly such a framework. An algorithm is said to be differentially private if the outcome of any analysis does not alter significantly whether or not any one person's data is included, thereby ensuring that individuals face no additional risk by choosing to participate. Definition 1.7 gives the formal definition.

DEFINITION 1.7 (Differential Privacy [**14**]). A randomized mechanism $\mathcal{M}$ with input space $\mathcal{X}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all neighboring inputs $x, x' \in \mathcal{X}$ such that $\|x - x'\|_1 \leq 1$:

$$\mathbb{P}[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(x') \in \mathcal{S}] + \delta,$$

where the probability space is over the randomness of $\mathcal{M}$.

In this thesis, differential privacy appears in two different roles. In Chapter 3, we employ differential privacy as a notion of algorithmic stability. At a high level, differential privacy guarantees that changing a single input example does not significantly alter the algorithm's output. This perspective suggests that a differentially private algorithm must be stable under small perturbations of the training sample. Since generalization from a particular training sample to the underlying data distribution can be viewed as requiring robustness to small perturbations of that sample, we draw on results from the differential privacy literature to establish generalization guarantees for our learning algorithms. In Chapter 4, differential

privacy plays a more direct role: our objective is to design an algorithm that generates synthetic datasets while satisfying prescribed (ε, δ)-privacy guarantees.

The standard way to achieve differential privacy for real-valued queries is the *Laplace mechanism*. The probability density function of the Laplace distribution is given in A.3. The mechanism protects privacy by adding a controlled amount of noise to the true query output. Definition 1.8 states this formally.

DEFINITION 1.8 (Laplace Mechanism [**14**]). Given a query $q : \mathcal{X}^n \to \mathbb{R}$ with ℓ_1-sensitivity $\Delta_1(q) = \rho$, the Laplace mechanism outputs

$$\mathcal{M}_L(X, q(\cdot), \varepsilon) = q(X) + \xi,$$

where ξ is drawn from Laplace$(\xi; \rho/\varepsilon)$, the Laplace distribution centered at 0 with scale ρ/ε.

A key fact about the Laplace mechanism is that it satisfies differential privacy, as stated in the next lemma.

LEMMA 1.9 ([**14**]). *The Laplace mechanism satisfies $(\varepsilon, 0)$-differential privacy.*

The tail behavior of the Laplace mechanism is also well understood.

LEMMA 1.10 (Rephrased from [**16**]). *Let $q : \mathcal{X}^n \to \mathbb{R}$ and let $v = \mathcal{M}_L(X, q(\cdot), \varepsilon)$. Suppose the ℓ_1-sensitivity of q is $\Delta_1(q) = \rho$. Then for all $\delta \in (0, 1]$,*

$$\mathbb{P}\left[\left|q(X) - v\right| \geq \ln\left(\frac{1}{\delta}\right) \cdot \left(\frac{\rho}{\varepsilon}\right)\right] \leq \delta.$$

There are two useful bounds on the privacy loss when we compose multiple algorithms: simple composition and advanced composition. Simple composition provides the elementary bound that, when a learner uses independent queries, its privacy equals to the sum of privacy of all queries. Advanced composition deals with the more complicated situation, one where the learner poses adaptive queries to the same database repeatedly. The exact statements of the two composition results are provided below.

LEMMA 1.11 (Simple Composition [**16**]). *Let $\mathcal{A}_i : X^k \to \{0,1\}$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm for $i = 1, \cdots, M$. Then $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_M)$ is differentially private with parameters $\left(\sum_{i=1}^M \varepsilon_i, \sum_{i=1}^M \delta_i\right)$.*

LEMMA 1.12 (Advanced Composition [**18**]). *For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ)-differentially private mechanisms satisfies $(\varepsilon', M\delta + \delta')$-differentially privacy under M-fold adaptive composition for*

$$\varepsilon' = \varepsilon\sqrt{2M\ln(1/\delta')} + M\varepsilon(e^\varepsilon - 1). \tag{1}$$

When ε is small, the M-fold adaptive composition achieves (ε', δ')-differential privacy for

$$\varepsilon' = O\left(\varepsilon\sqrt{2M\ln(1/\delta')}\right). \tag{2}$$

The proofs of Lemma 1.9, Lemma 1.10, Lemma 1.11, and (1) of Lemma 1.12 can be found in their respective references. The proof of (2) of Lemma 1.12 is given in Chapter A.

CHAPTER 2

On lower bounds for local versions of metric embeddings

1. Introduction

Local dimension reduction. In numerous applications, particularly those dealing with massive or inherently complex datasets like social networks or biological data, preserving the full distance information is often unnecessary and may be computationally expensive without incurring high distortion. Instead, maintaining the *local* structure of the data – the geometry and relationships of "neighboring" points – is of primary interest. This motivates the formal study of *local dimension reduction*, initiated by Abraham, Bartal, and Neiman [**1**] (see also the expanded journal version [**3**]). They proposed several novel formalisms [**1**, Definition 1] to capture the notion of local distortion that are based on the metric space's intrinsic structure, such as *k-local distortion*: an embedding f from a metric space (X, d_X) to a metric space (Y, d_Y) is said to have k-local distortion α if $d_Y(f(u), f(v)) \leq d_X(u, v)$ for all $u, v \in X$ and $d_Y(f(u), f(v)) \geq d_X(u, v)/\alpha$ for every u and every v which is among the k-nearest neighbors of u.

The results of [**1**] demonstrated that for their metric-based notions of local distortion, it is often possible to achieve embeddings with distortion or dimension dependent only on the locality parameter (e.g. k, in the notion of k-local distortion above) rather than the total number of points n. For instance, improving their result [**1**, Theorem 2], they showed in follow up work [**2**, Theorem 1] that any metric space (on n points) can be embedded into $\ell_p^{O(e^p \log^2 k)}$ with k-local distortion $O(\log k/p)$ (for any $k \leq n$, $1 \leq p \leq \log k$). In the low distortion setting of the JL lemma, they showed [**2**, Theorem 2] that for any $\varepsilon > 0$ and $p \geq 1$, an ultrametric space admits an embedding into $\ell_p^{O(\log k/\varepsilon^3)}$ with k-local distortion $(1 + \varepsilon)$.

A natural question (asked in [**1**, Section 11]) is whether the ultrametricity assumption in the above result can be removed; specifically, is there a k-local version of the JL lemma,

i.e. can any finite set of points in ℓ_2 be embedded into $\ell_2^{O(\log k/\varepsilon^2)}$ with distortion $(1+\varepsilon)$? This was answered in the negative by Schechtman and Shraibman [**38**], who constructed a set of $n+1$ points $X \subseteq \mathbb{R}^n$ such that any embedding $f : X \to \ell_2^m$ with 3-local distortion $(1+\varepsilon)$ (for a sufficiently small constant $\varepsilon > 0$) must have dimension $m = \Omega(\log n)$ [**38**, Theorem 9] (see also [**38**, Theorem 8] which obtains a lower bound with near optimal ε dependence under more requirements on the embedding).

Graphically local dimension reduction. The focus of this chapter is a different, graphical notion of local dimension reduction, which was studied by Schechtman and Shraibman [**38**, Section 5].

DEFINITION 2.1 (G-local Distortion). Let (X, d_X) and (Y, d_Y) be metric spaces. Let $G = (X, E)$ be a graph whose vertices are points of X. We say that a (not-necessarily injective) map $f : (X, d_X) \to (Y, d_Y)$ is a G-local D-embedding, where $D \geq 1$ is a real number, if there is a real number $r > 0$ such that

$$r \cdot d_X(x, y) \leq d_Y(f(x), f(y)) \leq D \cdot r \cdot d_X(x, y) \qquad \forall \{x, y\} \in E. \tag{3}$$

The infimum of the numbers $D \geq 1$ such that f is a G-local D-embedding is called the G-local distortion of f.

REMARK 2.2. In the case that G is the complete graph on X, this reduces to the usual notion of (global) distortion. Also, k-local distortion essentially corresponds to the case when G is the k-nearest neighbour graph on X.

Perhaps the most natural choice of locality parameter in this setting is the maximum degree Δ of the graph G. In a similar spirit as the work of Abraham, Bartal, and Neiman [**1**], one can ask whether it is possible to achieve embeddings with G-local distortion or dimension dependent only on this locality parameter Δ. The possibility of this is suggested by the following example, highlighted in both [**2**] and [**38**].

EXAMPLE 2.3. *Let (X, d_X) be the set of points $X = \{e_1, \ldots, e_n\} \subseteq \mathbb{R}^n$ endowed with the Euclidean metric. Alon* [**4**] *showed that any embedding $f : (X, d_X) \to \ell_2^m$ with distortion*

$(1+\varepsilon)$ must satisfy $m = \Omega(\log n/(\log(1/\varepsilon)\varepsilon^2))$. However, for any $G = (X, E)$ of maximum degree Δ, one may construct a G-local embedding $g : X \to \ell_2^{O(\log \Delta/\varepsilon^2)}$ with distortion $(1+\varepsilon)$ as follows. First, since G has maximum degree at most Δ, we may greedily find a map $h : X \to \{e_1, \ldots, e_{\Delta+1}\} \subseteq \ell_2^{\Delta+1}$ such that $h(u) \neq h(v)$ if $\{u, v\} \in E$. The crucial property of h is that it is a G-local isometry i.e for any $\{u, v\} \in E$, $d_X(u, v) = \|h(u) - h(v)\|_2$. Now, we compose h with a Johnson-Lindenstrauss map from $\ell_2^{\Delta+1}$ to $\ell_2^{O(\log \Delta/\varepsilon^2)}$ to obtain the desired embedding g.

Since Example 2.3 gives a standard "hardness" example for the Johnson–Lindenstrauss lemma (Lemma 1.1), one may ask whether there is a G-local version of the JL lemma whose dimension depends only on the maximum degree Δ. For instance, this was asked in [**30**], where it was observed that if this were true, the embedding achieving this result would necessarily have to be non-linear. To the best of our knowledge, the only lower bound known for this problem is due to Schechtman and Shraibman. In [**38**, Theorem 11], they show that for X as in Example 2.3 and for any d-regular $G = (X, E)$ with $d \geq 3$ and second eigenvalue bounded by $d/2$, any G-local $(1+\varepsilon)$-embedding $f : X \to \ell_2^m$ *which is non-contracting* (i.e. satisfies $\|f(x) - f(y)\|_2 \geq (1-\varepsilon)\|x - y\|_2$ for *all* $x, y \in X$) must satisfy $m = \Omega(\log n)$. Of course, the non-contracting assumption precludes the embedding of Example 2.3, which achieves $m = O(\log \Delta/\varepsilon^2)$.

Our contribution. We show that in many settings of interest, including those of the JL lemma and Bourgain's embedding theorem (Theorem 1.2), G-local embeddings do not provide *any* asymptotic saving in the dimension/distortion over global embeddings. This is in sharp contrast to metrically local embeddings (as in [**1**]) for which such savings are possible in high distortion regimes, as discussed above.

Our main technical contribution is a general reduction (Theorem 2.5), which takes as input a metric space (X, d_X) and constructs a metric extension (Z, d_Z) along with a graph $G = (Z, E)$ of maximum degree 3 such that any map $f : (Z, d_Z) \to (Y, d_Y)$ which (approximately) preserves distances for points in Z connected by an edge must necessarily (approximately) preserve *all* pairwise distances among points in X (in fact, there is a version

of this statement for any distortion D). By applying this reduction with (X, d_X) being a known hard instance for some metric embedding problem, we are able to construct hard instances for G-local metric embeddings. We illustrate this with two applications: Corollary 2.9, which is a lower bound for G-local JL which is optimal up to a constant (unless ε is too small as a function of n) and Corollary 2.8, which is a lower bound for G-local embeddings of arbitrary finite metric spaces into Euclidean space, which is again optimal up to a constant. We also highlight some open problems.

2. Statements and proofs

In light of the equilateral space example (2.3) and anticipating an open problem, we will track the *aspect ratio* of the metric spaces appearing in the statements of our results.

Definition 2.4 (Aspect Ratio). Let (X, d_X) be a finite metric space. The aspect ratio of X is defined to be the ratio of the maximum to minimum distance in X, i.e.

$$A_X := \frac{\max_{x \neq y} d_X(x, y)}{\min_{x \neq y} d_X(x, y)}.$$

Our main result is the following.

Theorem 2.5. *Let (X, d_X) be a metric space with $|X| = n$ and aspect ratio A_X.*

Suppose for a metric space (Y, d_Y), there exists a real number $D \geq 1$ such that any embedding $f : (X, d_X) \to (Y, d_Y)$ has distortion at least D. Then, for any $\delta \in (0, 1/100)$, there exists a metric space (Z, d_Z) – with $|Z| = O(n^2)$ and aspect ratio $A_Z = O(A_X \cdot D^2 \log n/\delta)$ – and a graph $G = (Z, E)$ with maximum degree 3 such that for any $h : (Z, d_Z) \to (Y, d_Y)$, the G-local distortion of h is at least $D - \delta$.

Moreover, if (X, d_X) is an ℓ_p-metric space for $p \in [1, \infty]$, then (Z, d_Z) can be taken to be an ℓ_p metric space as well.

Remark. We record several remarks about the optimality of this result.

(1) The bound of 3 on the maximum degree of G is the smallest possible. In Proposition 2.7, we show that for any finite metric space (X, d_X) and any graph $G = (X, E)$ of

maximum degree at most 2, there exists a G-local isometric embedding $f : X \to \ell_p^2$. On the other hand, as discussed earlier, there are examples of (X, d_X) such that any embedding into ℓ_2 incurs distortion $\Omega(\log n)$.

(2) The bound $|Z| = O(n^2)$ cannot be improved in general. Indeed, Indyk and Wagner [**29**, Theorem 6.2] showed that $\Omega(n^2 \log(1/\varepsilon))$ bits are required to sketch finite metric spaces on n points for which all distances are between 1 and 2, up to distortion $(1 + \varepsilon)$ (see [**29**, Section 2] for formal definitions). Applying our theorem with $(Y, d_Y) = \ell_\infty$ (into which every metric embeds isometrically) and $\delta = \varepsilon$, it follows that storing the resulting metric (Z, d_Z) *restricted* to the edges of $G = (Z, E)$, up to relative error $(1 + \varepsilon)$, gives a sketch for (X, d_X) with distortion $(1 + 2\varepsilon)$. By rounding each of the $O(|Z|)$ distances we need to store to the nearest integer power of $(1 + \varepsilon)$ and storing the exponent (after appropriate scaling), this requires $O(|Z| \log(\log(n/\varepsilon)/\varepsilon)) = O(|Z| \log(1/\varepsilon) + |Z| \log\log(n/\varepsilon))$ bits, which contradicts the lower bound in [**29**] (say for $\varepsilon < 1/\log n$) unless $|Z| = \Omega(n^2)$.

(3) On the other hand, we do not know of any obstruction showing that the bounds on the aspect ratio A_Z are not improvable. In particular, it would be interesting if there is a construction with both $\Delta(G)$ and A_Z depending only on A_X, D, δ (in particular, independent of n). See the remark following Corollary 2.9.

Proof of Theorem 2.5. Since $|X| = n$, it follows that (X, d_X) can be isometrically embedded in ℓ_∞^n (e.g., using the Frechét embedding, see [**35**]); in particular, we may identify X with a subset of points in ℓ_∞^n. By adding an arbitrary extra point to X if needed, we may (and will) assume without loss of generality that n is even.

Let $e_1, \ldots, e_{2n-2}$ denote the standard basis of ℓ_∞^{2n-2} and for $i \in [2n-2]$, let $v_i := \alpha \cdot e_i$, where α will be chosen later. Consider the following set of points in ℓ_∞^{2n-2},

$$Z = X \cup \{x + v_i : x \in X, i \in [2n-2]\}$$

and let (Z, d_Z) be the metric space induced by the ℓ_∞ metric on these points. Note that $|Z| = O(n^2)$, as claimed.

