How to Use Heuristics for Differential Privacy

Seth V. Neel
Department of Statistics
The Wharton School, University of Pennsylvania
Philadelphia, USA
Email: sethneel@wharton.upenn.edu

Aaron L. Roth
Department of Computer and Information Sciences
University of Pennsylvania
Philadelphia, USA
aaroth@cis.upenn.edu

Zhiwei Steven Wu
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, USA
zsw@umn.edu

Abstract*—We develop theory for using *heuristics* to solve computationally hard problems in differential privacy. Heuristic approaches have enjoyed tremendous success in machine learning, for which performance can be empirically evaluated. However, privacy guarantees cannot be evaluated empirically, and must be proven — without making heuristic assumptions. We show that learning problems over broad classes of functions — those that have polynomially sized universal identification sets — can be solved privately and efficiently, assuming the existence of a non-private oracle for solving the same problem. Our first algorithm yields a privacy guarantee that is contingent on the correctness of the oracle. We then give a reduction which applies to a class of heuristics which we call *certifiable*, which allows us to convert oracle-dependent privacy guarantees to worst-case privacy guarantee that hold even when the heuristic standing in for the oracle might fail in adversarial ways. Finally, we consider classes of functions for which both they and their dual classes have small universal identification sets. This includes most classes of simple boolean functions studied in the PAC learning literature, including conjunctions, disjunctions, parities, and discrete halfspaces. We show that there is an efficient algorithm for privately constructing synthetic data for any such class, given a non-private learning oracle. This in particular gives the first oracle-efficient algorithm for privately generating synthetic data for contingency tables. The most intriguing question left open by our work is whether or not *every problem* that can be solved differentially privately can be privately solved with an oracle-efficient algorithm. While we do not resolve this, we give a barrier result that suggests that any generic oracle-efficient reduction must fall outside of a natural class of algorithms (which includes the algorithms given in this paper).

***Keywords*-differential privacy; oracle-efficiency; synthetic data;**

Throughout when we refer to the Appendix, we mean the Appendix of the full version of the paper, hosted here: https://arxiv.org/abs/1811.07765.

I. Introduction

Differential privacy is compatible with a tremendous number of powerful data analysis tasks, including essentially any statistical learning problem [KLN+11, CMS11, BST14] and the generation of synthetic data consistent with exponentially large families of statistics [BLR13, RR10, HR10, GRU12, NTZ13]. Unfortunately, it is also beset with a comprehensive set of computational hardness results. Of course, it inherits all of the computational hardness results from the (non-private) agnostic learning literature: for example, even the simplest learning tasks — like finding the best conjunction or linear separator to approximately minimize classification error — are hard [FGKP09, FGRW12, DOSW11]. In addition, tasks that are easy absent privacy constraints can become hard when these constraints are added. For example, although information theoretically, it is possible to privately construct synthetic data consistent with all d-way marginals for d-dimensional data, privately constructing synthetic data for even 2-way marginals is computationally hard [UV10]. These hardness results extend even to providing numeric answers to more than quadratically many statistical queries [Ull16].

How should we proceed in the face of pervasive computational hardness? We might take inspiration from machine learning, which has not been slowed, despite the fact that its most basic prob-

2575-8454/19/$31.00 ©2019 IEEE
DOI 10.1109/FOCS.2019.00014

lems (e.g. learning linear separators) are already hard even to approximate. Instead, the field has employed heuristics with tremendous success — including exact optimization of convex surrogate loss functions (as in the case of SVMs), decision tree heuristics, gradient based methods for differentiable but non-convex problems (as in back-propogation for training neural networks), and integer programming solvers (as in recent work on interpretable machine learning [UR16]). Other fields such as operations research similarly have developed sophisticated heuristics including integer program solvers and SAT solvers that are able to routinely solve problems that are hard in the worst case.

The case of private data analysis is different, however. If we are only concerned with performance (as is the case for most machine learning and combinatorial optimization tasks), we have the freedom to try different heuristics, and evaluate our algorithms in practice. Thus the design of heuristics that perform well in practice can be undertaken as an empirical science. In contrast, differential privacy is an inherently worst-case guarantee that cannot be evaluated empirically (see [GM18] for lower bounds for black-box testing of privacy definitions).

In this paper, we build a theory for how to employ *non-private* heuristics (of which there are many, benefitting from many years of intense optimization) to solve computationally hard problems in differential privacy. Our goal is to guide the design of practical algorithms about which we can still prove theorems:

1) We will aim to prove accuracy theorems *under the assumption that our heuristics solve some non-private problem optimally*. We are happy to make this assumption when proving our accuracy theorems, because accuracy is something that can be empirically evaluated on the datasets that we are interested in. An assumption like this is also necessary, because we are designing algorithms for problems that are computationally hard in the worst case. However:
2) We aim to prove that our algorithms are differentially private in the worst case, even under the assumption that our heuristics might fail in an adversarial manner.

A. Overview of Our Results

Informally, we give a collection of results showing the existence of *oracle-efficient* algorithms for privately solving learning and synthetic data generation problems defined by discrete classes of functions $\mathcal{Q}$ that have a special (but common) combinatorial structure. One might initially ask whether it is possible to give a direct reduction from a non-private but efficient algorithm for solving a learning problem to an efficient private algorithm for solving the same learning problem *without requiring any special structure at all*. However, this is impossible, because there are classes of functions (namely those that have finite VC-dimension but infinite Littlestone dimension) that are known to be learnable absent the constraint of privacy, but are not privately learnable in an information-theoretic sense [BNSV15, ALMM18]. The main question we leave open is whether *being information theoretically learnable under the constraint of differential privacy* is sufficient for oracle-efficient private learning. We give a barrier result suggesting that it might not be.

Before we summarize our results in more detail, we give some informal definitions.

1) Definitions: We begin by defining the kinds of *oracles* that we will work with, and end-goals that we will aim for. We will assume the existence of oracles for (non-privately) solving learning problems: for example, an oracle which can solve the empirical risk minimization problem for discrete linear threshold functions. Because ultimately oracles will be implemented using heuristics, we consider two types of oracles:

1) *Certifiable* heuristic oracles might fail, but when they succeed, they come with a certificate of success. Many heuristics for solving integer programs are certifiable, including cutting planes methods and branch and bound methods. SAT Solvers (and any other heuristic for solving a decision problem in NP) are also certifiable.
2) On the other had, some heuristics are *non-certifiable*. These heuristics might produce incorrect answers, without any indication that they have failed. Support vector machines and logistic regression are examples of non-certifiable heuristic oracles for learning linear threshold functions.

We define an oracle-efficient *non-robustly* differentially private algorithm to be an algorithm that

runs in polynomial time in all relevant parameters given access to an oracle for some problem, and has an accuracy guarantee and a differential privacy guarantee which may both be *contingent* on the guarantees of the oracle — i.e. if the oracle is replaced with a heuristic, the algorithm may no longer be differentially private. Although in certain situations (e.g when we have very high confidence that our heuristics actually do succeed on all instances we will ever encounter) it might be acceptable to have a privacy guarantee that is contingent on having an infallible oracle, we would much prefer a privacy guarantee that held in the worst case. We say that an oracle-efficient algorithm is *robustly* differentially private if its privacy guarantee is not contingent on the behavior of the oracle, and holds in the worst case, even if an adversary is in control of the heuristic that stands in for our oracle.

2) Learning and Optimization: Our first result is a reduction from efficient non-private learning to efficient private learning over any class of functions $\mathcal{Q}$ that has a small universal identification set [GKS93]. A universal identification set of size m is a set of m examples such that the labelling of these examples by a function $q \in \mathcal{Q}$ is enough to uniquely identify q. Equivalently, a universal identification set can be viewed as a *separator set* [SKS16]: for any pair of functions $q \neq q' \in \mathcal{Q}$, there must be some example x in the universal identification set such that $q(x) \neq q(x')$. We will use these terms interchangeably throughout the paper. We show that if $\mathcal{Q}$ has a universal identification set of size m, then given an oracle which solves the empirical risk minimization problem (non-privately) over $\mathcal{Q}$, there is an ϵ-differentially private algorithm with additional running time scaling linearly with m and error scaling linearly with m^2/ϵ that solves the private empirical risk minimization problem over $\mathcal{Q}$. The error can be improved to $O(m^{1.5}\sqrt{\log 1/\delta}/\epsilon)$, while satisfying (ϵ, δ)-differential privacy. Many well studied discrete concept classes $\mathcal{Q}$ from the PAC learning literature have small universal identification sets. For example, in d dimensions, boolean conjunctions, disjunctions, parities, and halfspaces defined over the hypercube have universal identification sets of size d. This means that for these classes, our oracle-efficient algorithm has error that is larger than the generic optimal (and computationally inefficient) learner from [KLN+11] by a factor of $O(\sqrt{d})$. Other classes of functions also have small universal identification sets — for example, decision lists have universal identification sets of size d^2.

The reduction described above has the disadvantage that not only its accuracy guarantees — but also its proof of privacy — depend on the oracle correctly solving the empirical risk minimization problem it is given; it is *non-robustly* differentially private. This shortcoming motivates our main technical result: a generic reduction that takes as input any oracle-efficient non-robustly differentially private algorithm (i.e. an algorithm whose privacy proof might depend on the proper functioning of the oracle) and produces an oracle-efficient *robustly* differentially private algorithm, *whenever the oracle is implemented with a certifiable heuristic*. As discussed above, this class of heuristics includes the integer programming algorithms used in most commercial solvers. In combination with our first result, we obtain robustly differentially private oracle-efficient learning algorithms for conjunctions, disjunctions, discrete halfspaces, and any other class of functions with a small universal identification set.

3) Synthetic Data Generation: We then proceed to the task of constructing synthetic data consistent with a class of queries $\mathcal{Q}$. Following [HRU13, GGAH+14], we view the task of synthetic data generation as the process of computing an equilibrium of a particular zero sum game played between a data player and a query player. In order to compute this equilibrium, we need to be able to instantiate two objects in an oracle-efficient manner:

1) a private *learning* algorithm for $\mathcal{Q}$ (this corresponds to solving the best response problem for the "query player"), and
2) a *no-regret learning algorithm* for a dual class of functions $\mathcal{Q}_{\text{dual}}$ that results from swapping the role of the data element and the query function (this allows the "data player" to obtain a diminishing regret bound in simulated play of the game).

The no-regret learning algorithm need not be differentially private. From our earlier results, we are able to construct an oracle-efficient robustly differentially private learning algorithm for $\mathcal{Q}$ whenever it has a small universal identification set. On the other hand, Syrgkanis et al. [SKS16] show how to obtain an oracle-efficient no regret learning algorithm for a class of functions under

the same condition. Hence, we obtain an oracle-efficient robustly differentially private synthetic data generation algorithm for any class of functions $\mathcal{Q}$ for which both $\mathcal{Q}$ and $\mathcal{Q}_{\text{dual}}$ have small universal identification sets. Fortunately, this is the case for many interesting classes of functions, including boolean disjunctions, conjunctions, discrete halfspaces, and parity functions. The result is that we obtain oracle-efficient algorithms for generating private synthetic data for all of these classes. We note that the oracle used by the data player need not be certifiable.