We will construct a graph, $G = (Z, E)$ such that (Z, d_Z) and G will satisfy the conclusion of the theorem. The edges of G are the union of the following two sets E_1 and E_2:

E_1 is the union of $|X|$ disjoint complete binary trees, rooted at $x \in X$, where the different binary trees span the points $\{x, x+v_1, \ldots, x+v_{2n-2}\}$ for different choices of $x \in X$. Note that each tree has $\lceil (2n-1)/2 \rceil = n$ leaves; we will assume that these leaves are $x + v_1, \ldots, x + v_n$. Note that in E_1, the roots have degree 2, the leaves have degree 1, and all other vertices have degree 3.

We now describe the set E_2. Let K_n denote the complete graph on X (recall that $|X| = n$ is even, without loss of generality). By Baranyai's Theorem (Theorem A.4), the edges of K_n can be decomposed into a disjoint union of $(n-1)$ (perfect) matchings $M_1, \ldots, M_{n-1}$ (see Figure 2.1 for an illustration).

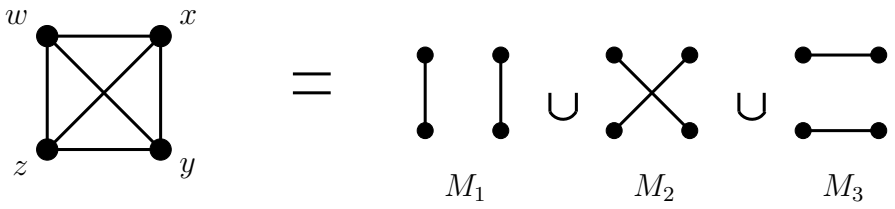

FIGURE 2.1. K_4 can be decomposed into three disjoint perfect matchings M_1, M_2, and M_3.

We define

$$E_2 = \Big\{ \{x + v_j, y + v_j\} : \{x, y\} \in M_j \text{ for some } j \in [n-1] \Big\}.$$

In words, for each pair of points $\{x, y\}$ in X, let M_j be the unique matching in which the edge $\{x, y\}$ appears. Then, E_2 contains an edge between $x + v_j$ and $y + v_j$. Clearly, E_2 is itself a matching, i.e. has maximum degree 1. Also, since $x + v_j$ is a leaf in E_1 for $j \in [n]$, it follows that $G = (Z, E)$ with $E = E_1 \cup E_2$ has maximum degree 3. See Figure 2.2 for an illustration.

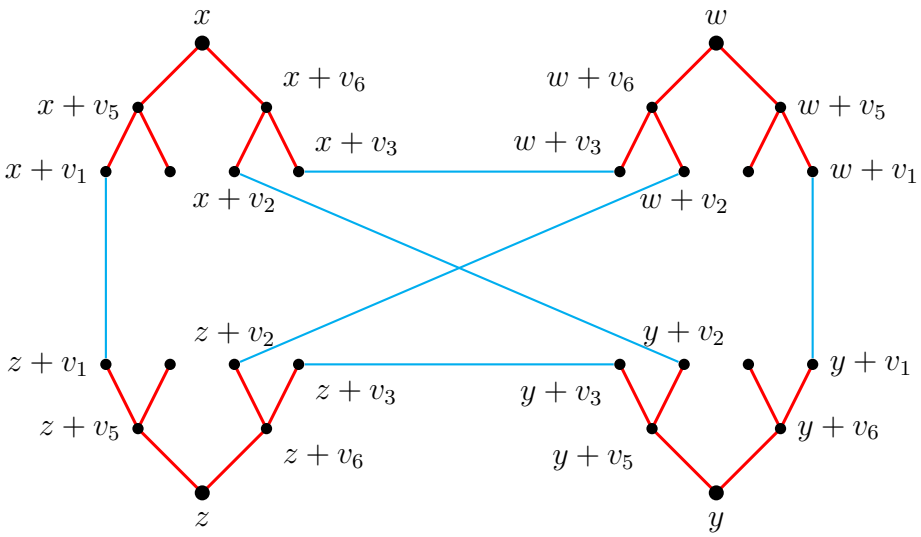

FIGURE 2.2. The construction of $G = (Z, E)$ for $X = \{w, x, y, z\}$. The edges in E', corresponding to the decomposition in Figure 2.1, are depicted in blue.

To complete our construction, we specify the choice of α. Let $\gamma := \min_{x \neq y} d_X(x, y)$ be the minimum distance between two distinct points of X. We set $\alpha = \delta\gamma/(8D^2 \log(2n))$. With this choice of parameters, the assertion about the aspect ratio of (Z, d_Z) is immediate. Additionally, we have the following.

CLAIM 2.6. Suppose the map $h : (Z, d_Z) \to (Y, d_Y)$ is a G-local D'-embedding with $D' \leq D$ (i.e. satisfies Equation (3) with some $r > 0$ and $1 \leq D' \leq D$). Then, for any $x \in X$ and $j \in [n]$,

$$d_Y(h(x), h(x + v_j)) \leq r \cdot \delta/8D \cdot \gamma$$

PROOF. By construction, there is a path from x to $x + v_j$ of length at most $\lceil \log(2n) \rceil$ in E. The edges of the path are of the form $\{x, x + v_k\}$ or $\{x + v_k, x + v_\ell\}$. Denoting the points

along the path by $x = y_1, y_2, \ldots, y_s = x + v_j$, we have

$$\begin{aligned} d_Y(h(x), h(x+v_j)) &\leq d_Y(h(y_1), h(y_2)) + \cdots + d_Y(h(y_{s-1}), h(y_s)) \\ &\leq r \cdot D \cdot (d_Z(y_1, y_2) + \cdots + d_Z(y_{s-1}, y_s)) \\ &\leq r \cdot D \cdot (s-1) \cdot \alpha \\ &\leq r \cdot \delta/8D \cdot \gamma; \end{aligned}$$

here, the first inequality is the triangle inequality; the second inequality uses that h is a G-local D'-embedding with $D' \leq D$; the third inequality uses $d_Z(y_k, y_{k+1}) = \alpha$, which is true by construction; and the last inequality uses our setting of α along with $s \leq \lceil \log n \rceil$. □

With this claim in hand, we proceed with the proof of the theorem. Suppose for contradiction that there exists $h : (Z, d_Z) \to (Y, d_Y)$ which is a G-local D'-embedding with $D' < D - \delta$. Let $f = h|_X : (X, d_X) \to (Y, d_Y)$ denote the restriction of h to X. We will show that the distortion of f is strictly less than D, thereby contradicting our assumption.

For $x, y \in X$, $x \neq y$, let $j \in [n-1]$ be such that $\{x + v_j, y + v_j\} \in E_2 \subseteq E$; recall that such a value of j is guaranteed to exist by construction. Then,

$$\begin{aligned} d_Y(f(x), f(y)) &= d_Y(h(x), h(y)) \\ &\leq d_Y(h(x), h(x+v_j)) + d_Y(h(x+v_j), h(y+v_j)) + d_Y(h(y+v_j), h(y)) \\ &\leq r \cdot (D-\delta) \cdot d_Z(x+v_j, y+v_j) + 2 \cdot r \cdot \delta/8D \cdot \gamma \\ &= r \cdot (D-\delta) \cdot d_Z(x, y) + r \cdot \delta/4D \cdot \gamma \\ &= r \cdot (D-\delta) \cdot d_X(x, y) + r \cdot \delta/4D \cdot \gamma \\ &\leq r \cdot (D-\delta) \cdot (1 + \delta/4D) \cdot d_X(x, y); \end{aligned}$$

here, the second line is the triangle inequality; the third line follows from Claim 2.6 and the assumption that h is a G-local $(D-\delta)$-embedding; the fourth line follows since the metric on Z is a norm; and the last line follows from the definition of γ.

Similarly, we have

$$\begin{aligned} d_Y(f(x), f(y)) &= d_Y(h(x), h(y)) \\ &\geq -d_Y(h(x), h(x+v_j)) + d_Y(h(x+v_j), h(y+v_j)) - d_Y(h(y+v_j), h(y)) \\ &\geq r \cdot d_Z(x+v_j, y+v_j) - 2 \cdot r \cdot \delta/8D \cdot \gamma \\ &= r \cdot d_X(x,y) - r \cdot \delta/4D \cdot \gamma \\ &\geq r(1-\delta/4D) \cdot d_X(x,y). \end{aligned}$$

Combining the above two equations shows that the distortion of f is bounded above by $(D-\delta)(1+\delta/4D)(1-\delta/4D)^{-1}$, which is at most D by our assumption $\delta < 1/100$; this gives us the desired contradiction.

Finally, for the "moreover" part, if (X, d_X) is an ℓ_p metric space to start with, then we can view X as a subset of points in ℓ_p^N for $N = \binom{n}{2}$ (see, e.g. [**35**, Proposition 1.4.2]) and repeat the proof above. □

2.1. Graphs of maximum degree two. As remarked after the statement of Theorem 2.5, one cannot prove the theorem with graphs of maximum degree 2. This follows from the below proposition due to Sidhanth Mohanty, who has kindly agreed to let us include its proof here.

PROPOSITION 2.7. *Let $X \subseteq \ell_p$ be a finite set of points. For any graph $G = (X, E)$ of maximum degree at most 2, there exists a G-local isometry (i.e. 1-embedding) $f : X \to \ell_p^2$.*

PROOF. Since G has maximum degree 2, it is a disjoint union of paths and cycles. Therefore, it suffices to prove the proposition in the special case when G is a path and when G is a cycle. When G is a path, say $x_1 \to x_2 \to \cdots \to x_n$, one may even construct a G-local isometry $f : X \to \ell_p^1$ by setting $f(x_1) = 0$, and iteratively, setting $f(x_i) = f(x_{i-1}) + \|x_i - x_{i-1}\|_p$.

Next, consider the case when G is a cycle: $x_1 \to \ldots x_n \to x_1$. We will construct a G-local isometry $f : X \to \ell_p^2$. Initialize by setting $f(x_1) = 0$ and $f(x_n) = \|x_1 - x_n\|_p \cdot e_1$. We

will construct $f(x_2), \ldots, f(x_{n-1})$ iteratively with the property that $\|f(x_i) - f(x_{i-1})\|_p = \|x_i - x_{i-1}\|_p$ and $\|f(x_i) - f(x_n)\|_p = \|x_i - x_n\|_p$ as follows. Let $S(x, r)$ denote the sphere in ℓ_p^2 of radius r, centered at x. Since $\|x_n - x_i\|_p + \|x_{i-1} - x_i\|_p \geq \|x_n - x_{i-1}\|_p$, it follows that $S(x_{i-1}, \|x_{i-1} - x_i\|_p) \cap S(x_n, \|x_n - x_i\|_p) \neq \emptyset$. To complete the iteration, let v_i denote an arbitrary point in this intersection and set $f(x_i) = v_i$. Figure 2.3 illustrates this construction for the case when G is a triangle: $x_1 \to x_2 \to x_3 \to x_1$. □

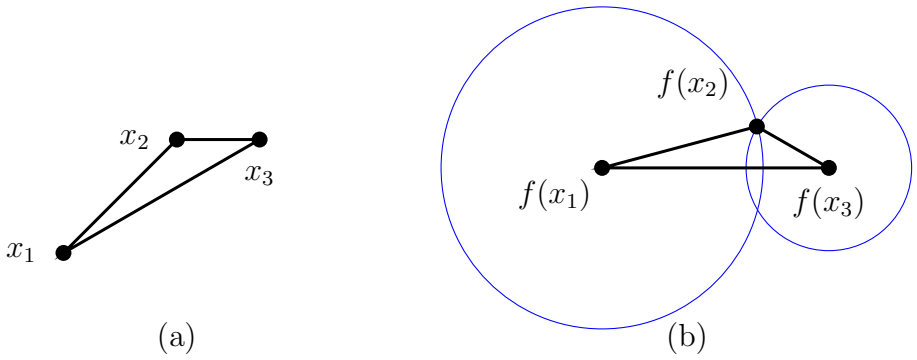

FIGURE 2.3. (a): G is a triangle: $x_1 \to x_2 \to x_3 \to x_1$.
(b): Construction of a G-local isometry to ℓ_p^2.

2.2. Applications. We conclude with two quick applications of Theorem 2.5. First, as discussed in the introduction, it was shown by Abraham, Bartal, and Neiman [**2**, Theorem 1] that any metric space on n points can be embedded into $\ell_p^{O(e^p \log^2 k)}$ with k-local distortion $O(\log k/p)$ (for any $k \leq n$ and $1 \leq p \leq \log k$). We show that for graphically local embeddings, this fails in a strong sense: below, we provide an example of an n point metric space and a graph G of maximum degree 3 such that any G-local embedding of this metric space into ℓ_2 incurs G-local distortion $\Omega(\log n)$ (recall that $O(\log n)$ *global* distortion is always achievable by Bourgain's theorem). We remark that a similar argument works for all ℓ_p spaces with fixed $p \geq 1$.

COROLLARY 2.8. *For any integer $n \geq 2$, there exists a metric space (Z, d_Z) with $|Z| = \Theta(n)$ and a graph $G = (Z, E)$ of maximum degree 3 such that any G-local embedding of (Z, d_Z) into ℓ_2 has G-local distortion $\Omega(\log n)$*

PROOF. It is well known that there exists a metric space (X, d_X) on $\Theta(\sqrt{n})$ points such that any embedding of (X, d_X) into ℓ_2 incurs distortion $\Omega(\log n)$; for instance, one can take (X, d_X) to be the graph metric on a sufficiently good expander with $\Theta(\sqrt{n})$ vertices (see, e.g. [**35**, Theorem 3.5.3]). The statement now follows by applying Theorem 2.5 to this with $D = 2\delta = \Theta(\log n)$. □

Next, we show that that, in general, G-local dimension reduction is no easier than global dimension reduction, even for graphs of maximum degree 3. Previously, such a construction was only known under the additional restriction that the embedding is noncontracting (recall [**38**, Theorem 11] due to Schechtman and Shraibman).

COROLLARY 2.9. *For any integers $n, d \geq 2$, any $\varepsilon \in (\log^{0.51} n/\sqrt{\min(\sqrt{n}, d)}, 1/100)$, there exists a set of points $Z \subseteq \mathbb{R}^d$ and graphs $G = (Z, E)$ of maximum degree 3 such that the following hold:*

- *$|Z| = O(n)$;*
- *the aspect ratio of the Euclidean metric on Z is $O(\log n \cdot \varepsilon^{-2})$;*
- *for any map $f : Z \to \mathbb{R}^m$ satisfying*

$$\|x - y\|^2 \leq \|f(x) - f(y)\|_2 \leq (1 + \varepsilon)\|x - y\|_2 \qquad \forall \{x, y\} \in E,$$

we must have

$$m = \Omega(\varepsilon^{-2} \log(\varepsilon^2 n)).$$

PROOF. Let $n' := \lfloor \sqrt{n} \rfloor$. The main result of Larsen and Nelson [**34**] shows that for any $n', d \geq 2$ and $\varepsilon \in (\log^{0.51} n'/\sqrt{\min(n', d)}, 1/100)$, there exists a set of points $X \subseteq \mathbb{R}^d$ of size n' with aspect ratio $O(\varepsilon^{-1})$ such that for the Euclidean metric space (X, d_X), any embedding $f : (X, d_X) \to \mathbb{R}^m$ with distortion $(1 + 2\varepsilon)$ must satisfy $m = \Omega(\varepsilon^{-2} \log(\varepsilon^2 n))$. The result now follows by applying the "moreover" part of Theorem 2.5 to (X, d_X) and $(\mathbb{R}^m, \|\cdot\|_2)$ with $D = 1 + 2\varepsilon$ and $\delta = \varepsilon$. □

3. Open Problems

Our result leaves open the following questions.

(1) For global dimension reduction, Larsen and Nelson [**34**] are able to provide such a lower bound for every integer n, d and any $\varepsilon \in (\log^{0.51n}/\sqrt{\min(n,d)}, 1/100)$. Is the same lower bound also true for G-local dimension reduction for the entire range of ε? In our result, when $d \gg \sqrt{n}$, the range of ε is more limited than in [**34**].

(2) More interestingly, do there exist lower bound instances with bounded aspect ratio? Or is it the case that for every Euclidean metric space X on n points whose distances are all between 1 and Φ, and for every graph $G = (X, E)$ with maximum degree Δ, for every $\varepsilon > 0$, there exists a G-local embedding of X into ℓ_2^m with distortion $(1+\varepsilon)$ and with $m = f(\varepsilon, \Delta, \Phi)$? Observe that this is true for the special case $\Phi = 1$ (Example 2.3). Also note that, since storing all distances appearing in G to $(1+\varepsilon)$ multiplicative error only requires $O(n\Delta \log(1/\varepsilon) + n\Delta \log\log \Phi)$ bits, the existence of such an embedding cannot be ruled out simply by sketching arguments (for the JL lemma, the tight lower bound can be obtained from lower bounds on sketching, see [**5**]).

Acknowledgements

Vishesh Jain was supported by NSF CAREER award DMS-2237646.

CHAPTER 3

On Sample Reuse Methods for Answering k-wise Statistical Queries

1. Introduction

In this chapter we study the sample complexity of answering M different k-wise statistical queries (SQs). k-wise statistical queries are a generalization of the statistical query model introduced by [**33**] and widely studied thereafter [**9, 10**]. Whereas statistical queries look at the expectation of a function $q : \mathcal{X} \to \mathbb{R}$ from one data point in some domain $\mathcal{X}$ onto the real line, k-wise queries $q : \mathcal{X}^k \to \mathbb{R}$ take a sample of size $k \geq 1$. The importance of being able to answer k-wise queries for larger values of k is illustrated by [**19**], who showed that as k increases, strictly more problems can be solved using k-wise queries.

To answer a sequence of M k-wise statistical queries accurately, the straightforward approach is to randomly split the data into M disjoint parts — one part for each query — and then compose independent k-wise samples within each part. This approach guarantees accuracy since the outputs for different queries are mutually independent, but it is wasteful in the use of data. There are two natural ways to improve beyond the baseline sampling approach:

(1) Given n data points, create n/k (assuming k divides n) mutually independent *pseudo-samples* by drawing k independent and identically distributed (i.i.d.) points from the sample domain at a time. The pseudo-samples are to be reused among the M queries in which case the queries would be considered adaptive.

(2) Given n data points, first split them into M disjoint parts (assuming M divides n) and then create k-wise samples within each part by taking all $\binom{n/M}{k}$ subsets of size k. Under this approach, the queries are considered non-adaptive whereas the k-wise samples used to evaluate each query are dependent.

A line of previous work investigates the sample complexity of answering adaptive SQs ([**25**], [**8**], [**20**], [**21**], etc.) In particular, up to logarithmic factors, [**20**] gives the same sample complexity lower bound as our method (1) for answering unary queries ($k = 1$) and [**8**] gives the same bound as our method (1) for general low sensitivity queries. Despite providing comparable bounds, our work tackles the problem at a different angle than previous work. We focus on the more general k-wise case and find a sample complexity trade-off between the baseline and method (1) depending on the relative values of k, the arity of the query, and M, the total number of queries to be answered. We also study improvement in time complexity as a result of reusing samples when answering low-sensitivity k-wise queries, which is discussed more in detail in Section 4. It is worth noting that since results for non-adaptive queries are already given by VC theory, we are only concerned with the case with adaptive queries for method (1). For method (2), we compare its accuracy in terms of the variance of the output and show that it performs no worse than the baseline non-reuse strategy, and possibly better.

In the remaining parts of the chapter, Section 2 provides rigorous statements of definitions and technical tools, Section 3 introduces the naive sampling approach, Section 4 and Section 5 discuss results of the two sample reuse methods, and Section 6 compares known sampling methods for k-wise SQs in terms of their sample and time complexity.

2. Preliminaries

We first give the definition of a k-wise SQ oracle. In this chapter, we restrict our attention to k-wise statistical queries with binary range, namely functions $q : \mathcal{X}^k \to \{0, 1\}$.

Definition 3.1. (k-wise SQ Oracle [**19**]) Let $\mathcal{D}$ be a distribution over a domain $\mathcal{X}$ and $\tau > 0$. A k-wise statistical query oracle $\mathrm{SQ}_{\mathcal{D}}^{(k)}(q, \tau)$ takes as input any query function $q : \mathcal{X}^k \to \{0, 1\}$ and a value $\tau > 0$ and returns some value v such that

$$\left| v - \mathbb{E}_{X \sim \mathcal{D}^k}[q(X)] \right| \leq \tau$$

.

One main technique we used to study reusing pseudo-samples among adaptive queries is the Transfer Theorem developed in [**8**]. Before introducing it, we first define what it means for an algorithm to be accurate with respect to *a collection of samples* and with respect to a *population*.

DEFINITION 3.2 ((α, β)-Accuracy [**8**]). A mechanism $\mathcal{M}$ is (α, β)-accurate with respect to a set of n samples $X \in \mathcal{X}$ for M adaptively chosen queries $q \in \mathcal{Q}$ if for all sequences of query output $(a_1, \cdots, a_M)$,

$$\mathbb{P}\left[\max_{j\in[M]} \left|a_j - \frac{1}{n}\sum_{i\in[n]} q_j(X_i)\right| \leq \alpha\right] \geq 1 - \beta.$$

A mechanism $\mathcal{M}$ is (α, β)-accurate with respect to the population for M adaptively chosen queries $q \in \mathcal{Q}$ given n samples $X \in \mathcal{X}$ if for all sequences of query output $(a_1, \cdots, a_M)$,

$$\mathbb{P}\left[\max_{j\in[M]} \left|a_j - \mathbb{E}_{X\sim\mathcal{D}}\, q_j(X)\right| \leq \alpha\right] \geq 1 - \beta.$$

We are now ready to state the Transfer Theorem, which shows that any differentially private learner that is accurate on its sample must also generalize to the population from which the sample was drawn. In the original notation of [**8**], the authors use the term "max-KL stability" to refer to the differential privacy model of [**15**], emphasizing it as one of the various notions of stability in machine learning. In Lemma 3.3, we adopt the term "(ε, δ)-differentially private" instead for consistency of notation with the rest of the chapter.

LEMMA 3.3 (Transfer Theorem [**8**]). *Let $\mathcal{Q}$ be a family of queries on $\mathcal{X}^k$ of ℓ_1-sensitivity $\Delta_1(q) = \rho$. Assume that for some $\alpha, \beta \in (0, 0.1)$, an algorithm $\mathcal{A}$ is*

(1) $(\varepsilon' = \alpha/64\rho n,\ \delta' = \alpha\beta/32\rho n)$-differentially private for M adaptively chosen queries from $\mathcal{Q}$ and

(2) $(\alpha' = \alpha/8,\ \beta' = \alpha\beta/16\rho n)$-accurate with respect to its n samples from $\mathcal{X}^k$ for M adaptively chosen queries from $\mathcal{Q}$.

Then $\mathcal{A}$ is (α, β)-accurate with respect to the population for M adaptively chosen queries from $\mathcal{Q}$ given n samples from $\mathcal{X}^k$.

One can achieve the privacy requirement in the Transfer Theorem through the Laplace mechanism as stated in Lemma 1.9.

3. Baseline simulation of an SQ oracle

In this section we discuss the sample complexity of learning with k-wise SQs without sample reuse. An algorithm simulates a k-wise SQ oracle by taking empirical averages. This simulation is extended from the one given by [**33**] for (one-wise) SQ oracles. In the k-wise case, to each query function q_i the learner feeds it a fresh batch of $\tilde{n}$ i.i.d. pseudo-samples $S_i = \{X_1, \cdots, X_{\tilde{n}}\}$, where each pseudo-sample $X_j = (x_{j_1}, \cdots x_{j_k})$ consists of k data points. Then it computes the empirical average of $q_i(X)$ over the set of pseudo-samples S_i. With high probability, the empirical average will fall within the amount of tolerance allowed by the SQ oracle from the true expectation of q_i, thanks to concentration inequalities.

PROPOSITION 3.4. *Let $\mathcal{C}$ be a class of functions. Suppose there exists an SQ learner that makes M k-wise statistical queries of tolerance τ to learn $\mathcal{C}$, then there exists a simulation algorithm, which does not reuse any samples, for which a set of i.i.d. samples of size*

$$n = O\left(k\frac{M}{\tau^2}\log\left(\frac{M}{\delta}\right)\right)$$

is sufficient to PAC learn $\mathcal{C}$ with error bounded by ε and probability of failure bounded by δ.

PROOF. We take the specified number of samples and partition them into

$$\tilde{n} = \frac{n}{Mk} = O\left(\frac{1}{\tau^2}\log\left(\frac{M}{\delta}\right)\right)$$

i.i.d. pseudo-samples for each query function q_i. The Hoeffding inequality (Theorem A.5) guarantees that an empirical average over $\tilde{n}$ pseudo-samples falls within $\pm\tau$ from $\mathbb{E}[q_i(X)]$ with probability $\geq 1 - \frac{\delta}{M}$. Then apply the union bound and we obtain that with probability $\geq 1 - \delta$, the empirical average falls within $\pm\tau$ from the true expectation for all queries. Hence we successfully simulate the k-wise SQ learner with high probability, fulfilling the PAC requirements. □

4. Independent pseudo-samples for adaptive queries

In this section we discuss reusing independent pseudo-samples for adaptive queries. Suppose there exists a k-wise SQ learner that efficiently SQ learns a function class by asking M adaptive k-wise queries $q_1, \cdots, q_M$. Similar to the baseline case, our algorithm (Algorithm 3.1) simulates a k-wise SQ oracle through taking empirical averages. However, what is different from the baseline case is that Algorithm 3.1 partitions the set of n samples $x \sim \mathcal{D}$ into $\tilde{n} = n/k$ parts to create $\tilde{n}$ i.i.d. pseudo-samples $X = (x_1, \cdots, x_k) \sim \mathcal{D}^k$. It then reuses the same set of pseudo-samples among all queries when taking their empirical averages.

Algorithm 3.1 Reusing Independent Pseudo-samples for Adaptive Queries

Input. Data points $x \in \mathcal{X}$ and k-wise Statistical Queries $q_1, \cdots, q_M$, where $q_i : \mathcal{X}^k \to \{0, 1\}$ for all $i \in [M]$.

Output. $v \in \mathbb{R}^M$.

1: Draw $O\Big(\frac{k^2\sqrt{M}}{\tau^2}\log(\frac{Mk}{\tau\delta})\Big)$ i.i.d. data points $x \sim \mathcal{D}$ to create $\tilde{n} = O\Big(\frac{k\sqrt{M}}{\tau^2}\log(\frac{Mk}{\tau\delta})\Big)$ i.i.d. pseudo-samples $X_j = (x_{j_1}, \cdots, x_{j_k}) \sim \mathcal{D}^k$, where $j = 1, \cdots, \tilde{n}$.
2: **for** $i = 1, \cdots, M$ **do**
3: **for** $j = 1, \cdots, \tilde{n}$ **do**
4: $a_{ij} \leftarrow q_i(X_j)$
5: Draw Laplace noise $\xi \sim \text{Laplace}\Big(\frac{128k^2\sqrt{M}}{\tau n}\Big)$.
6: $v_i \leftarrow \Big(\frac{1}{\tilde{n}}\sum_{j=1}^{\tilde{n}} a_{ij}\Big) + \xi$
7: $v \leftarrow (v_1, \cdots, v_M)$

Now we state our main sample complexity result. Theorem 3.5 provides the optimal sample complexity for an algorithm that reuses the same set of independent pseudo-samples while answering adaptive queries.

THEOREM 3.5. *Suppose there exists an SQ learner that makes M k-wise statistical queries of tolerance τ to learn over a class $\mathcal{C}$, then there exists a simulation algorithm, which reuses independent pseudo-samples among the M queries, for which a set of i.i.d. samples of size*

$$\begin{aligned} n &= O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\Big(\max\Big\{M, \frac{k}{\tau}\Big\}\frac{1}{\delta}\Big)\right) \\ &= O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\Big(\frac{Mk}{\tau\delta}\Big)\right) \end{aligned}$$

is sufficient to PAC learn $\mathcal{C}$ with error bounded by ε and probability of failure bounded by δ.

Comparing Theorem 3.5 to the naive bound in Proposition 3.4, we observe an interesting trade-off between arity of the query k, and the total number of queries M. The trade-off suggests that only when a learner uses a large amount of short queries ($k < \sqrt{M}$) is it worth to reuse pseudo-samples.

It is worth noting that Algorithm 3.1 is specific to k-wise statistical queries and it differs from approaches that work for low-sensitivity queries in general. In addition to having low sensitivity, statistical queries and their k-wise generalizations have the additional property that they can be evaluated on k points at a time and are therefore amenable to sampling techniques, which can produce potential speedups (see [**22**]). This allows us to evaluate our queries on pseudo-samples, each of which consists of k sample points.

Corollary 6.1 of [**8**] provides a sample complexity upper bound of $\tilde{O}\left(\frac{\sqrt{M}}{\tau^2}\right)$ for answering unary low-sensitivity queries (i.e., $k = 1$). This result matches the bound achieved by our Algorithm 3.1 suppressing logarithmic factors. Their mechanism runs in time $\text{poly}(n, \log|X|)$ per query, whereas Algorithm 3.1 runs in $\text{poly}(k, \log|X|)$ per query. This can potentially create a dramatic improvement in running time as the straightforward non-sampling approach for exactly evaluating a k-wise query on a sample of n points would be to evaluate it on all k-point subsets, which is indeed polynomial in n but exponential in k. In fact, we go on to analyze that particular approach towards the end of the paper.

In the remaining parts of this section, we first discuss a couple technical tools used to prove Theorem 3.5 and then we give the proof itself.

4.1. Privacy composition. To ensure that the simulation generalizes to the sample distribution, we apply Lemma 3.3 (Transfer Theorem), which demands the algorithm be differentially private. The algorithm composes multiple query functions, so in order to achieve the required level of privacy overall, we need to use results on privacy composition to figure out what level of privacy is required for each individual query.

As discussed in Chapter 1, there are two well-known bounds on the privacy of query composition: simple composition and advanced composition. We shall see that under

appropriate choice of parameters, advanced composition offers tighter privacy bound than simple composition (by a factor of $\sqrt{M}$). The exact statements of the two composition results are provided by Lemma 1.11 and Lemma 1.12.

Theorem 3.5 uses advanced composition of privacy. It is important to mention that advanced composition is necessary when analyzing pseudo-sample reuse. Since the algorithm uses adaptive queries, it needs to be strict when budgeting the privacy level for each query. Otherwise, an excess amount of Laplace noise would need to be added, which will overturn the effect of sample reuse. As shown in Theorem 3.6, if the algorithm composed privacy of the queries as if they were independent, the resulting sample complexity is actually worse than the baseline bound.

THEOREM 3.6. *Under the setting of Theorem 3.5, except that suppose the simulation algorithm treats the M queries as if they were independent and calculates their overall privacy through simple composition, a set of i.i.d. samples of size*

$$\begin{aligned} n &= O\left(\frac{k^2 M}{\tau^2}\log\left(\max\left\{M, \frac{k}{\tau}\right\}\frac{1}{\delta}\right)\right) \\ &= O\left(\frac{k^2 M}{\tau^2}\log\left(\frac{Mk}{\tau\delta}\right)\right) \end{aligned}$$

is sufficient to PAC learn $\mathcal{C}$ with error bounded by ε and probability of failure bounded by δ.