4) A Barrier Result: Finally, we exhibit a barrier to giving oracle-efficient private learning algorithms for *all* classes of functions $\mathcal{Q}$ known to be privately learnable. We identify a class of private learning algorithms called *perturbed empirical risk minimizers* (pERMs) which output the query that *exactly* minimizes some perturbation of their empirical risk on the dataset. This class of algorithms includes the ones we give in this paper, as well as many other differentially private learning algorithms, including the exponential mechanism and report-noisy-min. We show that any private pERM can be efficiently used as a no-regret learning algorithm with regret guarantees that depend on the scale of the perturbations it uses. This allows us to reduce to a lower bound on the running time of oracle-efficient online learning algorithms due to Hazan and Koren [HK16]. The result is that there exist finite classes of queries $\mathcal{Q}$ such that any oracle-efficient differentially private pERM algorithm must introduce perturbations that are polynomially large in the size of $|Q|$, whereas any such class is information-theoretically privately learnable with error that scales only with $\log|\mathcal{Q}|$.

The barrier implies that *if* oracle-efficient differentially private learning algorithms are as powerful as inefficient differentially private learning algorithms, then these general oracle efficient private algorithms must not be perturbed empirical risk minimizers. We conjecture that the set of problems solvable by oracle-efficient differentially private learners is strictly smaller than the set of problems solvable information theoretically under the constraint of differential privacy, but leave this as our main open question.

B. Additional Related Work

Conceptually, the most closely related piece of work is the "DualQuery" algorithm of [GGAH+14], which in the terminology of our paper is a robustly private oracle-efficient algorithm for generating synthetic data for k-way marginals for constant k. The main idea in [GGAH+14] is to formulate the private optimization problem that needs to be solved so that the only computationally hard task is one that does not depend on private data. There are other algorithms that can straightforwardly be put into this framework, like the projection algorithm from [NTZ13]. This approach immediately makes the privacy guarantees independent of the correctness of the oracle, but significantly limits the algorithm design space. In particular, the DualQuery algorithm (and the oracle-efficient version of the projection algorithm from [NTZ13]) has running time that is proportional to $|\mathcal{Q}|$, and so can only handle polynomially sized classes of queries (which is why k needs to be held constant). The main contribution of our paper is to be able to handle private optimization problems in which the hard computational step is *not* independent of the private data. This is significantly more challenging, and is what allows us to give oracle-efficient robustly private algorithms for constructing synthetic data for exponentially large families $\mathcal{Q}$. It is also what lets give oracle-efficient private *learning* algorithms over exponentially large $\mathcal{Q}$ for the first time.

A recent line of work starting with the "PATE" algorithm [PAE+16] together with more recent theoretical analyses of similar algorithms by Dwork and Feldman, and Bassily, Thakkar, and Thakurta [DF18, BTT18] can be viewed as giving oracle-efficient algorithms for an easier learning task, in which the goal is to produce a finite number of private *predictions* rather than privately output the model that makes the predictions. These can be turned into oracle efficient algorithms for outputting a private model *under the assumption* that the mechanism has access to an additional source of unlabeled data drawn from the same distribution as the private data, but that does not need privacy protections. In this setting, there is no need to take advantage of any special structure of the hypothesis class $\mathcal{Q}$, because the information theoretic lower bounds on private learning proven in [BNSV15, ALMM18] do not apply. In contrast, our results apply without the need for an auxiliary source of non-private data.

Privately producing *contingency tables*, and synthetic data that encode them — i.e. the answers to statistical queries defined by conjunctions of

features — has been a key challenge problem in differential privacy at least since [BCD+07]. Since then, a number of algorithms and hardness results have been given [UV10, GHRU13, KRSU10, TUV12, HRS12, FK14, CTUW14]. This paper gives the first oracle-efficient algorithm for generating synthetic data consistent with a full contingency table, and the first oracle-efficient algorithm for answering arbitrary conjunctions to near optimal error.

Technically, our work is inspired by Syrgkanis et al. [SKS16] who show how a small separator set (equivalently a small universal identification set) can be used to derive oracle-efficient no-regret algorithms in the contextual bandit setting. The small separator property has found other uses in online learning, including in the oracle-efficient construction of nearly revenue optimal auctions [DHL+17]. Hazan and Koren [HK16] show lower bounds for oracle-efficient no-regret learning algorithms in the experts setting, which forms the basis of our barrier result. More generally, there is a rich literature studying oracle-efficient algorithms in machine learning [BDH+05, BBB+08, BILM16] and optimization [BTHKM15] as a means of dealing with worst-case hardness, and more recently, for machine learning subject to fairness constraints [ABD+18, KNRW18, AIK18].

We also make crucial use of a property of differentially private algorithms, first shown by [CLN+16]: That when differentially private algorithms are run on databases of size n with privacy parameter $\epsilon \approx 1/\sqrt{n}$, then they have similar output distributions when run on datasets that are *sampled from the same distribution*, rather than just on neighboring datasets. In [CLN+16], this was used as a tool to show the existence of *robustly generalizing* algorithms (also known as *distributionally private* algorithms in [BLR13]). We prove a new variant of this fact that holds when the datasets are not sampled i.i.d. and use it for the first time in an analysis to prove differential privacy. The technique might be of independent interest.

II. Preliminaries

A. Differential Privacy Tools

Let $\mathcal{X}$ denote a d-dimensional data domain (e.g. $\mathbb{R}^d$ or $\{0,1\}^d$). We write n to denote the size of a dataset S. We call two *data sets* $S, S' \in \mathcal{X}^n$ *neighbors* (written as $S \sim S'$) if S can be derived from S' by replacing a single data point with some other element of $\mathcal{X}$.

Definition 1 (Differential Privacy [DMNS06, DKM+06])**.** *Fix $\varepsilon, \delta \geq 0$. A randomized algorithm $A : \mathcal{X}^* \to \mathcal{O}$ is (ε, δ)-differentially private if for every pair of neighboring data sets $S \sim S' \in \mathcal{X}^*$, and for every event $\Omega \subseteq \mathcal{O}$:*

$$\Pr[A(S) \in \Omega] \leq \exp(\varepsilon) \Pr[A(S') \in \Omega] + \delta.$$

Differentially private computations enjoy two nice properties:

Theorem 1 (Post Processing [DMNS06, DKM+06])**.** *Let $A : \mathcal{X}^* \to \mathcal{O}$ be any (ε, δ)-differentially private algorithm, and let $f : \mathcal{O} \to \mathcal{O}'$ be any function. Then the algorithm $f \circ A : \mathcal{X}^* \to \mathcal{O}'$ is also (ε, δ)-differentially private.*

Post-processing implies that, for example, every *decision* process based on the output of a differentially private algorithm is also differentially private.

Theorem 2 (Basic Composition [DMNS06, DKM+06])**.** *Let $A_1 : \mathcal{X}^* \to \mathcal{O}$, $A_2 : \mathcal{O} \times \mathcal{X}^* \to \mathcal{O}'$ be such that A_1 is $(\varepsilon_1, \delta_1)$-differentially private, and $A_2(o, \cdot)$ is $(\varepsilon_2, \delta_2)$-differentially private for every $o \in \mathcal{O}$. Then the algorithm $A : \mathcal{X}^* \to \mathcal{O}'$ defined as $A(x) = A_2(A_1(x), x)$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

The Laplace distribution plays a fundamental role in differential privacy. The Laplace Distribution centered at 0 with scale b is the distribution with probability density function $\text{Lap}(z|b) = \frac{1}{2b} e^{-\frac{|z|}{b}}$. We write $X \sim \text{Lap}(b)$ when X is a random variable drawn from a Laplace distribution with scale b. Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be an arbitrary function. The ℓ_1 *sensitivity* of f is defined to be $\Delta_1(f) = \max_{S \sim S'} \|f(S) - f(S')\|_1$. The *Laplace mechanism* with parameter ε simply adds noise drawn independently from $\text{Lap}(\frac{\Delta_1(f)}{\varepsilon})$ to each coordinate of $f(S)$.

Theorem 3 ([DMNS06])**.** *The Laplace mechanism is ε-differentially private.*

B. Statistical Queries and Separator Sets

We study learning (optimization) and synthetic data generation problems for statistical queries defined over a data universe $\mathcal{X}$. A statistical query over $\mathcal{X}$ is a function $q : \mathcal{X} \to \{0,1\}$. A statistical query can represent, e.g. any binary classification model or the binary loss function that it induces. Given a dataset $S \in \mathcal{X}^n$, the value of a statistical query q on S is defined to be $q(S) = \frac{1}{n} \sum_{i=1}^{n} q(S_i)$.

In this paper, we will generally think about query classes $\mathcal{Q}$ that represent standard *hypothesis classes* from learning theory – like conjunctions, disjunctions, halfspaces, etc.

In this paper, we will make crucial use of *universal identification sets* for classes of statistical queries. Universal identification sets are equivalent to *separator sets*, defined (in a slightly more general form) in [SKS16].

Definition 2 ([GKS93, SKS16])**.** *A set $U \subseteq \mathcal{X}$ is a* universal identification set *or* separator set *for a class of statistical queries $\mathcal{Q}$ if for every pair of distinct queries $q, q' \in \mathcal{Q}$, there is an $x \in U$ such that:*

$$q(x) \neq q(x')$$

If $|U| = m$, then we say that $\mathcal{Q}$ has a separator set of size m.

Many classes of statistical queries defined over the boolean hypercube have separator sets of size proportional to their VC-dimension. For example, boolean conjunctions, disjunctions, halfspaces defined over the hypercube, and parity functions in d dimensions all have separator sets of size d. When we solve learning problems over these classes, we will be interested in the set of queries that define the 0/1 loss function over these classes: but as we observe in Appendix A, if a hypothesis class has a separator set of size m, then so does the class of queries representing the empirical loss for functions in that hypothesis class.

C. Learning and Synthetic Data Generation

We study private learning as empirical risk minimization (the connection between in-sample risk and out-of-sample risk is standard, and follows from e.g. VC-dimension bounds [KV94] or directly from differential privacy (see e.g. [BST14, DFH+15])). Such problems can be cast as finding a function q in a class $\mathcal{Q}$ that minimizes $q(S)$, subject to differential privacy (observe that the empirical risk of a hypothesis is a statistical query — see Appendix A). We will therefore study minimization problems over classes of statistical queries generally:

Definition 3. *We say that a randomized algorithm $M : \mathcal{X}^n \to \mathcal{Q}$ is an (α, β)-minimizer for $\mathcal{Q}$ if for every dataset $S \in \mathcal{X}^n$, with probability $1 - \beta$, it outputs $M(S) = q$ such that:*

$$q(S) \leq \arg\min_{q^* \in \mathcal{Q}} q^*(S) + \alpha$$

Synthetic data generation, on the other hand, is the problem of constructing a *new* dataset $\hat{S}$ that approximately agrees with the original dataset with respect to a fixed set of statistical queries:

Definition 4. *We say that a randomized algorithm $M : \mathcal{X}^n \to \mathcal{X}^*$ is an (α, β)-accurate synthetic data generation algorithm for $\mathcal{Q}$ if for every dataset $S \in \mathcal{X}^n$, with probability $1 - \beta$, it outputs $M(S) = \hat{S}$ such that for all $q \in \mathcal{Q}$:*

$$|q(S) - q(\hat{S})| \leq \alpha$$

D. Oracles and Oracle Efficient Algorithms

We discuss several kinds of oracle-efficient algorithms in this paper. It will be useful for us to study oracles that solve weighted generalizations of the minimization problem, in which each datapoint $x_i \in S$ is paired with a real-valued weight w_i. In the literature on oracle-efficiency in machine learning, these are widely employed, and are known as *cost-sensitive classification oracles*. Via a simple translation and re-weighting argument, they are no more powerful than unweighted minimization oracles, but are more convenient to work with.