We omit the proof for Theorem 3.6 since it closely resembles that of Theorem 3.5, with the only difference being the privacy composition calculations.

4.2. Laplace mechanism. Now that we know to use advanced composition, let us consider how to achieve the desired level of privacy for each query function. As suggested by Lemma 1.9, we adopt Laplace mechanism, the standard technique that offers privacy guarantee for algorithms.

For each q_i, Algorithm 3.1 outputs $v_i = a_i + \xi$, where a_i is the empirical average of q_i over a large set of pseudo-samples and ξ is a the Laplace noise. There are two key considerations when choosing the parameters. First, the sample set needs to be large enough so that the

empirical average is close to the true expectation with high probability. Second, the Laplace noise needs to be small enough so that it does not steer the empirical average away from the expected average too far, but in the meantime it is still large enough to maintain privacy.

By Lemma 1.9, we choose $\xi \sim \text{Laplace}(\rho \cdot \frac{128k}{\tau})$, which preserves $(\frac{\tau}{128k}, 0)$-differential privacy for each query, surpassing the requirement of the Transfer Theorem (Lemma 3.3). Here ρ is the ℓ_1-sensitivity of the empirical average of q over all pseudo-samples.

PROPOSITION 3.7. *The ℓ_1-sensitivity of the empirical average of $q : \mathcal{X}^k \to \{0,1\}$ is $\rho = \Delta_1(q) \leq \frac{k}{n}$.*

PROOF. Among all pseudo-samples $X_i \in S$ and $X'_i \in S'$ where $i = 1, \cdots, \tilde{n}$, exactly one pair is different $X_j \neq X'_j$. Then $|q(X_i) - q(X'_i)| = 0$ for all $i \neq j$, while $|q(X_j) - q(X'_j)| \leq 1$. Therefore,

$$\rho = \Delta_1(q) = \max_{\substack{S,S' \subseteq \mathcal{X}^k \\ s.t.\ \|S - S'\|_1 = 1}} \left\| \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \Big(q(X_i) - q(X'_i) \Big) \right\|_1,$$

which can trivially be bounded as

$$\rho \leq 1/\tilde{n} = k/n.$$

□

4.3. Proof of Theorem 3.5. Given an efficient k-wise SQ learner that learns $\mathcal{C}$ approximately correct (to an error ε), the empirical average simulation wishes to mimic the learner's query outputs with high probability. In the language of the Transfer Theorem (Lemma 3.3), that is to say the simulator needs to be (τ, δ)-accurate with respect to the population. We prove Theorem 3.5 using the Transfer Theorem.

PROOF OF THEOREM 3.5. Given the total allowed error of τ, we allocate $\tau/2$ to the empirical average and $\tau/2$ to the added Laplace noise. We first analyze the empirical average. To achieve $(\tau/2, \delta)$-accuracy with respect to the population for M adaptively chosen queries, the Transfer Theorem demands

(i) the simulation is $\left(\frac{\tau}{128k}, \frac{\tau\delta}{64k}\right)$-differentially private for M adaptive queries,

(ii) the simulation is $\left(\frac{\tau}{16}, \frac{\tau\delta}{32k}\right)$-accurate with respect to n samples for M adaptive queries.

To satisfy (i), we adopt advanced composition. According to Lemma 1.12, each of the M queries needs to be $\left(\frac{\tau}{128k\sqrt{M}}, \frac{\tau\delta}{64kM}\right)$-differentially private to obtain the composed privacy stated in (i). We know each query has ℓ_1-sensitivity $\rho = k/n$ through Lemma 3.7. Then following the standard technique stated in Lemma 1.9, we add Laplace noise of scale $\frac{128k^2\sqrt{M}}{\tau n}$ to each query average, which achieves $\left(\frac{\tau}{128k\sqrt{M}}, 0\right)$-differential privacy, surpassing the needed amount. Lemma 1.10 verifies that the added Laplace noise is bounded above by $\frac{128k^2\sqrt{M}}{\tau n}\log\frac{2M}{\delta}$ with probability $\geq 1 - \frac{\delta}{2M}$. In order to restrict the Laplace noise within $\tau/2$ with high probability, we ask that

$$\frac{128k^2\sqrt{M}}{\tau n}\log\frac{2M}{\delta} \leq \frac{\tau}{2},$$

which implies that

$$n = O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\frac{M}{\delta}\right) \tag{4}$$

is sufficient. Now let us consider (ii). It suffices to show that for all queries q_i, the simulator's output a_i satisfies

$$\mathbb{P}\left[\left|a_i - \frac{1}{\tilde{n}}\sum_{j=1}^{\tilde{n}} q_i(X_j)\right| \leq \frac{\tau}{16}\right] \geq 1 - \frac{\tau\delta}{32k}.$$

We know that

$$a_i = \frac{1}{\tilde{n}}\sum_{j=1}^{\tilde{n}} q_i(X_j) + \text{Laplace}\left(\frac{128k\sqrt{M}}{\tau\tilde{n}}\right),$$

so we just need

$$\mathbb{P}\left[\left|\text{Laplace}\left(\frac{128k\sqrt{M}}{\tau\tilde{n}}\right)\right| \leq \frac{\tau}{16}\right] \geq 1 - \frac{\tau\delta}{32k}.$$

According to Lemma 1.10, it is easy to verify that with probability $\geq 1 - \frac{\tau\delta}{32k}$, the Laplace noise of scale $\frac{128k\sqrt{M}}{\tau\tilde{n}}$ is $O(\frac{k^2\sqrt{M}}{\tau n}\log\frac{k}{\tau\delta})$. To satisfy (ii), we ask that $\frac{128k^2\sqrt{M}}{\tau n}\log\frac{32k}{\tau\delta} \leq \frac{\tau}{16}$,

which implies an

$$n = O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\frac{k}{\tau\delta}\right) \tag{5}$$

is sufficient. Combining inequalities (4) and (5), we get

$$\begin{aligned} n =& O\left(\max\left\{\frac{k^2\sqrt{M}}{\tau^2}\log\frac{M}{\delta}, \frac{k^2\sqrt{M}}{\tau^2}\log\frac{k}{\tau\delta}\right\}\right) \\ =& O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\left(\max\left\{M, \frac{k}{\tau}\right\}\frac{1}{\delta}\right)\right) \\ =& O\left(\frac{k^2\sqrt{M}}{\tau^2}\log\left(\frac{Mk}{\tau\delta}\right)\right), \end{aligned}$$

completing the proof. □

5. Dependent k-wise samples for non-adaptive queries

Now we examine the second reuse method. Algorithm 3.2 draws n i.i.d. sample points $X \sim \mathcal{D}$ and partitions them into M equal parts, $S_1, \cdots, S_M$, to be used by M queries. Denote the size of each part $|S_i| = \hat{n}$, and the total number of samples is $n = M\hat{n}$. For each query, the algorithm calculates its empirical average over $\binom{\hat{n}}{k}$ k-wise samples, which are generated by taking all size k subsets of S_i.

Algorithm 3.2 Dependent k-wise Samples for Non-adaptive Queries

Input. Data points $x \in \mathcal{X}$ and k-wise Statistical Queries $q_i : \mathcal{X}^k \to \{0, 1\}$, where $i = 1, \cdots, M$.

Output. $v = (v_1, \cdots, v_M) \in \mathbb{R}^M$.

1: Draw n i.i.d. sample points $x \sim \mathcal{D}$ and partition them into M equal parts $S_1, \cdots, S_M$, where $|S_i| = \hat{n}$.

2: **for** $i = 1, \cdots, M$ **do**

3: Take all size k subsets of S_i to create k-wise samples $X_j = (x_{j_1}, \cdots, x_{j_k})$, where $j = 1, \cdots, \binom{\hat{n}}{k}$.

4: Compute the empirical average of q_i

$$v_i \leftarrow \frac{1}{\binom{\hat{n}}{k}}\sum_{j=1}^{\binom{\hat{n}}{k}} q_i(\mathbf{x}_j).$$

5: $v \leftarrow (v_1, \cdots, v_M)$.

In contrast to creating independent pseudo-samples, Algorithm 3.2 uses all k-subsets of the provided sample set, yielding additional k-wise samples, although it fails to maintain their independence since each point contributes to $(k-1)$ samples.

We can analyze the dependent k-wise samples from the perspective of a hypergraph. In the language of hypergraphs, we can think of each sample point as a vertex and each k-wise sample as a k-hyperedge. The learner is given K_n^k, a complete hypergraph on n vertices, whose hyperedges contain k vertices (assuming k divides n). The learner uses k-hyperedges as inputs to the queries. Notice that the hyperedges are not independent with each other. Fortunately, we can bypass the hyperedge dependence through Baranyai's Theorem (Theorem A.4), which says every K_n^k decomposes into a disjoint collection of 1-factors. Recall that a 1-factor is a set of hyperedges that touch each vertex in K_n^k exactly once. Intuitively, we can think of a 1-factor as a perfect matching. With the guarantee of decomposition given by Baranyai's Theorem, we are able to interpret the collection of dependent hyperedges as a set of perfect matchings. Although these matchings are dependent on one another, they each contain independent hyperedges within themselves. Figure 3.1 gives an example of when $n = 6$ and $k = 2$. As shown by Figure 3.1, K_6^2 can be decomposed into a disjoint union of 1-factors, each of which consists of three mutually independent edges.

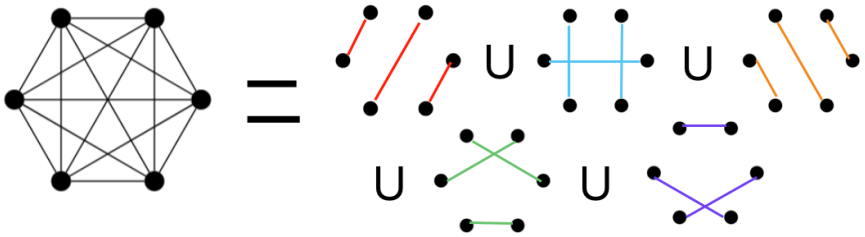

FIGURE 3.1. An illustration of a decomposition of K_6^2 into five disjoint perfect matchings.

How well do dependent k-wise samples perform when we use them to estimate the expected value through empirical average? In each 1-factor, the independent hyperedges act like pseudo-samples introduced in Algorithm 3.1. Accordingly, in Theorem 3.8 we provide accuracy bounds of dependent k-wise samples by comparing the variance to that of the baseline method without sample reuse.

To set up Theorem 3.8, let $q : \mathcal{X}^k \to \{0, 1\}$ be a k-wise statistical query, S be a set of samples $x \sim \mathcal{D}$, and suppose $|S| = n$, where k divides n. Let Y_a, Y_b be random variables that represent the empirical average of q under the two sampling schemes respectively: taking all $\binom{n}{k}$ k-subsets of S and the baseline method of creating n/k independent pseudo-samples. The expected value of Y_a and Y_b both equal to $\mathbb{E}[q]$.

THEOREM 3.8. *The variance of Y_a and Y_b satisfy*

$$\frac{1}{\binom{n-1}{k-1}}\mathrm{Var}(Y_b) \leq \mathrm{Var}(Y_a) \leq \mathrm{Var}(Y_b).$$

PROOF. We first study the upper bound. Construct a complete hypergraph K_n^k with the given n sample points. With guarantee from Baranyai's theorem, we can decompose K_n^k into 1-factors $G_1, \cdots, G_m$, where $m = \binom{n-1}{k-1}$. Each G_i contains n/k i.i.d. hyperedges of length k. The vertices in these i.i.d. hyperedges form i.i.d. pseudo-samples used in Algorithm 3.1. Let Y_{G_i} be a random variable that represents the empirical average of ϕ over pseudo-samples in G_i. By previous analysis, we know for all $i = 1, \cdots, \binom{n-1}{k-1}$,

$$\mathrm{Var}(Y_b) = \mathrm{Var}(Y_{G_i}).$$

Observe that taking an empirical average over all $\binom{n}{k}$ hyperedges in K_n^k is equivalent to taking an average of all the empirical averages over $G_1, \cdots, G_m$. Therefore,

$$\mathrm{Var}(Y_a) = \mathrm{Var}\left(\frac{1}{m}\sum_{i=1}^{m} Y_{G_i}\right).$$

Since Y_{G_i}'s are identically distributed, we can denote $\mathrm{Var}(Y_{G_i}) = \sigma^2$ for all $i = 1, \cdots, m$. Then we can prove the upper bound

$$\begin{aligned}
\mathrm{Var}(Y_a) &= \mathrm{Var}\left(\frac{1}{m}\sum_{i=1}^{m} Y_{G_i}\right) \\
&= \frac{1}{m^2}\left(\sum_i \mathrm{Var}(Y_{G_i}) + \sum_{i\neq j}\mathrm{Cov}(Y_{G_i}, Y_{G_j})\right) \\
&\leq \frac{1}{m^2}\Big(m\sigma^2 + (m^2-m)\sqrt{\sigma^2\sigma^2}\Big) \\
&= \sigma^2.
\end{aligned}$$

The inequality uses the well-known fact that for any two random variables X_i, X_j,

$$\mathrm{Cov}(X_i, X_j) \leq \sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_j)}.$$

The lower bound follows similar reasoning.

$$\begin{aligned}
\mathrm{Var}(Y_a) &= \frac{1}{m^2}\left(\sum_i \mathrm{Var}(Y_{G_i}) + \sum_{i\neq j}\mathrm{Cov}(Y_{G_i}, Y_{G_j})\right) \\
&\geq \frac{1}{m^2}\sum_i^m \sigma^2 \\
&= \frac{\sigma^2}{m}.
\end{aligned}$$

The inequality relies on the fact that $\mathrm{Cov}(Y_{G_i}, Y_{G_j}) \geq 0$ for all $i \neq j$. To prove this fact, we first decompose the covariance as follows, which uses the fact that hyperedges $e \in G_i$ and $f \in G_j$ are i.i.d. within each graph.

$$\begin{aligned}
\mathrm{Cov}(Y_{G_i}, Y_{G_j}) &= \mathrm{Cov}\left(\frac{1}{n/k}\sum_{e\in G_i}\phi(e),\ \frac{1}{n/k}\sum_{f\in G_j}\phi(f)\right) \\
&= (\frac{1}{n/k})^2 \sum_{e\in G_i}\sum_{f\in G_j}\mathrm{Cov}\Big(\phi(e),\ \phi(f)\Big).
\end{aligned}$$

It suffices to show that $\mathrm{Cov}(\phi(e), \phi(f)) \geq 0$ for all $e \in G_i$ and $f \in G_j$. There are three scenarios to consider.

(1) e and f are disjoint. In this case, e and f are independent, so

$$\mathrm{Cov}(\phi(e),\ \phi(f)) = 0.$$

(2) e and f are identical; that is, they use the same set of vertices. Then,

$$\mathrm{Cov}(\phi(e),\ \phi(f)) = \mathrm{Cov}(\phi(e),\ \phi(e)) = \mathrm{Var}(\phi(e)) \geq 0.$$

(3) e and f have a non-empty intersection. Denote $S := e \cap f$. For ease of notation, let $U_e := \phi(e)$ and $U_f := \phi(f)$ be the random variables that represent the query values and let X_S be the random variable that represents the selection of S. Then, by the law of total covariance,

$$\mathrm{Cov}(U_e, U_f) = \mathbb{E}(\mathrm{Cov}(U_e,\ U_f|X_S)) + \mathrm{Cov}(\mathbb{E}(U_e|X_S), \mathbb{E}(U_f|X_S)).$$

Since $e \setminus S$ and $f \setminus S$ are disjoint, they are independent and we have

$$\mathrm{Cov}(U_e,\ U_f|X_S) = 0,$$

which implies

$$\mathbb{E}(\mathrm{Cov}(U_e,\ U_f|X_S)) = 0.$$

Also, since $e \setminus S$ and $f \setminus S$ are identically distributed, we know $\mathbb{E}(U_e|X_S) = \mathbb{E}(U_f|X_S)$, which implies

$$\mathrm{Cov}(\mathbb{E}(U_e|X_S), \mathbb{E}(U_f|X_S)) = \mathrm{Cov}(\mathbb{E}(U_e|X_S), \mathbb{E}(U_e|X_S)) = \mathrm{Var}(\mathbb{E}(U_e|X_S)) \geq 0.$$

Hence, in this scenario, we still have $\mathrm{Cov}(U_e, U_f) \geq 0$.