Definition 5. *A weighted optimization oracle for a class of statistical queries $\mathcal{Q}$ is a function $\mathcal{O}^* : (\mathcal{X} \times \mathbb{R})^* \to \mathcal{Q}$ takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and outputs a query $q = \mathcal{O}^*(WD)$ such that*

$$q \in \arg\min_{q^* \in \mathcal{Q}} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i).$$

In this paper, we will study algorithms that have access to weighted optimization oracles for learning problems that are computationally hard. Since we do not believe that such oracles have worst-case polynomial time implementations, in practice, we will instantiate such oracles with heuristics that are not guaranteed to succeed. There are two failure modes for a heuristic: it can fail to produce an output at all, or it can output an incorrect query. The distinction can be important. We call a heuristic that might fail to produce an output, but never outputs an incorrect solution a certifiable heuristic optimization oracle:

Definition 6. *A certifiable heuristic optimization oracle for a class of queries $\mathcal{Q}$ is a polynomial time algorithm $\mathcal{O} : (\mathcal{X} \times \mathbb{R})^* \to (\mathcal{Q} \cup \perp)$ that takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and either outputs $\mathcal{O}(WD) = q \in \arg\min_{q^* \in \mathcal{Q}} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i)$ or else*

outputs $\perp$ ("Fail"). If it outputs a statistical query q, we say the oracle has succeeded.

In contrast, a heuristic optimization oracle (that is not certifiable) has no guarantees of correctness. Without loss of generality, such oracles never need to return "Fail" (since they can always instead output a default statistical query in this case).

Definition 7. *A (non-certifiable) heuristic optimization oracle for a class of queries $\mathcal{Q}$ is an arbitrary polynomial time algorithm $M : (\mathcal{X} \times \mathbb{R})^* \to \mathcal{Q}$. Given a call to the oracle defined by a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ we say that the oracle has succeeded on this call up to error α if it outputs a query q such that $\sum_{(x_i,w_i)\in WD} w_i q(x_i) \leq \min_{q^*\in\mathcal{Q}} \sum_{(x_i,w_i)\in WD} w_i q^*(x_i) + \alpha$. If it succeeds up to error 0, we just say that the heuristic oracle has succeeded. Note that there may not be any efficient procedure to determine whether the oracle has succeeded up to error α.*

We say an algorithm $\mathcal{A}_{\mathcal{O}}$ is (certifiable)-oracle dependent if throughout the course of its run it makes a series of (possibly adaptive) calls to a (certifiable) heuristic optimization oracle $\mathcal{O}$. An oracle-dependent algorithm $\mathcal{A}_{\mathcal{O}}$ is *oracle equivalent* to an algorithm $\mathcal{A}$ if given access to a perfect optimization oracle $\mathcal{O}^*$, $\mathcal{A}_{\mathcal{O}^*}$ induces the same distribution on outputs as $\mathcal{A}$. We now state an intuitive lemma (that could also be taken as a more formal definition of *oracle equivalence*). See the Appendix for a proof.

Lemma 1. *Let $\mathcal{A}_{\mathcal{O}}$ be a certifiable-oracle dependent algorithm that is oracle equivalent to $\mathcal{A}$. Then for any fixed input dataset S, there exists a coupling between $\mathcal{A}(S)$ and $\mathcal{A}_{\mathcal{O}}(S)$ such that $\Pr[\mathcal{A}_{\mathcal{O}}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp] = \Pr[\mathcal{A}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp]$.*

We will also discuss differentially private heuristic optimization oracles, in order to state additional consequences of our construction in Section IV. Note that because differential privacy precludes exact computations, differentially private heuristic oracles are necessarily non-certifiable, and will never succeed up to error 0.

Definition 8. *A weighted (ϵ, δ)-differentially private (α, β)-accurate learning oracle for a class of statistical queries $\mathcal{Q}$ is an (ϵ, δ) differentially private algorithm $\mathcal{O} : (\mathcal{X} \times \mathbb{R})^* \to C$ that takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and outputs a query $q_{priv} \in \mathcal{Q}$ such that with probability $1 - \beta$:*

$$\sum_{(x_i,w_i)\in WD} w_i q_{priv}(x_i) - \operatorname*{argmin}_{q^*\in C} \sum_{(x_i,w_i)\in WD} w_i q^*(x_i) \leq \alpha$$

We say that an algorithm is *oracle-efficient* if given access to an oracle (in this paper, always a weighted optimization oracle for a class of statistical queries) it runs in polynomial time in the length of its input, and makes a polynomial number of calls to the oracle. In practice, we will be interested in the performance of oracle-efficient algorithms when they are instantiated with heuristic oracles. Thus, we further require oracle-efficient algorithms to halt in polynomial time even when the oracle fails. When we design algorithms for optimization and synthetic data generation problems, their (α, β)-accuracy guarantees will generally rely on all queries to the oracle succeeding (possibly up to error $O(\alpha)$). If our algorithms are merely *oracle equivalent* to differentially private algorithms, then their privacy guarantees depend on the correctness of the oracle. However, we would prefer that the *privacy* guarantee of the algorithm not depend on the success of the oracle. We call such algorithms *robustly* differentially private.

Definition 9. *An oracle-efficient algorithm M is (ϵ, δ)-robustly differentially private if it satisfies (ϵ, δ)-differential privacy even under worst-case performance of a heuristic optimization oracle. In other words, it is differentially private for every heuristic oracle $\mathcal{O}$ that it might be instantiated with.*

We write that an oracle efficient algorithm is non-robustly differentially private to mean that it is oracle equivalent to a differentially private algorithm.

III. Oracle Efficient Optimization

In this section, we show how weighted optimization oracles can be used to give differentially private oracle-efficient optimization algorithms for many classes of queries with performance that is worse only by a $\sqrt{d}$ factor compared to that of the (computationally inefficient) exponential mechanism. The first algorithm we give is not robustly differentially private — that is, its differential privacy guarantee relies on having access to a perfect oracle. We then show how to make that algorithm (or any other algorithm that is oracle equivalent to a differentially private algorithm) robustly differentially private when instantiated with a certifiable heuristic optimization oracle.

A. A (Non-Robustly) Private Oracle Efficient Algorithm

In this section, we give an oracle-efficient (non-robustly) differentially private optimization algorithm that works for any class of statistical queries that has a small separator set. Intuitively, it is attempting to implement the "Report-Noisy-Min" algorithm (see e.g. [DR14]), which outputs the query q that minimizes a (perturbed) estimate $\hat{q}(S) \equiv q(S) + Z_q$ where $Z_q \sim \text{Lap}(1/\varepsilon)$ for each $q \in \mathcal{Q}$. Because Report-Noisy-Min samples an independent perturbation for each query $q \in \mathcal{Q}$, it is inefficient: its run time is linear in $|\mathcal{Q}|$. Our algorithm – "Report Separator-Perturbed Min" (RSPM) – instead augments the dataset S in a way that implicitly induces perturbations of the query values $q(S)$. The perturbations are no longer independent across queries, and so to prove privacy, we need to use the structure of a separator set.

The algorithm is straightforward: it simply augments the dataset with one copy of each element of the separator set, each with a weight drawn independently from the Laplace distribution. All original elements in the dataset are assigned weight 1. The algorithm then simply passes this weighted dataset to the weighted optimization oracle, and outputs the resulting query. The number of random variables that need to be sampled is therefore now equal to the size of the separator set, instead of the size of $\mathcal{Q}$. The algorithm is closely related to a no-regret learning algorithm given in [SKS16] — the only difference is in the magnitude of the noise added, and in the analysis, since we need a substantially stronger form of stability.

Report Separator-Perturbed Min (RSPM)

Given: A separator set $U = \{e_1, \ldots, e_m\}$ for a class of statistical queries $\mathcal{Q}$, a weighted optimization oracle $\mathcal{O}^*$ for $\mathcal{Q}$, and a privacy parameter ϵ.

Input: A dataset $S \in \mathcal{X}^n$ of size n.

Output: A statistical query $q \in \mathcal{Q}$.

Sample $\eta_i \sim Lap(2m/\epsilon)$ for $i \in \{1, \ldots, m\}$

Construct a weighted dataset WD of size $n+m$ as follows:

$$WD(S, \eta) = \{(x_i, 1) : x_i \in S\} \cup \{(e_i, \eta_i) : e_i \in U\}$$

Output $q = \mathcal{O}^*(WD(S, \eta))$.

It is thus immediate that the Report Separator-Perturbed Min algorithm is oracle-efficient whenever the size of the separator set m is polynomial: it simply augments the dataset with a single copy of each of m separator elements, makes m draws from the Laplace distribution, and then makes a single call to the oracle:

Theorem 4. *The Report Separator-Perturbed Min algorithm is oracle-efficient.*

The accuracy analysis for the Report Separator-Perturbed Min algorithm is also straightforward, and follows by bounding the weighted sum of the additional entries added to the original data set.

Theorem 5. *The Report Separator-Perturbed Min algorithm is an (α, β)-minimizer for $\mathcal{Q}$ for:*

$$\alpha = \frac{4m^2 \log(m/\beta)}{\epsilon n}$$

Proof: Let q' be the query returned by RSPM, and let q^* be the true minimizer $q^* = \arg\min_{q \in \mathcal{Q}} q^*(S)$. Then we show that with probability $1-\beta, q'(S) \leq q^*(S) + \alpha$. By the CDF of the Laplace distribution and a union bound over the m random variables η_i, we have that with probability $1-\beta$:

$$\forall i,\ |\eta_i| \leq \frac{2m \log(m/\beta)}{\epsilon}.$$

Since for every query q, $q(e_i) \in [0,1]$, this means that with probability $1-\beta$, $q'(WD) \geq q'(S) - m \cdot \frac{2m\log(m/\beta)}{\epsilon n}$. Similarly $q^*(WD) \leq q^*(S) + 2m \cdot \frac{m \log(m/\beta)}{\epsilon n}$. Combining these bounds gives:

$$q'(S) \leq q'(WD) + 2m^2 \frac{\log(m/\beta)}{\epsilon n} \leq$$

$$q^*(WD) + 2m^2 \frac{\log(m/\beta)}{\epsilon n} \leq q^*(S) + \frac{4m^2 \log(m/\beta)}{\epsilon n}$$

as desired, where the second inequality follows because by definition, q' is the true minimizer on the weighted dataset WD. ∎

Remark 1. *We can bound the expected error of RSPM using Theorem 5 as well. If we denote the error of RSPM by E, we've shown that for all β, $\Pr\left[E \geq \frac{4m^2\log(m/\beta)}{\epsilon n}\right] \leq \beta$. Thus $\Pr\left[\frac{\epsilon n E}{4m^2} - \log m \geq \log(1/\beta)\right] \leq \beta$ for all β. Let $\tilde{E} = \max(0, \frac{\epsilon n E}{4m^2} - \log m)$. Since $\tilde{E}$ is non-negative:*

$$\mathbb{E}\left[\tilde{E}\right] = \int_0^\infty \Pr\left[\tilde{E} \geq t\right] \leq \int_0^\infty e^{-t} = 1.$$

Hence $\frac{\epsilon n \mathbb{E}[E]}{4m^2} - \log m \leq \mathbb{E}\left[\tilde{E}\right] \leq 1$, and so $\mathbb{E}[E] \leq$

$\frac{4m^2}{\epsilon n}(1+\log m)$.