□

The bounds provided in Theorem 3.8 are tight, which can be shown by examples. First we study the lower bound. Draw a set S of n points (assuming k divides n) uniformly from $\{0, 1\}$. The probability density function of the Uniform distribution is given by A.1. q is a k-wise

statistical query which evaluates the expected parity of a size k subset of S. Specifically, if a subset X_i has even number of 1's, then $q(X_i) = 0$, otherwise $q(X_i) = 1$. Under this setting, the two sampling schemes correspond to the following random variables:

$$Y_a = \frac{1}{\binom{n}{k}} \sum_{i=1}^{\binom{n}{k}} q(X_i) = \frac{1}{\binom{n}{k}} \sum_{i=1}^{\binom{n}{k}} Y_i,$$

$$Y_b = \frac{1}{n/k} \sum_{j=1}^{n/k} q(X_j) = \frac{1}{n/k} \sum_{j=1}^{n/k} Y_j.$$

Here Y_i, Y_j are introduced for ease of notation, representing q evaluated on subsets X_i and X_j respectively. We can show such Y_a, Y_b achieve the lower bound in Theorem 3.8.

$$\begin{aligned}
\mathrm{Var}(Y_a) &= \mathrm{Var}\left(\frac{1}{\binom{n}{k}} \sum_{i=1}^{\binom{n}{k}} Y_i \right) \\
&= \left(\frac{1}{\binom{n}{k}} \right)^2 \cdot \mathrm{Var}\left(\sum_{i=1}^{\binom{n}{k}} Y_i \right) \\
&= \left(\frac{1}{\binom{n}{k}} \right)^2 \left(\sum_{i=1}^{\binom{n}{k}} \mathrm{Var}(Y_i) + 2 \sum_{i<j} \mathrm{Cov}(Y_i, Y_j) \right) \\
&= \left(\frac{1}{\binom{n}{k}} \right)^2 \sum_{i=1}^{\binom{n}{k}} \mathrm{Var}(Y_i) \\
&= \frac{1}{\binom{n-1}{k-1}} \cdot \frac{1}{(n/k)^2} \sum_{j=1}^{n/k} \mathrm{Var}(Y_j) \\
&= \frac{1}{\binom{n-1}{k-1}} \cdot \sum_{j=1}^{n/k} \frac{1}{(n/k)^2} \mathrm{Var}(Y_j) \\
&= \frac{1}{\binom{n-1}{k-1}} \cdot \mathrm{Var}\left(\frac{1}{n/k} \sum_{j=1}^{n/k} Y_j \right) \\
&= \frac{1}{\binom{n-1}{k-1}} \mathrm{Var}(Y_b).
\end{aligned}$$

In the above calculations, the crucial step is the fourth equation which uses the fact that for all $i \neq j$,

$$\mathrm{Cov}(Y_i, Y_j) = 0. \tag{6}$$

To prove Equation (6), first recall that for any two random variables Y_i and Y_j,

$$\mathrm{Cov}(Y_i, Y_j) = \mathbb{E}(Y_i Y_j) - \mathbb{E}(Y_i)\mathbb{E}(Y_j).$$

We know $\mathbb{E}(Y_i) = \frac{1}{2}$ for all $i = 1, \cdots, \binom{n}{k}$, so for Equation (6) to be true, we need to show that $\mathbb{E}(Y_i Y_j) = \frac{1}{4}$ for all $i, j \in [\binom{n}{k}]$. When Y_i and Y_j are independent, this is obviously true, since

$$\mathbb{E}(Y_i Y_j) = \mathbb{P}(Y_i = 1)\mathbb{P}(Y_j = 1) \cdot 1 \cdot 1 = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$

When Y_i and Y_j are dependent, it means subsets X_i and X_j has a non-empty intersection U. Notice that no matter the parity of U, the parity of the remainder of the two sets, $X_i \setminus U$ and $X_j \setminus U$, is evenly distributed and the two sets are mutually independent. Therefore, when Y_i and Y_j are dependent, we still have $\mathbb{E}(Y_i Y_j) = \frac{1}{4}$. We have hence successfully found a tight example for the lower bound in Theorem 3.8.

Now we consider the following example which achieves the upper bound in Theorem 3.8. We draw a set S of n points (assuming k divides n) uniformly from $\{0, 1\}$. For a size k subset $X_i \subset S$, define the k-wise statistical query ϕ to count the number of 1's contained in X_i. In this setting, Y_b goes through the set S once, counts how many points in S equal to 1 and then divides the counted value by n/k. Y_a goes through the same set $\binom{n-1}{k-1}$ times, each time counting how many points in S equal to 1, which is the same every time. Then to calculate Y_a, we sum up the counted values from all $\binom{n-1}{k-1}$ times and divide it by $\frac{n}{k} \cdot \binom{n-1}{k-1}$. Hence, Y_a and Y_b are equivalent random variables and have the same variance.

6. Comparison of known sampling methods

Throughout this paper, we discussed two sample reuse methods when working with k-wise statistical queries: (1) reusing independent pseudo-samples for each query and (2) reusing individual sample points to create dependent k-wise samples. The baseline approach, as

stated in Proposition 3.4, does not reuse any samples, so it uses neither of the two methods. Yet there are known algorithms that use either one of the methods or both at the same time. Namely, Algorithm 3.1 uses method (1), Algorithm 3.2 uses method (2), and the result in [**8**] for adaptive low sensitivity queries combines both methods: it creates dependent samples by taking all possible subsets of the given points and it reuses the same set of samples to answer all queries.

We have some idea of how the performances of the four known methods compare.

(1) To compare Algorithm 3.1 and the baseline, Proposition 3.4 and Theorem 3.5 tell us about the trade-off between k, the arity of the query and M the number of queries to be answered; such trade-off is a result of the added Laplace noise, which is necessary when working with adaptive queries.

(2) Algorithm 3.1 achieves the same sample complexity upper bound as the approach in [**8**], but benefits in that the running time is polynomial in k, the arity of the query and not n, the sample size.

7. Open problems

A natural question to ask is how does Algorithm 3.2 compare to the baseline in terms of sample complexity. Theorem 3.8 goes toward answering this question by providing variance bounds but fails at giving explicit sample bounds, which is a technically difficult task. In Algorithm 3.2, the k-wise samples used by each query are not mutually independent, so most classical concentration inequalities cannot be applied to obtain an explicit error bound for the query's empirical average. From our analysis, since the variance bounds are shown to be tight, we expect Algorithm 3.2 to have the same sample complexity upper bound as the baseline. Moreover, we expect the trade-off between k and M to carry over when comparing sample complexity of Algorithm 3.2 with the approach in [**8**].

Acknowledgments

This work was supported in part by National Science Foundation grant ECCS-2217023. We thank an anonymous reviewer of this paper for detailed and helpful comments.

Data availability

We do not generate or make use of any datasets, due to the fact that this work focuses on theoretical and mathematical analysis.

Conflict of interest

The authors declare that they have no conflict of interest.

CHAPTER 4

An Efficient Algorithm for Generating Private Synthetic Data

1. Introduction

Differential privacy provides a principled framework for releasing information about sensitive datasets while limiting disclosure risk to individuals [**14**]. A central problem in this area is *private query release*: given a database $X \in \mathcal{X}^n$ and a class of queries $\mathcal{Q}$, one seeks to answer the queries $q(X)$ accurately while satisfying (ε, δ)-differential privacy [**11**]. One appealing approach is to release a *synthetic database* $\hat{X}$ whose query answers approximate those of the real database, enabling unrestricted post-processing without further privacy loss [**24**].

From a computational perspective, private synthetic data generation involves inherent tradeoffs. In the worst case, producing a database that is accurate for an arbitrary query class is computationally intractable [**41**]. As a result, existing polynomial-time methods typically apply only to restricted families of queries such as counting queries [**11**], while more general approaches often rely on oracle-efficient frameworks that invoke computationally hard optimization subroutines in the worst case [**44, 17**]. This motivates the study of intermediate regimes that admit fully explicit algorithms while relaxing accuracy guarantees in a controlled manner.

In this chapter, we develop an *efficient private synopsis generator*: an explicit, polynomial-time algorithm that produces a synthetic database intended to answer a pre-specified query set $\mathcal{Q}$ under differential privacy. Our approach focuses on smooth query families and avoids oracle assumptions entirely. In the general case, rather than aiming to satisfy all queries simultaneously, the algorithm can be used to produce a *base synopsis*—a synthetic database that answers a large fraction of queries accurately with respect to any fixed distribution over

$\mathcal{Q}$. This notion, introduced by [**17**], aligns naturally with private boosting frameworks, which can subsequently be applied to strengthen accuracy guarantees when desired.

Our algorithm proceeds in two stages. First, it releases noisy query answers using the Laplace mechanism together with standard composition guarantees. Second, it constructs a synthetic database by fitting these noisy answers via an explicit optimization procedure. Crucially, this second stage operates solely on privatized information and therefore incurs no additional privacy loss.

A central design choice in this optimization step is the selection of a surrogate loss function. Our goal is not to minimize average error, but rather to drive as many queries as possible below a fixed accuracy threshold. To this end, we adopt a smooth symmetric exponential surrogate that strongly penalizes large query deviations while assigning relatively little weight to already well-approximated queries. (See [**26**] for more on surragate losses.) This loss functions as a potential for the threshold-based accuracy objective underlying the base-synopsis definition and integrates naturally with second-order optimization methods.

We emphasize that, as with squared error and other symmetric losses, this surrogate does not preserve convexity when composed with general nonlinear queries. Accordingly, our approach does not rely on global convexity guarantees or convergence to a global optimum. Instead, the algorithm terminates as soon as the base-synopsis criterion is satisfied, and any such iterate constitutes a valid output. Empirically, we find that the resulting optimization landscape is well behaved for the query families considered, enabling efficient optimization in practice. We also note that our method requires the queries to be twice continuously differentiable, and thus does not apply directly to many standard tabular data queries such as marginal and counting queries. In order to adapt our algorithm to these nonsmooth queries, we may make estimates to their derivatives, which will result in a more relaxed accuracy guarantee.

We analyze the privacy, accuracy, and runtime properties of the resulting algorithm and show that it runs in time polynomial in the database size and number of queries,conditioned on that the base synopsis can be obtained efficiently. We further demonstrate how the

output can be combined with existing private boosting techniques to obtain stronger accuracy guarantees. Finally, we present experimental results illustrating the practical behavior of the method. For smooth convex query families, such as collections of ℓ_p-norms, the algorithm reliably produces accurate base synopses. We also consider structured nonconvex query classes, where we observe that modest algorithmic refinements yield accurate synthetic data in practice.

2. Preliminaries

In this section, we give the notation and introduce some of the technical tools that are useful in our analysis.

Given a data domain $\mathcal{X}$ and a query set $\mathcal{Q}$, let $X \in \mathcal{X}^n$ be a database of size n and let $q \in \mathcal{Q}$ be a function $q : \mathcal{X}^n \to \mathbb{R}$. We seek to approximate answers to the queries $q \in \mathcal{Q}$ evaluated on the real database X. We will assume the queries have ℓ_1-sensitivity bounded by a parameter ρ, as defined in Definition 1.6.

Whenever gradients or Hessians are used, we implicitly assume that the corresponding query admits a smooth extension to a neighborhood of $\mathcal{X}^n$, and all derivatives are taken with respect to this extension.

Our efficient synthetic data generator (Algorithm 4.1) outputs a synthetic database $\hat{X}$ that, with high probability, answers a majority of the queries in $\mathcal{Q}$ accurately. Algorithm 4.1 is a (λ, η, β)-*base synopsis generator* as defined by [**17**]. The term "base synopsis" comes from the traditional boosting algorithms such as AdaBoost [**23**]; $\hat{X}$ can be viewed as a weak learner for the real database X, with an edge $\eta \in (0, 1/2]$ over random guessing.

Definition 4.1 (Base Synopsis Generator [**17**]). Given a data universe $\mathcal{X}$ and a query set $\mathcal{Q}$ along with its database $X \in \mathcal{X}^n$, an algorithm $\mathcal{M}$ is a (λ, η, β)-base synopsis generator if for any distribution $\mathcal{D}$ on $\mathcal{Q}$, with all but β probability over the randomness of $\mathcal{M}$, the synopsis S that $\mathcal{M}$ outputs is λ-accurate for at least $(1/2 + \eta)$-fraction of the mass of $\mathcal{Q}$ as weighted by $\mathcal{D}$, that is,

$$\mathbb{P}_{q \sim \mathcal{D}}[|q(S) - q(X)| \leq \lambda] \geq \tfrac{1}{2} + \eta.$$

Such a synopsis $\mathcal{S}$ is called a (λ, η, β)-base synopsis.

The goal for our work is to construct a synthetic dataset that maintains the privacy of the original data. To this end, our synthetic data generator will also be (ε, δ)-differentially private with respect to the dataset X, under the standard notion of differential privacy, given in Chapter 1 Section 3.

3. The Efficient Private Synopsis Generator

In this section, we describe our efficient private synopsis generator and the objective used to construct a synthetic database from noisy query answers. Throughout, we assume that each query $q \in \mathcal{Q}$ is twice continuously differentiable. We do *not* assume that the resulting optimization problem is globally convex or can provably be solved efficiently. Instead, our goal is to design a smooth surrogate objective that is well aligned with the threshold-style accuracy guarantees required for a base synopsis and that admits efficient optimization in practice.

3.1. Overview of the Approach. Given a real database $X \in \mathcal{X}^n$ and a finite query set $\mathcal{Q}$, our objective is to produce a synthetic database $\hat{X}$ that approximately matches the answers $q(X)$ for a large fraction of queries $q \in \mathcal{Q}$, while satisfying differential privacy.

The algorithm proceeds in two stages. First, for each query $q \in \mathcal{Q}$, we release a noisy answer $\tilde{q}(X) = q(X) + \xi_q$, where ξ_q is drawn from a suitably calibrated Laplace distribution. By standard composition results, this stage satisfies (ε, δ)-differential privacy for the collection of released answers. Then, using the noisy answers $\{\tilde{q}(X)\}_{q \in \mathcal{Q}}$ as targets, we construct a synthetic database $\hat{X}$ by minimizing a smooth surrogate loss that penalizes large deviations $|q(\hat{X}) - \tilde{q}(X)|$. This optimization step involves no further access to the real data and therefore incurs no additional privacy loss.

The quality of the resulting synthetic database is evaluated using a threshold-based criterion: for a fixed tolerance parameter $\lambda > 0$, we require that a $(1/2 + \eta)$-fraction of the

queries (with respect to a fixed distribution $\mathcal{D}$ over $\mathcal{Q}$) satisfy

$$|q(\hat{X}) - q(X)| \leq \lambda.$$

3.2. Surrogate Loss Design. A direct approach to enforcing the above criterion would be to minimize a 0–1 loss of the form $\mathbf{1}_{\{|q(\hat{X})-\tilde{q}(X)|>\lambda\}}$, which counts the number of queries whose error exceeds the tolerance λ. However, this loss is discontinuous and unsuitable for gradient-based optimization.