The privacy analysis is more delicate, and relies on the correctness of the oracle.

Theorem 6. *If $\mathcal{O}^*$ is a weighted optimization oracle for $\mathcal{Q}$, then the Report Separator-Perturbed Min algorithm is ϵ-differentially private.*

Proof: We begin by introducing some notation. Given a weighted dataset $WD(S,\eta)$, and a query $q \in \mathcal{Q}$, let $q(S,\eta) = q(S) + \sum_{e_i \in U} q(e_i)\eta_i$ be the value when q is evaluated on the weighted dataset given the realization of the noise η. To allow us to distinguish queries that are output by the algorithm on different datasets and different realizations of the perturbations, write $\mathcal{Q}(S,\eta) = \mathcal{O}^*(WD(S,\eta))$. Fix any $q \in \mathcal{Q}$, and define:

$$\mathcal{E}(q,S) = \{\eta : \mathcal{Q}(S,\eta) = q\}$$

to be the event defined on the perturbations η that the mechanism outputs query q. Given a fixed $q \in \mathcal{Q}$ we define a mapping $f_q(\eta) : \mathbb{R}^m \to \mathbb{R}^m$ on noise vectors as follows:

1) If $q(e_i) = 1, f_q(\eta)_i = \eta_i - 2$
2) If $q(e_i) = 0, f_q(\eta)_i = \eta_i + 2$

Equivalently, $f_q(\eta)_i = \eta_i + 2(1 - 2q(e_i))$.

We now make a couple of observations about the function f_q.

Lemma 2. *Fix any $\hat{q} \in \mathcal{Q}$ and any pair of neighboring datasets S, S'. Let $\eta \in \mathcal{E}(\hat{q}, S)$ be such that $\hat{q}$ is the unique minimizer $\hat{q} \in \inf_{q \in \mathcal{Q}} q(S,\eta)$. Then $f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S')$. In particular, this implies that for any such η:*

$$\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S)) \leq \mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))$$

Proof: For this argument, it will be convenient to work with *un-normalized* versions of our queries, so that $q(S) = \sum_{x_i \in S} q(x_i)$ — i.e. we do not divide by the dataset size n. Note that this change of normalization does not change the identity of the minimizer. Under this normalization, the queries q are now 1-sensitive, rather than $1/n$ sensitive.

Recall that $\mathcal{Q}(S,\eta) = \hat{q}$. Suppose for point of contradiction that $\mathcal{Q}(S', f_{\hat{q}}(\eta)) = \tilde{q} \neq \hat{q}$. This in particular implies that $\tilde{q}(S', f_{\hat{q}}(\eta)) \leq \hat{q}(S', f_{\hat{q}}(\eta))$.

We first observe that $\hat{q}(S',\eta) - \tilde{q}(S',\eta) < 2$. This follows because:

$$\tilde{q}(S',\eta) \geq \tilde{q}(S,\eta) - 1 > \hat{q}(S,\eta) - 1 \geq \hat{q}(S',\eta) - 2 \quad (1)$$

Here the first inequality follows because the un-normalized queries q are 1-sensitive, the second follows because $\hat{q} \in \arg\min_{q \in \mathcal{Q}} q(S,\eta)$ is the unique minimizer, and the last inequality follows from the fact that S and S' are neighbors.

Next, we write:

$$\tilde{q}(S', f_{\hat{q}}(\eta)) - \hat{q}(S', f_{\hat{q}}(\eta)) = \tilde{q}(S',\eta) - \hat{q}(S',\eta) + \sum_{i=1}^{m} (\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i)$$

Consider each term in the final sum: $(\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i)$. Observe that by construction, each of these terms is non-negative: Clearly if $\tilde{q}(e_i) = \hat{q}(e_i)$, then the term is 0. Further, if $\tilde{q}(e_i) \neq \hat{q}(e_i)$, then by construction, $(\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i) = 2$. Finally, by the definition of a separator set, we know that there is at least one index i such that $\tilde{q}(e_i) \neq \hat{q}(e_i)$. Thus, we can conclude:

$$\tilde{q}(S', f_{\hat{q}}(\eta)) - \hat{q}(S', f_{\hat{q}}(\eta)) \geq \tilde{q}(S',\eta) - \hat{q}(S',\eta) + 2 > 0$$

where the final inequality follows from applying inequality (**??**). But rearranging, this means that $\hat{q}(S', f_{\hat{q}}(\eta)) < \tilde{q}(S', f_{\hat{q}}(\eta))$, which contradicts the assumption that $\mathcal{Q}(S', f_{\hat{q}}(\eta)) = \tilde{q}$. ∎

Let p denote the probability density function of the joint distribution of the Laplace random variables η, and by abuse of notation also of each individual η_i.

Lemma 3. *For any $r \in \mathbb{R}^m, q \in \mathcal{Q}$:*

$$p(\eta = r) \leq e^{\epsilon} p(\eta = f_q(r))$$

Proof: For any index i and $z \in \mathbb{R}$, we have $p(\eta_i = z) = \frac{\epsilon}{4m} e^{-|z|\epsilon/(2m)}$. In particular, if $|x - y| \leq 2$, $p(\eta_i = y) \leq e^{\epsilon/m} p(\eta_i = x)$. Since for all i and $r \in \mathbb{R}^m$ $|f_q(r)_i - r_i| \leq 1$, we have:

$$\frac{p(\eta = f_q(r))}{p(\eta = r)} = \prod_{i=1}^{m} \frac{p(\eta_i = f_q(r)_i)}{p(\eta_i = r_i)} \leq \prod_{i=1}^{m} e^{\epsilon/m} = e^{\epsilon}.$$

∎

Lemma 4. *Fix any class of queries $\mathcal{Q}$ that has a finite separator set $U = \{e_1, \ldots, e_m\}$. For every dataset S there is a subset $B \subseteq \mathbb{R}^m$ such that:*

1) $\Pr[\eta \in B] = 0$ *and*
2) *On the restricted domain $\mathbb{R}^m \setminus B$, there is a unique minimizer $q' \in \arg\min_{q \in \mathcal{Q}} q(S,\eta)$*

Proof: Let:

$$B = \{\eta : \left| \arg\min_{q \in \mathcal{Q}} (q(S) + \sum_{i=1}^{m} \eta_i q(e_i)) \right| > 1\}$$

be the set of η values that do *not* result in unique minimizers q'.

Because $\mathcal{Q}$ is a finite set[1], by a union bound it suffices to show that for any two distinct queries $q_1, q_2 \in \mathcal{Q}$,

$$\Pr_{\eta}\left[q_1(S) + \sum_{i=1}^{m} \eta_i q_1(e_i) = q_2(S) + \sum_{i=1}^{m} \eta_i q_2(e_i)\right] = 0.$$

This follows from the continuity of the Laplace distribution. Let i be any index such that $q_1(e_i) \neq q_2(e_i)$ (recall that by the definition of a separator set, such an index is guaranteed to exist). For any fixed realization of $\{\eta_j\}_{j\neq i}$, there is a single value of η_i that equalizes $q_1(S,\eta)$ and $q_2(S,\eta)$. But any single value is realized with probability 0.

■

We now have enough to complete the proof. We have for any query $\hat{q}$:

$$\begin{aligned}
&\Pr[RSPM(S) = \hat{q}] = \Pr[\eta \in \mathcal{E}(\hat{q}, S)] \\
&= \int_{\mathbb{R}^m} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S))d\eta \\
&= \int_{\mathbb{R}^m \setminus B} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S))d\eta && \text{Lemma 9} \\
&\leq \int_{\mathbb{R}^m \setminus B} p(\eta)\mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))d\eta && \text{Lemma 9} \\
&\leq \int_{\mathbb{R}^m \setminus B} e^{\epsilon} p(f_{\hat{q}}(\eta))\mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))d\eta && \text{Lemma 8} \\
&\leq \int_{\mathbb{R}^m \setminus f_{\hat{q}}(B)} e^{\epsilon} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S'))\left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| d\eta && \eta \to f_{\hat{q}}(\eta) \\
&= \int_{\mathbb{R}^m} e^{\epsilon} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S'))d\eta && \left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| = 1 \\
&= e^{\epsilon} \Pr[\eta \in \mathcal{E}(\hat{q}, S')] \\
&= e^{\epsilon} \Pr[RSPM(S') = \hat{q}]
\end{aligned}$$

■

In Appendix B4, we give a somewhat more complicated analysis to show that by using Gaussian perturbations rather than Laplace perturbations, it is possible to improve the accuracy of the RSPM algorithm by a factor of $\sqrt{m}$, at the cost of satisfying (ϵ, δ)-differential privacy:

Theorem 7. *The Gaussian RSPM algorithm is (ϵ, δ)-differentially private, and is an oracle-efficient (α, β)-minimizer for any class of functions $\mathcal{Q}$ that has a universal identifications sequence of size m for:*

$$\alpha = O\left(\frac{m\sqrt{m\ln(m/\beta)\ln(1/\delta)}}{\varepsilon n}\right)$$

See Appendix B4 for the algorithm and its analysis.

It is instructive to compare the accuracy that we can obtain with oracle-efficient algorithms to the accuracy that can be obtained via the (inefficient, and generally optimal) exponential mechanism based generic learner from [KLN+11]. The existence of a universal identification set for $\mathcal{Q}$ of size m implies $|\mathcal{Q}| \leq 2^m$ (and for many interesting classes of queries, including conjunctions, disjunctions, parities, and discrete halfspaces over the hypercube, this is an equality — see Appendix A). Thus, the exponential-mechanism based learner from [KLN+11] is (α, β)-accurate for:

$$\alpha = O\left(\frac{m + \log(1/\beta)}{\epsilon n}\right).$$

Comparing this bound to ours, we see that we can obtain oracle-efficiency at a cost of roughly a factor of $\sqrt{m}$ in our error bound. Whether or not this cost is necessary is an interesting open question.

We can conclude that for a wide range of hypothesis classes $\mathcal{Q}$ including boolean conjunctions, disjunctions, decision lists, discrete halfspaces, and several families of circuits of logarithmic depth (see Appendix A) there is an oracle-efficient differentially private learning algorithm that obtains accuracy guarantees within small polynomial factors of the optimal guarantees of the (inefficient) exponential mechanism.

B. A Robustly Differentially Private Oracle-Efficient Algorithm

The RSPM algorithm is not *robustly* differentially private, because its privacy proof depends on the oracle succeeding. This is an undesirable property for RSPM and other algorithms like it, because we do not expect to have access to *actual* oracles for hard problems even if we expect that there are certain families of problems for which we can reliably solve typical instances[2]. In this section, we show how to remedy this: we

[1] Any class of queries $\mathcal{Q}$ with a separator set of size m can be no larger than 2^m.