Instead, we introduce a smooth symmetric exponential surrogate. For a single query $q \in \mathcal{Q}$, define

$$\ell_q(\hat{X}) = \exp\Big(\tfrac{q(\hat{X})-\tilde{q}(X)}{\lambda} - 1\Big) + \exp\Big(-\tfrac{q(\hat{X})-\tilde{q}(X)}{\lambda} - 1\Big),$$

and define the aggregate loss

$$\mathcal{L}_{\pm\exp}(\hat{X}; \mathcal{Q}) = \sum_{q\in\mathcal{Q}} \ell_q(\hat{X}). \tag{7}$$

This surrogate has several properties that make it well suited to our setting:

- Smoothness: The loss is infinitely differentiable, enabling the use of second-order optimization methods.
- Alignment with threshold accuracy: The loss grows rapidly once $|q(\hat{X}) - \tilde{q}(X)|$ exceeds λ, while assigning comparatively smaller weight to already well-approximated queries. As a result, optimization effort is concentrated on correcting the largest query deviations, which is consistent with the goal of producing a base synopsis that satisfies a majority of queries within tolerance.
- Potential-function behavior: The exponential form provides a natural potential function for reasoning about the fraction of queries with large error, similar in spirit to exponential potentials used in boosting and multiplicative-weights methods.

We emphasize that, although the surrogate loss is convex as a function of the scalar residual $q(\hat{X}) - \tilde{q}(X)$, it is *not* guaranteed to be convex as a function of the synthetic database $\hat{X}$ when the queries q are nonlinear. This limitation is unavoidable for symmetric two-sided

losses composed with general nonlinear queries and also applies to standard alternatives such as squared error [**13**, Section 3.2.4]. Our approach therefore does not rely on global convexity guarantees. In particular, our algorithm does not seek a global minimizer of $\mathcal{L}_{\pm\exp}$. Instead, it terminates as soon as the current synthetic database satisfies the base-synopsis criterion, and any such iterate constitutes a valid output.

We note that alternative *tube-style* losses—i.e., objectives that impose little or no penalty when the residual lies within a tolerance band and increase the penalty outside this band—are well studied in robust regression, most notably the ε-insensitive and Huberized losses used in support vector regression [**28, 39, 43**]. These could also be employed; however, the smooth symmetric exponential surrogate used here provides stronger gradients for large violations and integrates naturally with our fraction-of-queries accuracy criterion, making it particularly well suited to the base-synopsis setting considered in this work.

3.3. Our Algorithm. There are two main stages to the algorithm and we use the parameter $\theta \in (0, 1)$ to split the accuracy budget between the two stages.

(1) For each query $q \in \mathcal{Q}$, we compute the output $q(X)$ evaluated on the real database, and then modify it by adding Laplace noise $\xi \sim \text{Laplace}\Big(\rho\sqrt{2k\ln(1/\delta)}/\varepsilon\Big)$. This guarantees $\big(\varepsilon/\sqrt{2k\ln(1/\delta)}, 0\big)$-differential privacy for each query output, which by known results on privacy composition (Lemma 1.12) amounts to $(\varepsilon, 0)$-differential privacy for the entire set of queries $\mathcal{Q}$. Using the probability tail bound known for the Laplace distribution (Lemma 1.10), we make sure that for all $q \in \mathcal{Q}$, the added Laplace noise
$$\left|\text{Laplace}\left(\frac{\rho\sqrt{2k\ln(1/\delta)}}{\varepsilon}\right)\right| \leq \theta\lambda$$
with probability at least $1 - \beta/k$.

(2) Let $X^0 \in \mathcal{X}^n$ be an arbitrary initial guess for the database. Using the surrogate loss $\mathcal{L}_{\pm\exp}$ (7) as the objective, we run Algorithm 4.2 until $(1/2 + \eta)$-fraction of the queries $q \in \mathcal{Q}$, weighted by a fixed distribution $\mathcal{D}$, satisfy $[(1 - \theta)\lambda]$-accuracy with

respect to the noisy output $\tilde{q}(X) = q(X) + \xi$:

$$\left| q(\hat{X}) - \tilde{q}(X) \right| \leq (1-\theta)\lambda.$$

Hence overall, by the triangle inequality, the error of $\hat{X}$ is bounded as

$$\begin{aligned} |q(\hat{X}) - q(X)| &\leq |\tilde{q}(X) - q(X)| + |q(\hat{X}) - \tilde{q}(X)| \\ &\leq \theta\lambda + (1-\theta)\lambda \\ &= \lambda. \end{aligned} \tag{8}$$

Algorithm 4.1 Efficient Synthetic Synopsis Generator

Input: Query set $\mathcal{Q}$, query distribution $\mathcal{D}$, real database X, initial database X^0, loss function $\mathcal{L}(X; \mathcal{Q})$, satisfaction ratio η.
Parameters: Accuracy λ, satisfaction fraction η, failure probability β, accuracy budget θ.
Output: Private database $\hat{X}$.

For each $q \in \mathcal{Q}$, compute its noisy output

$$\tilde{q}(X) \leftarrow q(X) + \text{Laplace}\left(\frac{\rho\sqrt{2\text{k}\ln(1/\delta)}}{\varepsilon} \right).$$

Set $t = 0$.
while $\mathbb{P}_{q\sim\mathcal{D}}[|q(X^t) - \tilde{q}(X)| < (1-\theta)\lambda] < \frac{1}{2} + \eta$ **do**
    if $t = 0$ **or** $|\mathcal{L}_{\pm\exp}(X^t) - \mathcal{L}_{\pm\exp}(X^{t-1})| \geq \tau$ **then**
        Run Coordinate Newton's Method and get X^{t+1}.
    $t \leftarrow t + 1$
$\hat{X} \leftarrow X^t$.
return $\hat{X}$

3.4. Coordinate Newton's Method. Algorithm 4.1 uses coordinate-wise Newton's method as an optimization subroutine. We state Coordinate Newton's Method (Algorithm 4.2) below, together with its well-known convergence guarantee [**13**].

To carry out Algorithm 4.2, we start with an initial guess X^0 for the database X. At iteration t, we select the coordinate $x_i \in X^t$ with the largest magnitude of the loss gradient. We then update this coordinate by fitting a local quadratic approximation,

$$X_i^{t+1} = X_i^t - \alpha_t \frac{\nabla_i \mathcal{L}_{\pm\exp}(X^t)}{\nabla_i^2 \mathcal{L}_{\pm\exp}(X^t)}.$$

Algorithm 4.2 Coordinate Newton's Method

Input: Query set $\mathcal{Q}$ and distribution $\mathcal{D}$, real database X, initial database X^0, loss function $\mathcal{L}_{\pm\exp}(X;\mathcal{Q})$, satisfaction ratio η.
Parameters: Iteration counter t, loss stopping threshold τ.
Output: X^t.

Set $t = 0$.
while $(t = 0$ **or** $|\mathcal{L}_{\pm\exp}(X^t;\mathcal{Q}) - \mathcal{L}_{\pm\exp}(X^{t-1};\mathcal{Q})| \geq \tau)$ **and** satisfied queries $\leq (1/2+\eta)|\mathcal{Q}|$ **do**
    Choose $x_{i_t} \in X^t$ such that $i_t = \text{argmax}_j |\nabla_j \mathcal{L}_{\pm\exp}|$
    $X^{t+1}_{i_t} \leftarrow x_{i_t} - \alpha_t(\nabla_{i_t}\mathcal{L}_{\pm\exp}/\nabla^2_{i_t}\mathcal{L}_{\pm\exp})$.
    $t \leftarrow t + 1$.
return X^t

The step size α_t is found through backtracking line search using the Armijo–Goldstein ([**6**]) stopping condition

$$f(x + \alpha_t d) \leq f(x) + c_0\alpha_t \nabla f(x)^\top d,$$

where $f(x)$ denotes a generic objective function and $c_0 \in (0, 1)$ is a fixed constant. Adapting this condition to our setting, we start with $\alpha_t = 1$ and iteratively check the Armijo-Goldstein condition while shrinking α_t, until (9) is satisfied:

$$\mathcal{L}_{\pm\exp}(\hat{X} + \alpha_t d) \leq \mathcal{L}_{\pm\exp}(\hat{X}) + c_0\alpha_t \nabla_i \mathcal{L}_{\pm\exp}(\hat{X})^\top d, \tag{9}$$

where d is the direction of descent given by

$$d = -\frac{\nabla_i \mathcal{L}_{\pm\exp}(X^t)}{\nabla^2_i \mathcal{L}_{\pm\exp}(X^t)}.$$

The Armijo–Goldstein condition guarantees that the chosen step size yields a sufficient decrease in the objective, no less than the amount projected by a first-order approximation locally. By using the Armijo–Goldstein condition, we prevent Newton's method from overshooting and ensure monotone decrease of the overall loss as the algorithm updates one coordinate of the database at a time.

If $\mathcal{L}_{\pm\exp}$ is convex locally, the Armijo–Goldstein condition chooses a step size that yields a sufficient decrease in the objective, no less than the amount projected by a first-order approximation. By using the Armijo–Goldstein condition, we prevent Newton's method from

overshooting and ensure monotone decrease of the loss at the nice regions of $\mathcal{L}_{\pm\exp}$ where it is convex.

Algorithm 4.2 stops either when a $(1/2+\eta)$-fraction of the queries are accurate or when the per-coordinate loss decrease falls below a threshold $\tau > 0$. Although $\mathcal{L}_{\pm\exp}$ has no guarantee of global convexity, it penalizes large errors $|q(\hat{X}) - \tilde{q}(X)| > \lambda$ aggressively, and Algorithm 4.1 greedily selects the coordinate giving the largest descent. In practice, this makes the method effective, and it typically reaches a satisfactory synopsis efficiently despite the absence of a formal convergence guarantee. We note that the ouput of Algorithm 4.2 is not guaranteed to fall within the same domain as the real database X.

3.5. Privacy Guarantee. Let $k = |\mathcal{Q}|$ denote the number of queries, and let $\mathcal{D}$ be a fixed distribution over $\mathcal{Q}$. We run Algorithm 4.1 with the goal of producing a (λ, η, β)-base synopsis $\hat{X}$ such that, with probability at least $1-\beta$ over the randomness of the algorithm $\mathcal{M}$, $\hat{X}$ is λ-accurate on $(1/2+\eta)$-fraction of queries drawn from $\mathcal{D}$:

$$\mathbb{P}_{\mathcal{M}}\Big[\mathbb{P}_{q\sim\mathcal{D}}\Big[|q(\hat{X}) - q(X)| \le \lambda\Big] \ge \tfrac{1}{2} + \eta\Big] \ge 1 - \beta. \tag{10}$$

Theorem 4.2 states the privacy guarantee for Algorithm 4.1.

THEOREM 4.2. *Let $\mathcal{Q}$ be a set of twice continuously differentiable queries with ℓ_1-sensitivity ρ, and let X be a database. Let $k = |\mathcal{Q}|$. For any fixed $\theta \in (0,1)$, $\lambda > 0$, $\beta \in (0,1)$, and $\delta \in (0,1)$, Algorithm 4.1 satisfies (ε, δ)-differential privacy, where*

$$\varepsilon = O\left(\frac{\rho\sqrt{k\ln(1/\delta)}}{\theta\lambda}\ln\frac{k}{\beta}\right).$$

PROOF. The algorithm releases noisy answers to $k = |\mathcal{Q}|$ queries, which constitutes a k-fold composition of query-answering mechanisms. By (2) in Lemma 1.12, it suffices to ensure that each individual query output is $\big(\varepsilon/\sqrt{2k\ln(1/\delta)},\, 0\big)$-differentially private.

To achieve this, we apply the Laplace mechanism and add independent noise $\xi_q \sim \text{Laplace}\Big(\rho\sqrt{2k\ln(1/\delta)}/\varepsilon\Big)$ to each query output $q(X)$.

Fix an arbitrary parameter $\theta \in (0,1)$. We calibrate the noise so that the probability that the magnitude of the added noise exceeds $\theta\lambda$ for any query is small. By Lemma 1.10,

$$\mathbb{P}\left[|\xi_q| \geq \ln\left(\frac{k}{\beta}\right) \cdot \frac{\rho\sqrt{2k\ln(1/\delta)}}{\varepsilon}\right] \leq \frac{\beta}{k}.$$

Thus, requiring

$$\theta\lambda \;\geq\; \ln\left(\frac{k}{\beta}\right) \cdot \frac{\rho\sqrt{2k\ln(1/\delta)}}{\varepsilon} \tag{11}$$

ensures that $|\xi_q| \leq \theta\lambda$ for all $q \in \mathcal{Q}$ with probability at least $1-\beta$. Rearranging (11) yields the stated bound on ε. □

COROLLARY 4.3. *Assume ε satisfies the noise calibration condition in* (11). *Let $\tilde{q}(X) = q(X) + \xi_q$ denote the noisy query answers. If Algorithm 4.1 returns an iterate $\hat{X}$ such that*

$$\mathbb{P}_{q\sim\mathcal{D}}\Big[|q(\hat{X}) - \tilde{q}(X)| \leq (1-\theta)\lambda\Big] \geq \tfrac{1}{2} + \eta,$$

then, with probability at least $1-\beta$ over the Laplace randomness, $\hat{X}$ is a (λ, η, β)-base synopsis for X.

PROOF. By the noise calibration condition (11) and a union bound over the k queries, with probability at least $1-\beta$ we have $|\tilde{q}(X) - q(X)| \leq \theta\lambda$ for all $q \in \mathcal{Q}$. Conditioned on this event, by (8), $|q(\hat{X}) - q(X)| \leq \lambda$ for any $q \in \mathcal{Q}$. Therefore, every query that is $(1-\theta)\lambda$-accurate with respect to the noisy answers is λ-accurate with respect to the true answers. Combining this implication with the assumed optimization guarantee yields

$$\mathbb{P}_{q\sim\mathcal{D}}\Big[|q(\hat{X}) - q(X)| \leq \lambda\Big] \geq \tfrac{1}{2} + \eta,$$

which holds with probability at least $1-\beta$. □

4. Boosting for Queries

For a small fixed $\eta \in (0, 1/2]$, we observe that in practice, Algorithm 4.1 effectively finds a synthetic synopsis that satisfies a $(1+\eta)$-fraction of the queries when it is allowed a modest number of random initializations. From the perspective of query boosting introduced by [**17**],

we can view the synthetic database $\hat{X}$ output by Algorithm 4.1 as a *weak learner* for X: it answers slightly more than a $(1/2)$-fraction of the queries in $\mathcal{Q}$ accurately. We call such $\hat{X}$ a *base synopsis* and thus Algorithm 4.1 serves as a *base synopsis generator*.

The "boosting for queries" approach of [**17**] builds a base synopsis by taking a random sample of $\mathcal{Q}$. Since our algorithm evaluates the loss accumulated across all queries $q \in \mathcal{Q}$, we omit this sampling procedure and in return achieve a tighter accuracy bound (λ instead of $\lambda + \mu$, where μ is an additional parameter used in [**17**]). We adapt results developed in [**17**] to our setting in Theorem 4.4. The proof for Theorem 4.4 follows [**17**] exactly.

THEOREM 4.4 (Adapted from [**17**]). *Let $\mathcal{Q}$ be a set of twice continuously differentiable queries and let $k = |\mathcal{Q}|$. For a fixed $\eta \in (0, 1/2]$ and for any distribution $\mathcal{D}$ on $\mathcal{Q}$, suppose Algorithm 4.1 produces a (λ, η, β)-base synopsis $\hat{X}_{base}$ that satisfies $(\varepsilon_{base}, \delta_{base})$-differential privacy. Then after $T = (\log k)/\eta^2$ rounds of boosting, Algorithm 4.3 produces a $(T\varepsilon_{base}, T\delta_{base})$-differentially private synthetic database $\hat{X}$ that is λ-accurate for all $q \in \mathcal{Q}$ with probability at least $1 - T\beta$.*

We note that Algorithm 4.3 does not produce a synthetic database that shares the identical structure as the real database. Instead, it produces $\hat{X} = (X_1, \cdots, X_T)$, which is a tuple of synthetic databases that serves as a query-answering synopsis. This discrepency in the structure may restrict downstream analysis on the algorithm's output, but it does not deminish our intended usecase for the algorithm.