[2] There may be situations in which it is acceptable to use non robustly differentially private oracle-efficient algorithms — for example, if the optimization oracle is so reliable that it has never been observed to fail on the domain of interest. But robust differential privacy provides a worst-case guarantee which is preferable.

give a black box reduction, starting from a (non-robustly) differentially private algorithm $\mathcal{A}_{\mathcal{O}}$ that is implemented using a *certifiable* heuristic[3] oracle $\mathcal{O}$, and producing a robustly differentially private algorithm $\tilde{\mathcal{A}}_{\mathcal{O}}$ for solving the same problem. $\tilde{\mathcal{A}}_{\mathcal{O}}$ will be (ϵ, δ)-differentially private for a parameter δ that we may choose, and will have a factor of roughly $\tilde{O}(1/\delta)$ running time overhead on top of $\mathcal{A}_{\mathcal{O}}$. So if $\mathcal{A}_{\mathcal{O}}$ is oracle efficient, so is $\tilde{\mathcal{A}}_{\mathcal{O}}$ whenever the chosen value of $\delta \geq 1/\text{poly}(n)$. If the oracle never fails, then we can prove utility guarantees for it when $\mathcal{A}_{\mathcal{O}}$ has such guarantees, since it just runs $\mathcal{A}_{\mathcal{O}}$ (using a smaller privacy parameter) on a random sub-sample of the original dataset. But the privacy guarantees hold even in the worst case of the behavior of the oracle. We call this reduction the *Private Robust Subsampling Meta Algorithm* or **PRSMA**.

1) Intuition and Proof Outline: Before we describe the analysis of **PRSMA**, a couple of remarks are helpful in order to set the stage.

1) At first blush, one might be tempted to assert that if an oracle-efficient non-robustly differentially private algorithm is implemented using a certifiable heuristic oracle, then it will sample from a differentially private distribution *conditioned on the event that the heuristic oracle doesn't fail*. But a moment's thought reveals that this isn't so: the possibility of failures both on the original dataset S and on the (exponentially many) neighboring datasets S' can substantially change the probabilities of arbitrary events Ω, and how these probabilities differ between neighboring datasets.
2) Next, one might think of the following simple candidate solution: Run the algorithm $\mathcal{A}_{\mathcal{O}}(S)$ roughly $\tilde{O}(1/\delta)$ many times in order to check that the failure probability of the heuristic algorithm on S is $\ll \delta$, and then output a sample of $\mathcal{A}_{\mathcal{O}}(S)$ only if this is so. But this doesn't work either: the failure probability itself will change if we replace S with a neighboring dataset S', and so this won't be differentially private. In fact, there is no reason to think that the failure probability of $\mathcal{A}_{\mathcal{O}}$ will be a low sensitivity function of S, so there is no way to privately estimate the failure probability to non-trivial error.

[3]We recall that heuristics for solving integer programs (such as cutting planes methods, branch and bound, and branch and cut methods, as implemented in commercial solvers) and SAT solvers are certifiable.

Private Robust Subsampling Meta Algorithm (PRSMA)

Given: Privacy parameters $\epsilon, \delta \geq 0$ and an oracle-efficient differentially private algorithm $\mathcal{A}^{\epsilon}_{\mathcal{O}} : \mathcal{X}^n \to \mathcal{M}$, implemented with a certifiable heuristic oracle $\mathcal{O}$.

Input: A dataset $S \in \mathcal{X}^n$ of size n.

Output: An output $m \in \mathcal{M}$ or $\perp$ ("Fail").

1: Randomly partition S into $K = \frac{1}{\epsilon}(1 + \log(\frac{2}{\delta}))$ equally sized datasets $\{S_i\}_{i=1}^{K}$. (If n is not divisible by K, first discard $n \mod K$ elements at random.)
2: **for** $i = 1 \ldots K$ **do**
3: Set $o_i = PASS$
4: **for** $t = 1 \ldots \frac{\log(K/\delta)}{\delta}$ **do**
5: Compute $\mathcal{A}^{\epsilon'}_{\mathcal{O}}(S_i) = a_{it}$, where $\epsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$
6: If $a_{it} = \perp$, set $o_i = \perp$
7: **end for**
8: **end for**
9: Compute $T = \#\{o_i \neq \perp\}$. Let $\tilde{T} = T + z$, where $z \sim \text{Lap}(\frac{1}{\epsilon})$.
10: Test if $\tilde{T} > \frac{1}{\epsilon}(1 + \log(\frac{1}{\delta}))$, if no output $\perp$ and halt. **Else:**
11: Sample a uniformly at random from $\{a_{it} : o_i \neq \perp\}$.
12: Output a.

It *is* possible to use the *subsample-and-aggregate* procedure of [NRS07] to randomly partition the dataset into K pieces S_i, and privately estimate on *how many* of these pieces $\mathcal{A}_{\mathcal{O}}(S_i)$ fails with probability $\ll \delta$. The algorithm can then then fail if this private count is not sufficiently large. In fact, this is the first thing that **PRSMA** does, in lines 1-10, setting $o_i = PASS$ for those pieces S_i such that it seems that the probability of failure is $\ll \delta$, and setting $o_i = \perp$ for the others.

But the next step of the algorithm is to randomly select one of the partition elements S_i amongst the set that passed the earlier test: i.e. amongst the set such that $o_i \neq \perp$ — and return one of the outputs a that had been produced by running $\mathcal{A}_{\mathcal{O}}(S_i)$. It is not immediately clear why this should be private, because *which* partition elements passed the test $\{i : o_i \neq \perp\}$ is not itself differentially private.

Showing that this results in a differentially private output is the difficult part of the analysis.

To get an idea of the problem that we need to overcome, consider the following situation which our analysis must rule out: Fix a partition of the dataset $S_1, \ldots, S_K$, and imagine that each partition element passes: we have $o_i \neq \perp$ for all i. Now suppose that there is some event Ω such that $\Pr[\mathcal{A}_{\mathcal{O}}(S_1) \in \Omega] \geq 1/2$, but $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega]$ is close to 0 for all $i \neq 1$. Since $K \approx 1/\epsilon$, and the final output is drawn from a uniformly random partition element, this means that **PRSMA** outputs an element of Ω with probability $\Omega(\epsilon)$. Suppose that on a neighboring dataset S', S_1 no longer passes the test and has $o_1 = \perp$. Since it is no longer a candidate to be selected at the last step, we now have that on S', **PRSMA** outputs an element of Ω with probability close to 0. This is a violation of (ϵ, δ)-differential privacy for any non-trivial value of δ (i.e. $\delta \leq O(\epsilon)$).

The problem is that (fixing a partition of S into $S_1, \ldots, S_K$) moving to a neighboring dataset S' can potentially arbitrarily change the probability that any single element S_i survives to step 11 of the algorithm, which can in principle change the probability of arbitrary events Ω by an *additive* $\pm O(\epsilon)$ term, rather than a *multiplicative* $1 \pm O(\epsilon)$ *factor*.

Since we are guaranteed that (with high probability) if we make it to step 11 without failing, then at least $\Omega(1/\epsilon)$ elements S_i have survived with $o_i \neq \perp$, it would be sufficient for differential privacy if for every event Ω, the probabilities $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega]$ were within a constant factor of each other, for all i. Then a change of whether a single partition element S_i survives with $o_i \neq \perp$ or not would only add or remove an ϵ *fraction* of the total probability mass on event Ω. While this seems like a "differential-privacy" like property, but it is not clear that the fact that $\mathcal{A}_{\mathcal{O}^*}$ is differentially private can help us here, because the partition elements S_i, S_j are not neighboring datasets — in fact, they are disjoint. But as we show, it does in fact guarantee this property *if* we set the privacy parameter ϵ' to be sufficiently small — to roughly $O(1/\sqrt{n/K})$ in step 5.

With this intuition setting the stage, the roadmap of the proof is as follows. For notational simplicity, we write $\mathcal{A}(\cdot)$ to denote $\mathcal{A}_{\mathcal{O}^*}(\cdot)$, the oracle-efficient algorithm when implemented with a perfect oracle.

1) We observe that ϵ-differential privacy implies that the log-probability of any event Ω when $\mathcal{A}(\cdot)$ is run on S_i changes by less than an additive factor of ϵ when an element of S_i is changed. We use a method of bounded differences argument to show that this implies that the log-probability density function concentrates around its expectation, where the randomness is over the subsampling of S_i from S. A similar result is proven in [CLN+16] to show that differentially private algorithms achieve what they call "perfect generalization." We need to prove a generalization of their result because in our case, the elements of S_i are not selected independently of one another. This guides our choice of ϵ' in step 5 of the algorithm.
2) We show that with high probability, for every S_i such that $o_i \neq \perp$ after step 10 of the algorithm, $\mathcal{A}_{\mathcal{O}}(S_i)$ fails with probability at most $O(\delta)$. By Lemma 1, this implies that it is δ-close in total variation distance to $\mathcal{A}(S_i)$.
3) We observe that fixing a partition, on a neighboring dataset, only one of the partition elements S_i changes — and hence changes its probability of having $o_i \neq \perp$. Since with high probability, conditioned on **PRSMA** not failing, $\Omega(1/\epsilon)$ partition elements survive with $o_i \neq \perp$, parts 1 and 2 imply that changing a single partition element S_i only changes the probability of realizing any outcome event by a *multiplicative* factor of $\approx 1 + \epsilon$.

Theorem 8. ***PRSMA*** *is* (ϵ, δ) *differentially private when given as input:*

1) *An oracle-efficient non-robustly differentially private algorithm* $\mathcal{A}_{\mathcal{O}}$ *implemented with a certifiable heuristic oracle* $\mathcal{O}$*, and*
2) *Privacy parameters* (ϵ^*, δ^*) *where* $\epsilon^* = \frac{\epsilon}{62} \leq \frac{1}{2}$ *and* $\delta^* = \frac{\delta}{11} \leq \frac{1}{2}$.

We now turn to **PRSMA**'s accuracy guarantees. Note that when **PRSMA** starts with an algorithm $\mathcal{A}_{\mathcal{O}^*}$ instantiated with a perfect oracle $\mathcal{O}^*$, it with high probability outputs the result of running $\mathcal{A}_{\mathcal{O}^*}$ on a subsampled dataset S_i of size $n/K \approx \epsilon n$, with privacy parameter $\epsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. In general, therefore, the accuracy guarantees of **PRSMA** depend on how robust the guarantees of $\mathcal{A}$ are to subsampling, which is typical of "Subsample and Aggregate" approaches, and also to its specific privacy-accuracy tradeoff. Learning algorithms are robust to sub-sampling however: below we derive

an accuracy theorem for **PRSMA** when instantiated with our oracle-efficient RSPM algorithm.

Theorem 9. *Let $\mathcal{Q}$ a class of statistical queries with a separator set of size m. Let $\mathcal{A}_{\mathcal{O}^*}$ denote the RSPM algorithm with access to $\mathcal{O}^*$, a perfect weighted optimization oracle for $\mathcal{Q}$. Then **PRSMA** instantiated with $\mathcal{A}_{\mathcal{O}^*}$, run on a dataset S of size n, with input parameters ϵ and δ is an (α,β)-minimizer for any $\beta > \delta$ and*

$$\alpha \leq \tilde{O}\left(\frac{m^2 \log\left(\frac{m}{\beta-\delta}\right)\log(1/\delta) + \sqrt{\log(1/\delta)\log\left(\frac{|\mathcal{Q}|}{\beta-\delta}\right)}}{\sqrt{n\epsilon}}\right),$$

where the $\tilde{O}$ hides logarithmic factors in $\frac{1}{\epsilon}, \log(\frac{1}{\delta})$.

Proof: With probability at least $1-\delta$, **PRSMA** outputs the result of RSPM run on an n/K fraction of the dataset, with privacy parameter $\epsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. We will condition on this event for the remainder of the proof, which occurs except with probability δ.

Let q^* denote the true minimizer on S. Let S_K denote the random subsample, and let q_K denote the true minimizer on S_K. By Theorem 5, we know that for any $\eta > 0$, with probability $1-\eta$, the error on S_K is bounded as follows:

$$\hat{q}(S_K) - q_K(S_K) \leq \frac{2m^2\log(m/\eta)}{\epsilon'\frac{n}{K}}$$

We next bound $\max_{q\in\mathcal{Q}} q(S_K) - q(S)$, the maximum difference between the value that any query takes on S_K compared to the value that it takes on S. By a Chernoff bound for subsampled random variables (see e.g. Theorem 1.2 of [BM13]), for any $q \in \mathcal{Q}, t > 0$,

$$\Pr[q(S_K) - q(S) \geq t] \leq \exp\left(-2\frac{n}{K}t^2\right).$$

By a union bound over $\mathcal{Q}$, this means that with probability $1-\eta$,

$$\max_{q\in\mathcal{Q}} q(S_K) - q(S) \leq \sqrt{\frac{K}{2n}\log\left(\frac{2|\mathcal{Q}|}{\eta}\right)}$$

We now have all the ingredients to complete the bound:

$$\begin{aligned}\hat{q}(S) - q^*(S) &= [\hat{q}(S) - \hat{q}(S_K)] + [\hat{q}(S_K) - q^*(S_K)] + [q^*(S_K) - q^*(S)] \\ &\leq 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q^*(S_K) \\ &= 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q_K(S_K) + q_K(S_K) - q^*(S_K) \\ &\leq 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q_K(S_K).\end{aligned}$$

By a union bound and the results above, we know that the righthand side is less than

$$2\sqrt{\frac{K}{2n}\log(\frac{2|\mathcal{Q}|}{\eta})} + \frac{2m^2\log(m/\eta)}{\epsilon'\frac{n}{K}},$$

with probability at least $1-\eta$. Substituting $\eta = \beta-\delta$, $\epsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$, $K = O(\frac{1}{\epsilon}(1+\log(2/\delta)))$ gives the desired result. ∎

We remark that we can convert this (α,β)-accuracy bound into a bound on the expected error using the same technique we used to compute the expected error of RSPM. The expected error of **PRSMA** with the above inputs is $\tilde{O}(\frac{2m^2(\log m+1)}{\sqrt{n\epsilon}} + \frac{\sqrt{(\log|\mathcal{Q}|+1)}}{\sqrt{n\epsilon}})$.

IV. OracleQuery: Oracle-Efficient Private Synthetic Data Generation

We now apply the oracle-efficient optimization methods we have developed to the problem of generating *private synthetic data*. In particular, given a private dataset S and a query class $\mathcal{Q}$, we would like to compute a synthetic dataset $\hat{S}$ subject to differential privacy such that the error $\max_{q\in\mathcal{Q}} |q(\hat{S}) - q(S)|$ is bounded by some target parameter α. We provide a general algorithmic framework called OracleQuery for designing oracle-efficient algorithms. The crucial property of the query class we rely on to obtain oracle efficiency is *dual separability*, which requires both the query class and its dual class have separator sets. Informally, the dual of a query class $\mathcal{Q}$ is the query class $\mathcal{Q}_{\text{dual}}$ that results from swapping the role of the functions $q \in \mathcal{Q}$ and the data elements $x \in \mathcal{X}$. More formally:

Definition 10 (Dual class and dual separability)**.** *Fix a class of queries $\mathcal{Q}$. For every element x in $\mathcal{X}$, let $h_x : \mathcal{Q} \to \{0,1\}$ be defined such that $h_x(q) = q(x)$. The* dual class $\mathcal{Q}_{\text{dual}}$ *of $\mathcal{Q}$ is the set of all such functions defined by elements in $\mathcal{X}$:*

$$\mathcal{Q}_{\text{dual}} = \{h_x \mid x \in \mathcal{X}\}.$$

We say that the class $\mathcal{Q}$ is (m_1, m_2)-dually separable if

there exists a separator set of size m_1 for $\mathcal{Q}$, and there exists a separator set of size m_2 for $\mathcal{Q}_{\text{dual}}$.

As we will show (see Appendix A), many widely studied query classes, including discrete halfspaces, conjunctions, disjunctions, and parities are dually separable, often with $m_1 = m_2 = d$ (in fact, many of these classes are *self-dual*, meaning $\mathcal{Q} = \mathcal{Q}_{\text{dual}}$). For any $q \in \mathcal{Q}$, define its negation $\neg q$ to be $\neg q(x) = 1 - q(x)$. Let $\neg\mathcal{Q} = \{\neg q \mid q \in \mathcal{Q}\}$ be the *negation of* $\mathcal{Q}$. It will simplify several aspects of our exposition to deal with classes that are closed under negation. For any class $\mathcal{Q}$, define $\overline{\mathcal{Q}} = \mathcal{Q} \cup \neg\mathcal{Q}$ to be the closure of $\mathcal{Q}$ under negation. Note that whenever we have a weighted minimization oracle for $\mathcal{Q}$, we have one for $\neg\mathcal{Q}$ as well — simply by negating the weights. Further, if U is a separator set for $\mathcal{Q}$, it is also a separator set for $\neg\mathcal{Q}$. This implies that we also have oracle efficient learners for $\overline{\mathcal{Q}}$, since we can separately learn over $\mathcal{Q}$ and $\neg\mathcal{Q}$, and then privately take the minimum value query that results from the two procedures (using e.g. report-noisy-min [DR14]).

For this camera-ready version we defer the technical details of our algorithm and its analysis to the full version, and instead state the consequences of our main theorem (that follow from instantiating it with different oracle-efficient learners).

Theorem 10. *Let $\mathcal{Q}$ be an (m_1, m_2)-dually separable query class. Then given access to a weighted minimization oracle $\mathcal{O}$ over the class $\mathcal{Q}_{\text{dual}}$ and a differentially private weighted minimization algorithm $\mathcal{O}_{\epsilon_0,\delta_0}$ for the class $\mathcal{Q}$ (with appropriately chosen privacy parameters ϵ_0 and δ_0), the algorithm* OracleQuery *is oracle-efficient, (ϵ, δ)-differentially private, and (α, β)-accurate with α depending on the instantiation of $\mathcal{O}_{\epsilon_0,\delta_0}$. If $\mathcal{O}_{\epsilon_0,\delta_0}$ is robustly differentially private, then so is* OracleQuery.

1) *If $\mathcal{O}_{\epsilon_0,\delta_0}$ is instantiated with the Gaussian RSPM algorithm, then*

$$\alpha \leq \tilde{O}\left(\frac{m_1^{3/2} m_2^{3/4} \sqrt{\log(m_1/\beta)\log|\mathcal{X}|\log(1/\delta)}}{n\epsilon}\right)^{1/2}$$

In this case, OracleQuery *is oracle equivalent to a differentially private algorithm, but is not robustly differentially private.*

2) *If $\mathcal{O}_{\epsilon_0,\delta_0}$ is instantiated with the **PRSMA** algorithm (using the Laplace RSPM as $\mathcal{A}_{\mathcal{O}^*}$), then*

$$\alpha \leq \tilde{O}\left(\frac{(m_1^{4/3} + \log^{1/3}(|\mathcal{Q}|)) m_2^{1/4} \log^{1/6}(|\mathcal{X}|)}{(n\varepsilon)^{1/3}}\right).$$

$$\text{polylog}\left(\frac{1}{\beta - \delta}\right)$$

as long as $\beta > \delta$. In this case, OracleQuery *is robustly differentially private.*

3) *If $\mathcal{O}_{\epsilon_0,\delta_0}$ is an (α_0, β_0)-accurate differentially private oracle with $\alpha_0 = O(\log(|\mathcal{Q}|/(\epsilon_0 n)))$, then*

$$\alpha \leq \tilde{O}\left(\frac{m_2^{3/4}\sqrt{\log|\mathcal{X}|\log(1/\delta)}\log(|\mathcal{Q}|/\beta)}{n\epsilon}\right)^{1/2}$$

In this case, OracleQuery *is robustly differentially private.*

where the $\tilde{O}$ hides logarithmic factors in $\frac{1}{\delta}, \frac{1}{\beta}, m_1, m_2, n$ and $\log(|\mathcal{X}|)$.

A couple of remarks are in order.

Remark 2. *The first two bounds quoted in Theorem 10 result from plugging in constructions of oracle-efficient differentially private learners that we gave in Section III. These constructions start with a* non-private *optimization oracle. The third bound quoted in Theorem 10 assumes the existence of a* differentially private *oracle with error bounds comparable to the (inefficient) exponential mechanism based learner of [KLN+11]. We don't know if such oracles can be constructed from non-private (exact) optimization oracles. But this bound is analgous to the bounds given in the non-private oracle-efficient learning literature. This literature gives constructions assuming the existence of perfect learning oracles, but in practice, these oracles are instantiated with heuristics like regression or support vector machines, which exactly optimize some convex surrogate loss function. This is often reasonable, because although these heuristics don't have strong worst-case guarantees, they often perform very well in practice. The same exercise makes sense for private problems: we can use a differentially private convex minimization algorithm to optimize a surrogate loss function (e.g. [CMS11, BST14]), and hope that it does a good job minimizing classification error in practice. It no longer makes sense to assume that the heuristic exactly solves the learning problem (since this is impossible subject to differential privacy) — instead, the analogous assumption is that it does as well as the best inefficient private learner.*

Remark 3. *It is useful to compare the bounds we obtain to the best bounds that can be obtained with inefficient algorithms. To be concrete, consider the class of boolean conjunctions defined over the boolean hypercube $\mathcal{X} = \{0,1\}^d$ (see Appendix A), which are dually-separable with $m_1 = m_2 = d$. The best (ineffi-*

cient) bounds for constructing synthetic data useful for conjunctions [HR10, GRU12] obtain error: $\alpha = O\left(\frac{\sqrt{\log|\mathcal{Q}|}(\log|\mathcal{X}|)^{1/4}}{\sqrt{\epsilon n}}\right)$. *In the case of boolean conjunctions,* $\log|\mathcal{X}| = \log|\mathcal{Q}| = d$, *and so this bound becomes:* $\alpha = O\left(\frac{d^{3/4}}{\sqrt{\epsilon n}}\right)$. *In contrast, the three oracle efficient bounds given in Theorem 10, when instantiated for boolean conjunctions are:*

1) $\alpha = O\left(\frac{d^{11/8}}{\sqrt{\epsilon n}}\right)$,
2) $\alpha = O\left(\frac{d^{7/4}}{(\epsilon n)^{1/3}}\right)$, *and*
3) $\alpha = O\left(\frac{d^{9/8}}{\sqrt{\epsilon n}}\right)$

respectively. Therefore the costs in terms of error that we pay, in exchange for oracle efficiency are $d^{5/8}$, $\frac{d}{(\epsilon n)^{1/6}}$, *and* $d^{3/8}$ *respectively.*

We now give a brief overview of our construction before diving into the technical details in the Appendix.

Proof overview:: We present our solution in three main steps.