5. Experiments

In this section, we demonstrate the usage of Algorithm 4.1 by implementing it on both convex and nonconvex queries. These experiments serve as a proof of concept for our algorithm. Accordingly, we omit a detailed empirical evaluation or comparisons with other methods.

5.1. Convex Queries. We illustrate the accuracy bound that follows from Theorem 4.2 for a simple family of convex queries: the ℓ_p norms of a real-valued database. Although the

Algorithm 4.3 Boosting for Queries, adapted from [**17**]

Input: Real database $X \in \mathcal{X}^n$, query set $\mathcal{Q}$, where each $q \in \mathcal{Q}$ is a function $q : \mathcal{X}^n \to \mathbb{R}$ with sensitivity at most ρ.
Parameters: Parameters used by Algorithm 4.1.
Output: $\hat{X} = (X_1, \cdots, X_T)$.

Initialize $\mathcal{D}_1$ to be the uniform distribution over $\mathcal{Q}$.
for $t = 1, \ldots, T$ **do**
  while X_t is not $(1-\theta)\lambda$-accurate for at least $(1/2+\eta)$ mass of $\mathcal{D}_t$ w.h.p. **do**
    Take an arbitrary initial guess $X_t^0 \in \mathcal{X}^n$.
    Run Algorithm 4.1 to compute a base synopsis X_t that is w.h.p. $(1-\theta)\lambda$-accurate for at least $1/2+\eta$ of the mass of $\mathcal{D}_t$.
  Reweigh the queries. For each $q \in \mathcal{Q}$:
  if X_t is $(1-\theta)\lambda$-accurate, then $a_{t,q} \leftarrow 1$.
  if X_t is not $(1-\theta)\lambda$-accurate, then $a_{t,q} \leftarrow -1$.
  $u_{t,q} \leftarrow \exp\Big(-\alpha \cdot \sum_{j=1}^{t} a_{j,q}\Big)$, where $\alpha = (1/2)\ln((1+2\eta)/(1-2\eta))$.
  Renormalize:
$$Z_t \leftarrow \sum_{q \in \mathcal{Q}} u_{t,q}, \quad \mathcal{D}_{t+1}[q] = u_{t,q}/Z_t.$$
return $\hat{X} = (X_1, \cdots, X_T)$. For $q \in \mathcal{Q}$:
$$q(\hat{X}) = \text{median}\{q(X_1, \cdots, q(X_T))\}.$$

theorem is stated as a privacy guarantee, for this example it is more natural to interpret it in terms of accuracy: once the required Laplace noise is added to ensure (ε, δ)-privacy, the theorem directly yields an explicit accuracy guarantee λ for the resulting synthetic database.

Experimental Setup.

- **Database size:** $n = |X| = 60$.
- **Real database:** We generate $X = (x_1, \ldots, x_n)$ with i.i.d. entries $x_i \sim \text{Uniform}[-1, 1]$ on a 0.01-step grid.
- **Query family:** We use the following queries which compute ℓ_p norms for different values of p
$$\mathcal{Q} = \{q_p : X \mapsto \tfrac{1}{n}(\sum_{i=1}^{n} |x_i|^p)^{1/p}\},$$
with $p \in (1, 5]$ and $|\mathcal{Q}| = 20$.
- **Initialization:** X_0 has each coordinate drawn i.i.d. from the standard Normal distribution $\mathcal{N}(0, 1)$. The probability density function of the Normal distribution is given by A.2.

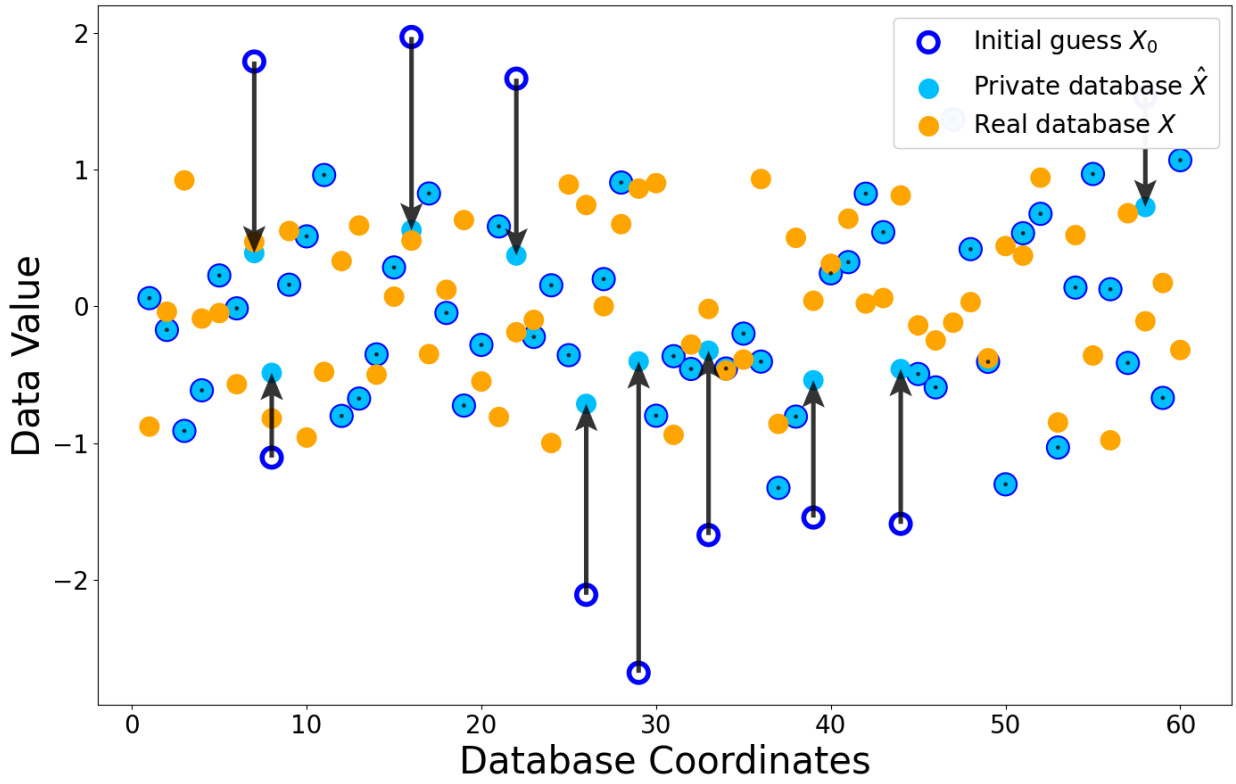

FIGURE 4.1. Initial Guess vs. Final Private Database with Movement Arrows.

- **Random seed:** We fix a global seed for the entire algorithm, chosen at random to the value 1384686389.
- **Privacy and failure parameters:** $\varepsilon = 10$, $\delta = 0.1$, $\beta = 0.05$, and $\eta = 1/2$, so we aim for $\hat{X}$ to satisfy all queries in $\mathcal{Q}$.

Figure 4.1 shows the initial guess X_0 and the synthetic database $\hat{X}$ computed by Algorithm 4.1 under this setup. Algorithm 4.1 terminates after 9 iterations of Newton's method and moves a small subset of the coordinates of X_0 before reaching a satisfactory $\hat{X}$.

Privacy and Accuracy Guarantee. Since each $q_p \in \mathcal{Q}$ is an ℓ_p-norm, its ℓ_1-sensitivity (Definition 1.6) is $\rho = \Delta_1(q) = \frac{1-0}{n} = \frac{1}{60} \approx 0.016$.

Using Lemma 1.10 and Lemma 1.12, to ensure (ε, δ)-privacy for the full query set, we add independent noise $\xi \sim \text{Laplace}\left(\rho\sqrt{2k\ln(1/\delta)}/\varepsilon\right)$ to each query answer. Plugging in $k = 20$, $\rho = 0.016$, $\varepsilon = 10$, and $\delta = 0.1$, we obtain $\xi \sim \text{Laplace}(0.015)$.

For this example, we choose $\theta = 0.6$, which allocates 60% of the error budget to the added Laplace noise and the rest 40% to the optimization procedure. Thus we aim for

$$|\xi| \leq 0.6\lambda, \qquad |q(\hat{X}) - \tilde{q}(X)| \leq 0.4\lambda \quad \forall q \in \mathcal{Q}.$$

With these parameter choices, Theorem 4.2 yields the accuracy bound

$$\lambda = \frac{\ln(k/\beta)\rho\sqrt{2k\ln(1/\delta)}}{\varepsilon\theta} \approx 0.153.$$

Overall, for this set of ℓ_p-norm queries, Algorithm 4.1 produces a synthetic database $\hat{X}$ that is $(10, 0.1)$-differentially private and 0.153-accurate for all $q \in \mathcal{Q}$.

Figure 4.2 shows the performance of $\hat{X}$ on all queries $q \in \mathcal{Q}$. As predicted, the errors lie within the theoretical bound $\lambda \approx 0.153$. It may appear surprising that for every $q \in \mathcal{Q}$, the synthetic output $q(\hat{X})$ is larger than the noisy value $\tilde{q}(X)$, even when the added Laplace noise ξ is negative. We point out that this behavior is natural. For a fixed $p \in (1, 5]$ and a given $\hat{X}$, increasing a coordinate of $\hat{X}$ increases its ℓ_p -norm far more efficiently than decreasing it lowers the norm. Newton's method is greedy and therefore prioritizes changing coordinates that most effectively reduce the overall loss. Moreover, since the loss function aggregates the losses over all $q \in \mathcal{Q}$, increasing a chosen x_i can still reduce the total loss even if some individual queries received negative noise. As a result, Algorithm 4.1 often terminates at a synthetic database $\hat{X}$ that produces larger query outputs when the query class consists of ℓ_p norms.

5.2. Nonconvex Queries. As mentioned in Section 4, for nicely structured classes of nonconvex queries, Algorithm 4.3 is able to produce a synthetic database when allowed a moderate number of random initializations. Such a base synopsis may be a local minimizer for $\mathcal{L}_{\pm\exp}$, which satisfies the accuracy requirement for $(1/2 + \eta)$-fraction of the queries in $\mathcal{Q}$ for some small $\eta \in (0, 1/2]$. To give a concrete example, we consider the following set up.

- **Database size:** $n = |X| = 1600$.

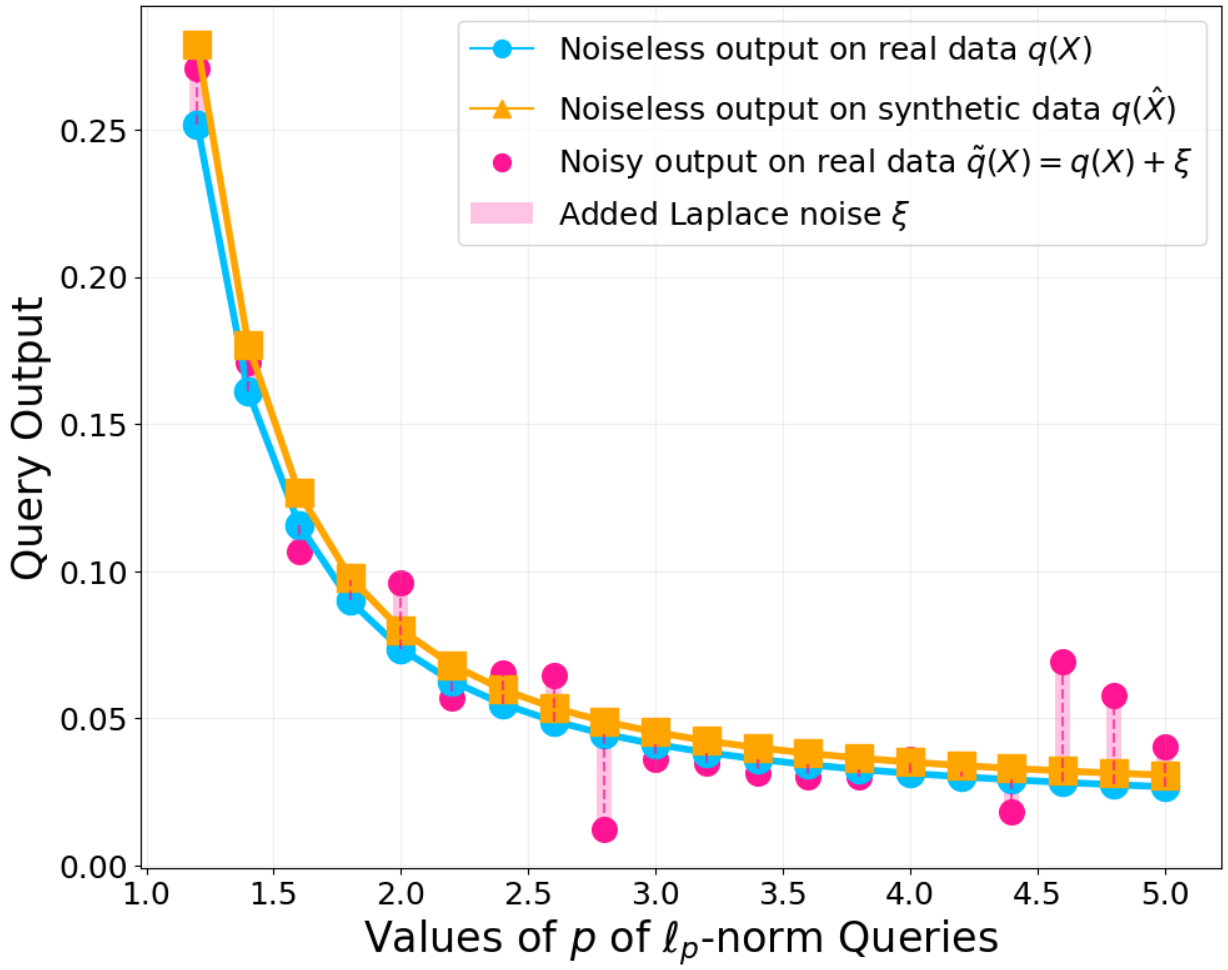

FIGURE 4.2. ℓ_p-norms evaluated on the real and private databases.

- **Real database:** $X = (x_1, \ldots, x_n)$ with coordinates independently drawn as $x_i \sim \mathcal{N}(0, 1)$. If a coordinate's value falls outside of the interval $[-\pi, \pi]$, we redraw for the coordinate.
- **Query family:** Each query is a "spiky" perturbed parabola,

$$q_j(X) = \frac{1}{n} \sum_{i=1}^{n} \big(x_i^2 + a_{j,i} \phi_j(w_j x_i)\big),$$

where $a_{j,i}$ are per-coordinate amplitudes, w_j is the frequency, and $\phi_j \in \{\sin, \cos\}$ is a trig flag that is determined uniformly at random.

- **Parameter generation:** We generate $k = 40$ queries in total, where a $p_{\text{hard}} = 0.05$ fraction of them are "hard" queries. Easy and hard queries differ only by their amplitude and frequency ranges:
  (1) **Easy queries** $(40 \cdot (1 - 0.05) = 38$ queries): $a_{j,i} \sim \text{Uniform}[0.36, 0.55]$ and $w_j \sim \text{Uniform}[2.4, 2.9]$.
  (2) **Hard queries** $(40 \cdot 0.05 = 2$ queries): $a_{j,i} \sim \text{Uniform}[0.45, 0.9]$ and $w_j \sim \text{Uniform}[5.5, 7.0]$.
  (3) **Random seed:** We fix a global seed for the entire algorithm, chosen at random to the value 56567663060557674 38.

Figure 4.3 shows examples of a typical perturbed parabola query from the easy and hard group.