1) We first revisit the formulation by [HRU13] that views the synthetic data generation problem as a zero-sum game between a *Data player* and a *Query player*. We leverage the fact that at any approximate equilibrium, the data player's mixed strategy (over $\mathcal{X}$) represents a good synthetic dataset S' with respect to $\mathcal{Q}$.
2) Using the seminal result of [FS96], we will compute the equilibrium for the zero-sum game by simulating *no-regret dynamics* between the two players: in rounds, the Data player plays according to an oracle-efficient online learning algorithm due to [SKS16], and the Query player best responds to the Data player by using a differentially private oracle efficient optimization algorithm. At the end of the dynamics, the average play of the Data player is an approximate minimax strategy for the game, and hence a good synthetic dataset.
3) We instantiate the private best response procedure of the Query player using different oracle-efficient methods, which we have derived in this paper, each of which gives different accuracy guarantees. Finally, we apply our result to several query classes of interest.

V. A Barrier

In this paper, we give *oracle-efficient* private algorithms for learning and synthetic data generation for classes of queries $\mathcal{Q}$ that exhibit special structure: small universal identification sets. Because of information theoretic lower bounds for differentially private learning [BNSV15, ALMM18], we know that these results cannot be extended to *all* learnable classes of queries $\mathcal{Q}$. But can they be extended to all classes of queries that are information theoretically learnable subject to differential privacy? Maybe — this is the most interesting question left open by our work. But here, we present a "barrier" illustrating a difficulty that one would have to overcome in trying to prove this result. Our argument has three parts:

1) First, we observe a folklore connection between differentially private learning and online learning: any differentially private empirical risk minimization algorithm $\mathcal{A}$ for a class $\mathcal{Q}$ that *always outputs the exact minimizer of a data-independent perturbation of the empirical risks* can also be used as a no-regret learning algorithm, using the "follow the perturbed leader" analysis of Kalai and Vempala [KV05]. The per-round run-time of this algorithm is exactly equal to the run-time of $\mathcal{A}$.
2) Oracle-efficient no-regret learning algorithms are subject to a lower bound of Hazan and Koren [HK16], that states that even given access to an oracle which solves optimization problems over a set of experts $\mathcal{Q}$ in unit time, there exist finite classes $\mathcal{Q}$ such that obtaining non-trivial regret guarantees requires total running time larger than poly($|Q|$). This implies a lower bound on the magnitude of the perturbations that an algorithm of the type described in (1) must use.
3) Finally, we observe for any finite class of hypotheses $\mathcal{Q}$, information theoretically, it is possible to solve the empirical risk minimization problem on a dataset of size T up to error $O(\frac{\log|\mathcal{Q}|}{\epsilon T})$ using the generic learner from [KLN$^+$11]. This implies a separation between the kinds of algorithms described in 1), and the (non-efficiently) achievable information theoretic bounds consistent with differential privacy.

We emphasize that oracle efficient algorithms for

learning over $\mathcal{Q}$ have access to a non-private oracle which exactly solves the learning problem over $\mathcal{Q}$ — not an NP oracle, for which the situation is different (see the discussion in Section VI).

First we define the class of mechanisms our barrier result applies to:

Definition 11. *We say that an (ϵ,δ)-differentially private learning algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{Q}$ for $\mathcal{Q}$ is a perturbed Empirical Risk Minimizer (pERM) if there is some distribution $\mathcal{D}_{\epsilon,\delta}$ (defined independently of the data S) over perturbations $Z \in \mathbb{R}^{|\mathcal{Q}|}$ such that on input $S \in \mathcal{X}^n$, $\mathcal{A}$ outputs:*

$$\mathcal{A}(S) = \arg\min_{q\in\mathcal{Q}} \left(n \cdot q(S) + Z_q \right)$$

where $Z \sim \mathcal{D}_{\epsilon,\delta}$.

We note that many algorithms are pERM algorithms. The most obvious example is *report-noisy-min*, in which each $Z_q \sim \mathrm{Lap}(1/\epsilon)$ independently. The exponential mechanism instantiated with empirical loss as its quality score (i.e. the generic learner of [KLN+11]) is also a pERM algorithm, in which each Z_q is drawn independently from a Gumbel distribution [DR14]. But note that the coordinates Z_q need not be drawn independently: The oracle-efficient RSPM algorithm we give in Section III is also a pERM algorithm, in which the perturbations Z_q are introduced in an implicit (correlated) way by perturbing the dataset itself. And it is natural to imagine that many algorithms that employ weighted optimization oracles — which after all solve an exact minimization problem — will fall into this class. The expected error guarantees of these algorithms are proven by bounding $\mathbb{E}[\|Z\|_\infty]$, which is typically a tight bound.

We now briefly recall the online learning setting. Let $\mathcal{Q}$ be an arbitrary class of functions $q : \mathcal{X} \to [0,1]$. In rounds $t = 1,\ldots,T$, the learner selects a function $q^t \in \mathcal{Q}$, and an (adaptive) adversary selects an example $x^t \in \mathcal{X}$, as a function of the sequence $(q^1, x^1, \ldots, q^{t-1}, x^{t-1})$. The learner incurs a loss of $\ell^t = q^t(x^t)$. A standard objective is to minimize the *expected average regret*:

$$R(T) = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} q^t(x^t)\right] - \min_{q\in\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} q(x^t)$$

where the expectation is taken over the randomness of the learner. A weighted optimization oracle in the online learning setting is exactly the same thing as it is in our setting: Given a weighted dataset (S, w), it returns $\arg\min_{q\in\mathcal{Q}} \sum_{x_i\in S} w_i \cdot q(x_i)$.

A natural way to try to use a private learning algorithm in the online learning setting is just to run it at each round t on the dataset defined on the set of data points observed so far: $S_t = \{x^1, \ldots, x^{t-1}\}$.

Definition 12. *Follow the Private Leader, instantiated with $\mathcal{A}$, is the online learning algorithm that at every round t selects $q^t = \mathcal{A}(S_t)$.*

The follow the private leader algorithm instantiated with $\mathcal{A}$ has a controllable regret bound whenever $\mathcal{A}$ is a differentially private pERM algorithm. The following theorem is folklore, but follows essentially from the original analysis of "follow the perturbed leader" by Kalai and Vempala [KV05]. See e.g. the lecture notes from [RS17] or [ALMT17] for an example of this analysis cast in the language of differential privacy. We include a proof in Appendix F for completeness.

Theorem 11. *Let $\varepsilon, \delta \in (0,1)$ and let $\mathcal{A}$ be an (ϵ,δ) differentially private pERM algorithm for query class $\mathcal{Q}$, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$. Then Follow the Private Leader instantiated with $\mathcal{A}$ has expected regret bounded by:*

$$R(T) \le O\left(\epsilon + \delta + \frac{\mathbb{E}_{Z\sim\mathcal{D}_{\varepsilon,\delta}}[\|Z\|_\infty]}{T}\right)$$

Note that the regret is controlled by $\mathbb{E}_{Z\sim\mathcal{D}_{\varepsilon,\delta}}[\|Z\|_\infty]$, which also controls the error of $\mathcal{A}$ as a learning algorithm.

We wish to exploit a lower bound on the running time of oracle efficient online learners over arbitrary sets $\mathcal{Q}$ due to Hazan and Koren [HK16]:

Theorem 12 ([HK16])**.** *For every algorithm with access to a weighted optimization oracle $\mathcal{O}$, there exists a class of functions $\mathcal{Q}$ such that the algorithm cannot guarantee that its expected average regret will be smaller than 1/16 in total time less than $O\left(\sqrt{|\mathcal{Q}|}/\log^3(|\mathcal{Q}|)\right)$.*

Here, it is assumed that calls to the oracle $\mathcal{O}$ can be carried out in unit time, and *total* time refers to the cumulative time over all T rounds of interaction. Hence, if $\mathcal{A}$ is oracle-efficient — i.e. it runs in time $f(t) = \mathrm{poly}(t, \log|\mathcal{Q}|)$ when given as input a dataset of size t, the *total* run time of follow the private leader instantiated with $\mathcal{A}$ is: $\sum_{t=1}^{T} f(t) \le T \cdot f(T) = \mathrm{poly}(T, \log|\mathcal{Q}|)$.

This theorem is almost what we want — except that the order of quantifiers is reversed. It in principle leaves open the possibility that for

every class $\mathcal{Q}$, there is a different oracle efficient algorithm (tailored to the class) that can efficiently obtain low regret. After all, our RSPM algorithm is non-uniform in this way — for each new class of functions $\mathcal{Q}$, it must be instantiated with a separator set for that class.

Via a min-max argument together with an equilibrium sparsification technique, we can give a version of the lower bound of [HK16] that has the order of quantifiers we want — see Appendix F for the proof.

Theorem 13. *For any d, there is a fixed finite class of statistical queries $\mathcal{Q}$ of size $|\mathcal{Q}| = N = 2^d$ defined over a data universe of size $|\mathcal{X}| = O(N^5 \log^2 N)$ such that for every online learning algorithm with access to a weighted optimization oracle for $\mathcal{Q}$, it cannot guarantee that its expected average regret will be $o(1)$ in total time less than $\Omega(\sqrt{N}/\log^3(N))$.*

Theorem 13 therefore implies that follow the private leader, when instantiated with any oracle-efficient differentially private pERM algorithm $\mathcal{A}$ cannot obtain diminishing regret $R(T) = o(1)$ unless the number of rounds $T = \Omega(|Q|^c)$ for some $c > 0$. In combination with Theorem 11, this implies our barrier result:

Theorem 14. *Any oracle efficient (i.e. running in time poly$(n, \log|\mathcal{Q}|)$) (ϵ, δ)-differentially private pERM algorithm instantiated with a weighted optimization oracle for the query class $\mathcal{Q}$ defined in Theorem 13, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$ must be such that for every $(\epsilon + \delta) = o(1)$:*

$$\mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty] \geq \Omega(|\mathcal{Q}|^c)$$

for some constant $c > 0$.

If the accuracy guarantee of $\mathcal{A}$ is proportional to $\mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty]$ (as it is for all pERM algorithms that we know of), this means that there exist finite classes of statistical queries $\mathcal{Q}$ such that no oracle-efficient algorithm can obtain non-trivial error unless the dataset size $n \geq \text{poly}(|\mathcal{Q}|)$. Of course, if $n \geq \text{poly}(|\mathcal{Q}|)$, then algorithms such as report-noisy-min and the exponential mechanism can be run in polynomial time.

This is in contrast with what we can obtain via the generic (inefficient) private learner of [KLN+11], which obtains expected error $O\left(\frac{\log|\mathcal{Q}|}{\epsilon n}\right)$, which is non-trivial whenever $n = \Omega\left(\frac{\log|\mathcal{Q}|}{\epsilon}\right)$. Similarly, because we show in Theorem 13 that the hard class $\mathcal{Q}$ can be taken to have universe size $\mathcal{X} = \text{poly}(\mathcal{Q})$, this means that information theoretically, it is even possible to privately solve the (harder) problem of α-accurate synthetic data for $\mathcal{Q}$ for $\alpha = O\left(\left(\frac{\log^2|\mathcal{Q}|}{\epsilon n}\right)^{1/3}\right)$ using the (inefficient) synthetic data generation algorithm of [BLR13]. This is non-trivial whenever $n = \Omega\left(\frac{\log^2|\mathcal{Q}|}{\epsilon}\right)$. In contrast, our barrier result states is that *if* there exists an oracle-efficient learner $\mathcal{A}$ for this class $\mathcal{Q}$ that has polynomially related sample complexity to what is obtainable absent a guarantee of oracle efficiency, then $\mathcal{A}$ must either:

1) Not be a pERM algorithm, or:
2) Have expected error that is $O\left(\frac{\text{poly}(\log \mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty])}{n}\right)$.

Condition 2. seems especially implausible, as for every pERM we are aware of, $\mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty]$ is a tight bound (up to log factors) on its expected error. In particular, this barrier implies that there is no oracle efficient algorithm for *sampling* from the exponential mechanism distribution used in the generic learner of [KLN+11] for arbitrary query classes $\mathcal{Q}$.

VI. Conclusion and Open Questions

In this paper, we have initiated the systematic study of the power of *oracle-efficient* differentially private algorithms, and have made the distinction between oracle-dependent non-robust differential privacy and robust differential privacy. This is a new direction that suggests a number of fascinating open questions. In our opinion, the most interesting of these is:

> "Can every learning and synthetic data generation problem that is solvable subject to differential privacy be solved with an oracle-efficient (robustly) differentially private algorithm, with only a polynomial blow-up in sample complexity?"

It remains an open question whether or not finite Littlestone dimension characterizes private learnability (it is known that infinite Littlestone dimension precludes private learnability [ALMM18]) — and so one avenue towards resolving both open questions in the affirmative simultaneously would be to show that finite Littlestone dimension can be leveraged to obtain oracle-efficient differentially private learning algorithms.

However, because of our barrier result, we conjecture that the set of query classes that are pri-

vately learnable in an oracle-efficient manner is a *strict subset* of the set that are privately learnable. If this is so, can we precisely characterize this set? What is the right structural property, and is it more general than the sufficient condition of having small universal identification sets that we have discovered?

Even restricting attention to query classes with universal identification sets of size m, there are interesting quantitative questions. The Gaussian version of our RSPM algorithm efficiently obtains error that scales as $m^{3/2}$, but information-theoretically, it is possible to obtain error scaling only linearly with m. Is this optimal error rate possible to obtain in an oracle-efficient manner, or is the $\sqrt{m}$ error overhead that comes with our approach necessary for oracle efficiency?

Our **PRSMA** algorithm shows how to generically reduce from an oracle-dependent guarantee of differential privacy to a guarantee of robust differential privacy — *but at a cost*, both in terms of running time, and in terms of error. Are these costs necessary? Without further assumptions on the construction of the oracle, it seems difficult to avoid the $O(1/\delta)$-overhead in running time, but perhaps there are natural assumptions that can be placed on the failure-mode of the oracle that can avoid this. It is less clear whether the error overhead that we introduce — by running the original algorithm on an ϵ fraction of the dataset, with a privacy parameter $\epsilon' \approx 1/\sqrt{\epsilon n}$ — is necessary. Doing this is a key feature of our algorithm and analysis, because we take advantage of the fact that differentially private algorithms are actually *distributionally private* when ϵ' is set this small — but perhaps it can be avoided entirely with a different approach.

Our barrier result takes advantage of a connection between differentially private learnability and online learnability. Because private pERM algorithms can be used efficiently as no-regret learning algorithms, they are subject to the lower bounds on oracle-efficient online learning proven in [HK16]. But perhaps the connection between differentially private learnability and online learnability runs deeper. Can *every* differentially private learning algorithm be used in a black box manner to efficiently obtain a no-regret learning algorithm? Note that it is already known that private learnability implies finite Littlestone dimension, so the open question here concerns whether there is an *efficient blackbox* reduction from private ERM algorithms to online learning algorithms. If true, this would convert our barrier for pERM algorithms into a full lower-bound for oracle-efficient private learning algorithms generally.

Finally, a more open ended question — that applies both to our work and to work on oracle efficiency in machine learning more generally — concerns how to refine the model of oracle efficiency. Ideally, the learning problems fed to the oracle should be "natural" — e.g. a small perturbation or re-weighting of the original (non-private) learning problem, as is the case for the algorithms we present in our paper. This is desirable because presumably we believe that the heuristics which can solve hard learning problems in practice work for "natural" instances, rather than arbitrary problems. However, the definition for oracle efficiency that we use in this paper allows for un-natural algorithms. For example, it is possible to show that the problem of sampling from the exponential mechanism of [MT07] defined by rational valued quality scores that are efficiently computable lies in $\mathbf{BPP}^{\mathbf{NP}}$ — in other words, the sampling can be done in polynomial time given access to an oracle for solving circuit-satisfiability problems[4]. This implies in particular, that there exists an oracle efficient algorithm (as we have defined them) for any NP hard learning problem — because the learning oracle can be used as an arbitrary **NP** oracle via gadget reductions[5]. The same logic implies that there are oracle efficient no-regret learning algorithms for any class of experts for which offline optimization is NP hard — because an NP oracle can be used to sample from the multiplicative weights distribution. But these kinds of gadget

[4]This construction is due to Jonathan Ullman and Salil Vadhan (personal communication). It starts from the ability to sample uniformly at random amongst the set of satisfying assignments of an arbitrary polynomially sized boolean circuit given an NP oracle, using the algorithm of [BGP00]. For any distribution $\mathcal{P}$ such that there is a polynomially sized circuit C for which the relative probability mass on any discrete input x can be computed by $C(x)$, we can construct a boolean circuit C' that computes for bounded bit-length rational numbers w: $C'(x, w) = 1$ if $C(x) \geq w$. The marginal distribution on elements x when sampling uniformly at random from the satisfying assignments of this circuit is $\mathcal{P}$.

[5]Note that this procedure is not *robustly* differentially private, since sampling from the correct distribution occurs only if the oracle does not fail. But it could be fed into our **PRSMA** algorithm to obtain robust privacy. It also does not solve synthetic data generation oracle efficiently because the quality score used for synthetic data generation in [BLR13] is not computable by a polynomially sized circuit generally.

reductions seem to be an abuse of the model of oracle efficiency, which currently reduces to all of $\mathbf{BPP}^{\mathbf{NP}}$ when the given oracle is solving an NP hard problem[6]. Ambitiously, might there be a refinement of the model of oracle efficiency that requires one to prove a utility theorem along the following lines: assuming an oracle which can with high probability solve learning problems drawn from the actual data distribution, the oracle efficient algorithm will (with slightly lower probability) solve the private learning problem when the underlying instance is drawn from the same distribution. Theorems of this sort would be of great interest, and would (presumably) rule out "unnatural" algorithms relying on gadget reductions.

Acknowledgements: We thank Michael Kearns, Adam Smith, Jon Ullman and Salil Vadhan for insightful conversations about this work. Aaron Roth is supported in part by the Sloan foundation, the DARPA Brandeis project, and NSF awards 1253345 and 1513694.

References

[ABD+18] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 60–69. JMLR.org, 2018.

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[AIK18] Daniel Alabi, Nicole Immorlica, and Adam Kalai. Unleashing linear optimizers for group-fair learning and optimization. In *Conference On Learning Theory*, pages 2043–2066, 2018.

[6]This does not contradict the lower bound of [HK16] for oracle efficient online learning, or our barrier result/conjectured separation in the case of private learning algorithms. This is because oracles solving problems that don't have polynomial time algorithms, but *are not NP hard* cannot be used to encode the arbitrary circuit-SAT instances needed to implement an NP oracle.

[ALMM18] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. *arXiv preprint arXiv:1806.00949*, 2018.

[ALMT17] Jacob Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. *arXiv preprint arXiv:1711.10019*, 2017.

[BBB+08] Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72(1-2):139–153, 2008.

[BCD+07] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282. ACM, 2007.

[BDH+05] Alina Beygelzimer, Varsha Dani, Thomas P. Hayes, John Langford, and Bianca Zadrozny. Error limiting reductions between classification tasks. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 49–56. ACM, 2005.

[BGP00] Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Information and Computation*, 163(2):510–526, 2000.

[BILM16] Alina Beygelzimer, Hal Daumé III, John Langford, and Paul Mineiro. Learning reductions that really work. *Proceedings of the IEEE*, 104(1):136–147, 2016.

[BLR13] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database

privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[BM13] R. Bardenet and O.-A. Maillard. Concentration inequalities for sampling without replacement. *ArXiv e-prints*, September 2013.

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science*, 2015.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 464–473. IEEE, 2014.

[BTHKM15] Aharon Ben-Tal, Elad Hazan, Tomer Koren, and Shie Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[BTT18] Raef Bassily, Om Thakkar, and Abhradeep Thakurta. Model-agnostic private learning via stability. *arXiv preprint arXiv:1803.05101*, 2018.

[CLN+16] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 772–814, 2016.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[CTUW14] Karthekeyan Chandrasekaran, Justin Thaler, Jonathan Ullman, and Andrew Wan. Faster private release of marginals on small databases. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 387–402. ACM, 2014.

[DF18] Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. *arXiv preprint arXiv:1803.10266*, 2018.

[DFH+15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015.

[DHL+17] Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient online learning and auction design. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 528–539. IEEE, 2017.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[DOSW11] Ilias Diakonikolas, Ryan O'Donnell, Rocco A Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 1590–1606. Society for Industrial and Applied Mathematics, 2011.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of*

the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10, pages 51–60, Washington, DC, USA, 2010. IEEE Computer Society.

[FGKP09] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

[FGRW12] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

[FK14] Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *Conference on Learning Theory*, pages 679–702, 2014.

[FS96] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996.*, pages 325–332, 1996.

[GGAH+14] Marco Gaboardi, Emilio Jesús Gallego-Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. *CoRR*, abs/1402.1526, 2014.

[GHRU13] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM Journal on Computing*, 42(4):1494–1520, 2013.

[GKS93] Sally A Goldman, Michael J Kearns, and Robert E Schapire. Exact identification of read-once formulas using fixed points of amplification functions. *SIAM Journal on Computing*, 22(4):705–726, 1993.

[GM18] Anna Gilbert and Audra McMillan. Property testing for differential privacy. *arXiv preprint arXiv:1806.06427*, 2018.

[GRU12] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of cryptography conference*, pages 339–356. Springer, 2012.

[HK16] Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 128–141, 2016.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 61–70, Washington, DC, USA, 2010. IEEE Computer Society.

[HRS12] Moritz Hardt, Guy N Rothblum, and Rocco A Servedio. Private data release via learning thresholds. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 168–187. Society for Industrial and Applied Mathematics, 2012.

[HRU13] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 341–350, New York, NY, USA, 2013. ACM.

[KLN+11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KNRW18] Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR*

Workshop and Conference Proceedings, pages 2569–2577. JMLR.org, 2018.

[KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 775–784. ACM, 2010.

[KV94] Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[KV05] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[LM00] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, 10 2000.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

[NTZ13] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 351–360. ACM, 2013.

[PAE+16] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[RR10] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010.

[RS17] Aaron Roth and Adam Smith. Lecture 15: Algorithmic foundations of adaptive data analysis. *https://adaptivedataanalysis.files.wordpress.com/2017/11* 2017.

[SKS16] Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E. Schapire. Efficient algorithms for adversarial contextual learning. *CoRR*, abs/1602.02454, 2016.

[TUV12] Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

[Ull16] Jonathan Ullman. Answering n^2+o(1) counting queries with differential privacy is hard. *SIAM Journal on Computing*, 45(2):473–496, 2016.

[UR16] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

[UV10] John Ullman and Salil P. Vadhan. Pcps and the hardness of generating private synthetic data. *IACR Theory of Cryptography Conference (TCC)*, 2010.