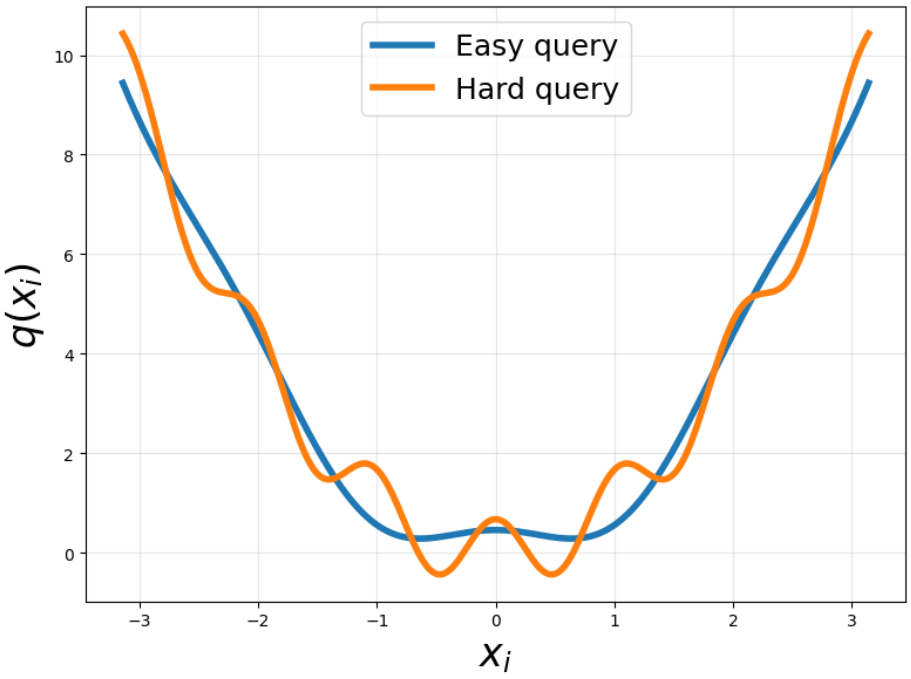

FIGURE 4.3. Typical perturbed parabola queries from the easy and hard group.

In our experiment, after $T = 3$ iterations of boosting, Algorithm 4.3 reaches a λ-accurate synthetic database $\hat{X} = (X_1, X_2, X_3)$ for all $k = 40$ queries. Table 4.1 shows detailed record of the performance of the base synopsis generated during iteration $t = 1, 2$, and 3 of boosting.

Statistic	**t=1**	**t=2**	**t=3**
Random Initialization Attempts	4	1	8
Base-synopsis Satisfaction Ratio	0.525	1.000	1.000
Median-of-answers Sat. Ratio	0.525	0.925	1.000
Stopping threshold: $\lvert q(\hat{X}) - \tilde{q}(X)\rvert \leq (1-\theta)\lambda = 0.812.$			

TABLE 4.1. Boosting run summary (seed: 5656766306055767438).

Privacy Guarantee. The privacy loss comes from two stages of the algorithm: base synopsis generation and boosting.

(1) **Base synopsis.**

The ℓ_1-sensitivity of the queries is

$$\rho = \Delta q = \frac{\pi^2 + 1}{n} = \frac{\pi^2 + 1}{1600} = 0.0068.$$

For the base synopsis, we aim for

$$\varepsilon_{\text{base}} = 5 \quad \text{and} \quad \delta_{\text{base}} = 0.1.$$

We choose $\theta = 0.1$, which allocates 10% of the error budget to the Laplace noise and 90% to optimization. We set the overall failure probability to $\beta = 0.3$. Under this set up, the base synopsis achieves accuracy $\lambda = 0.9022$ via Theorem 4.2.

(2) **Entire database (from boosting).**

Every boosting iteration accumulates $(\varepsilon_{\text{base}}, \delta_{\text{base}}) = (5, 0.1)$-differential privacy. In our experiment, boosting reaches a satisfactory database after $T = 3$ iterations, hence the overall privacy loss is

$$\begin{aligned}(\varepsilon_{\text{boost}}, \delta_{\text{boost}}) &= (T\varepsilon_{\text{base}}, T\delta_{\text{base}}) \\ &= (3 \cdot 5,\ 3 \cdot 0.1) \\ &= (15, 0.3).\end{aligned}$$

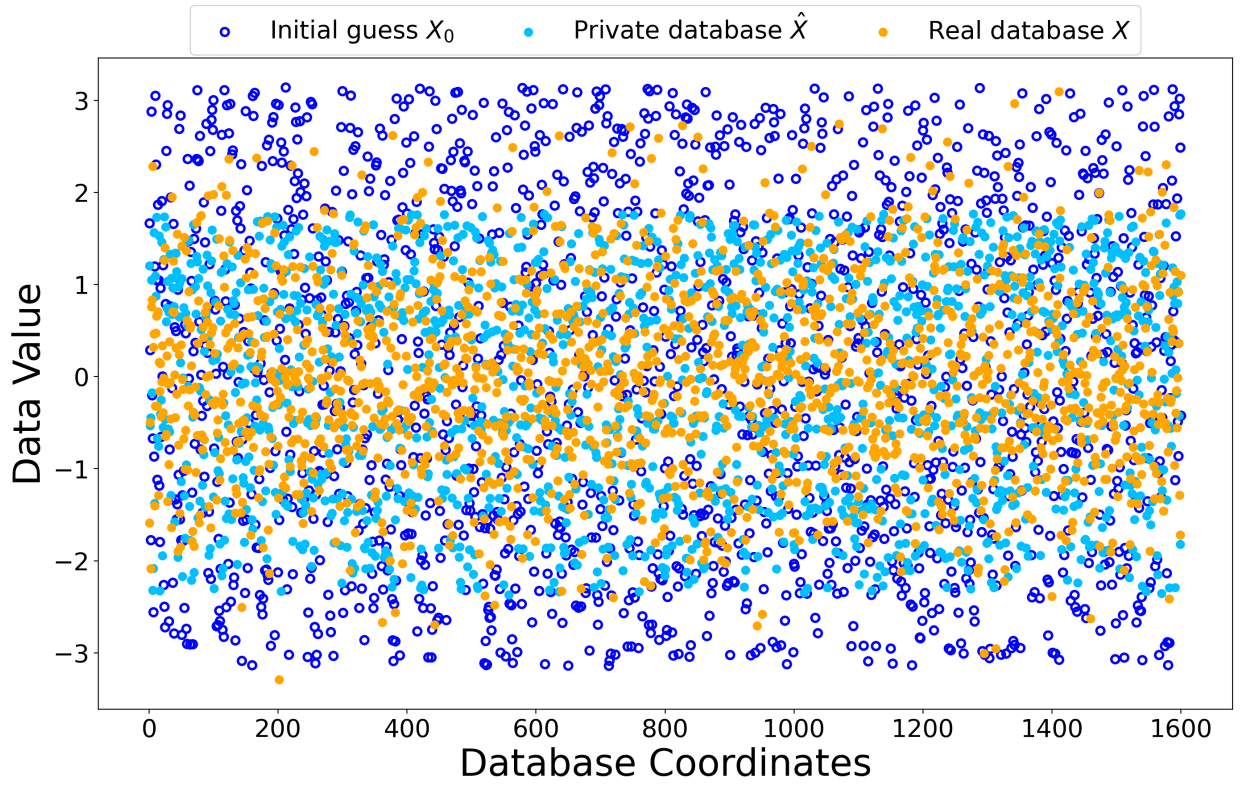

FIGURE 4.4. Initial guess vs. final private database vs. real database.

Accuracy Guarantee. Algorithm 4.3 outputs the sequence of synthetic synopses produced during each iteration of boosting $\hat{X} = (X_1, \cdots, X_T)$. After round t, Algorithm 4.3 checks for each $q \in \mathcal{Q}$ if median$\{q(X_1), \cdots, q(X_t)\}$ is $(1-\theta)\lambda$-accurate with respect to the noisy output $\tilde{q}(X)$. Since we know the error introduced by the Laplace noise is bounded above by $|\xi_q| \leq \theta\lambda$, (8) ensures that for all $q \in \mathcal{Q}$,

$$|q(\hat{X}) - q(X)| \leq \lambda.$$

CHAPTER 5

Vitae

Duan Tu

Education

Unversity of Florida, Gainesville, FL May 2020
B.S. in Mathematics, *magna cum laude*

Publications

1. L. Reyzin* and D. Tu*. "On Sample Reuse Methods for Answering k-wise Statistical Queries." *ISAIM*, 2024. (Extended version invited to *AMAI*.)
2. L. Reyzin* and D. Tu*. "An Efficient Algorithm for Generating Private Synthetic Data." *Submitted*.
3. V. Jain* and D. Tu*. "On Lower Bounds For Local Versions of Metric Embeddings." *Manuscript*.
4. F. Shaerzadeh, L. Phan, D. Miller, . . . , D. Tu, . . . , H. Khoshbouei. "Microglia Senescence Occurs in Both Substantia Nigra and Ventral Tegmental Area." *Glia*, 2020.

Teaching

Instructor
Math 182: Emerging Scholars Workshop for Calculus II, Spring 2024

Teaching Assistant
Math 181: Calculus II, Spring 2023, Fall 2023, Spring 2024
Math 180: Calculus I, Fall 2020, Spring 2021, Spring 2022
Math 110: College Algebra, Fall 2021
Math 105: Mathematical Reasoning, Fall 2022

Grader
STAT 401: Introduction to Probability, Fall 2023

Honors and Awards

Davis United World College Scholar , 2016 - 2020

APPENDIX A

Technical Lemmas

DEFINITION A.1 (Uniform Distribution). If $X \sim \text{Uniform}\{a, a+1, \ldots, b\}$, then its probability mass function is

$$\mathbb{P}(X = k) = \begin{cases} \dfrac{1}{b-a+1}, & k = a, a+1, \ldots, b, \\ 0, & \text{otherwise.} \end{cases}$$

DEFINITION A.2 (Normal Distribution). If $X \sim \mathcal{N}(0,1)$, then its density is

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

DEFINITION A.3 (Laplace Distribution). If X is drawn from the Laplace distribution centered at 0 with scale parameter b, then its density is

$$\text{Laplace}(x; b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right).$$

THEOREM A.4 ([**7**]). *Every K_n^k hypergraph decomposes into a disjoint collection of 1-factors.*

THEOREM A.5 ([**27**]). *Let $X_1, \ldots, X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ almost surely for all $i = 1, \ldots, n$. Let $S_n = \sum_{i=1}^n X_i$. Then for every $t > 0$,*

$$\mathbb{P}\big(S_n - \mathbb{E}[S_n] \geq t\big) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbb{P}\big(|S_n - \mathbb{E}[S_n]| \geq t\big) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

LEMMA A.6 (Advanced Composition [**18**]). *For all $\varepsilon, \delta, \delta' \geq 0$, the class of (ε, δ)-differentially private mechanisms satisfies $(\varepsilon', M\delta + \delta')$-differentially privacy under M-fold*

adaptive composition for

$$\varepsilon' = \varepsilon\sqrt{2M\ln(1/\delta')} + M\varepsilon(e^{\varepsilon} - 1). \tag{1}$$

When ε is small, the M-fold adaptive composition achieves (ε', δ')-differential privacy for

$$\varepsilon' = O\Big(\varepsilon\sqrt{2M\ln(1/\delta')}\Big). \tag{2}$$

PROOF. To prove (2), observe that for any ε_b, we can find some constant $c > 0$ such that $e^{\varepsilon_b} - 1 < c\varepsilon_b$. Therefore, we know

$$\begin{aligned}
\varepsilon_c &= \varepsilon_b\sqrt{2k\ln(1/\delta_c)} + k\varepsilon_b(e^{\varepsilon_b - 1}) \\
&\leq \varepsilon_b\sqrt{2k\ln(1/\delta_c)} + k\ c\ \varepsilon_b^2 \\
&= O\Big(\varepsilon_b\sqrt{k\ln(1/\delta_c)}\Big),
\end{aligned}$$

when ε_b is small compared to $\sqrt{k\ln(1/\delta_c)}$. □

APPENDIX B

Copyright Agreement

RightsLink

SPRINGER NATURE

On sample reuse methods for answering k-wise statistical queries

Author: Lev Reyzin et al

Publication: Annals of Mathematics and Artificial Intelligence

Publisher: Springer Nature

Date: Sep 8, 2025

Creative Commons

Bibliography

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Local embeddings of metric spaces. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 631–640, 2007.

[2] Ittai Abraham, Yair Bartal, and Ofer Neiman. On low dimensional local embeddings. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 875–884. SIAM, 2009.

[3] Ittai Abraham, Yair Bartal, and Ofer Neiman. Local embeddings of metric spaces. *Algorithmica*, 72(2):539–606, 2015.

[4] Noga Alon. Problems and results in extremal combinatorics—I. *Discrete Mathematics*, 273(1-3):31–53, 2003.

[5] Noga Alon and Bo'az Klartag. Optimal compression of approximate inner products and dimension reduction. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–650. IEEE, 2017.

[6] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[7] Zsolt Baranyai. On the factrization of the complete uniform hypergraphs. *Infinite and Finite Sets*, 1, 1974.

[8] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. *SIAM Journal on Computing*, 50(3):STOC16–377–STOC16–405, 2021.

[9] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, page 253–262, New York, NY, USA, 1994. Association for Computing Machinery.

[10] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[11] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. *Journal of the ACM*, 60(2):1–25, 2013.

[12] Jean Bourgain. On lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. *J. Priv. Confidentiality*, 7(3):17–51, 2016.

[16] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014.

[17] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 51–60. IEEE, 2010.

[18] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, 2010.

[19] Vitaly Feldman and Badih Ghazi. On the Power of Learning from k-Wise Queries. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:32, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[20] Vitaly Feldman and Thomas Steinke. Generalization for adaptively-chosen estimators via stable median, 2017.
[21] Vitaly Feldman and Thomas Steinke. Calibrating noise to variance in adaptive data analysis, 2018.
[22] B Fish, L Reyzin, and B Rubinstein. Sampling without compromising accuracy in adaptive data analysis. *31st International Conference on Algorithmic Learning Theory*, 117, 2020.
[23] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
[24] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010.
[25] Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard, 2014.
[26] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.
[27] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
[28] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
[29] Piotr Indyk and Tal Wagner. Near-optimal (euclidean) metric compression. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 710–723. SIAM, 2017.
[30] Vishesh Jain, Huy Pham, and Thuy-Duong Vuong. Dimension reduction for maximum matchings and the fastest mixing Markov chain. *Comptes Rendus. Mathématique*, 361(G5):869–876, 2023.
[31] Vishesh Jain and Duan Tu. On lower bounds for local versions of metric embeddings. Manuscript available at `https://duantu.github.io/PAPER_Lower_bounds_for_local_metric_embeddings.pdf`.
[32] William B Johnson, Joram Lindenstrauss, et al. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
[33] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.
[34] Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017.
[35] Jirı Matoušek. Lecture notes on metric embeddings. Technical report, Technical report, ETH Zürich, 2013.
[36] Lev Reyzin. Statistical queries and statistical algorithms: Foundations and applications, 2020.
[37] Lev Reyzin and Duan Tu. On sample reuse methods for answering k-wise statistical queries. *Annals of Mathematics and Artificial Intelligence*, 2025.
[38] Gideon Schechtman and Adi Shraibman. Lower bounds for local versions of dimension reductions. *Discrete & Computational Geometry*, 41(2):273–283, 2009.
[39] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.
[40] Jonathan Ullman. Answering n2+o(1) counting queries with differential privacy is hard. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 361–370, New York, NY, USA, 2013. Association for Computing Machinery.
[41] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pages 400–416. Springer, 2011.
[42] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.
[43] Vladimir Vapnik. The nature of statistical learning theory. *Springer*, 1995.
[44] Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Steven Wu. New oracle-efficient algorithms for private synthetic data release. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9765–9774. PMLR, 13–18 Jul 2020.