Private and Online Learnability Are Equivalent

NOGA ALON, Princeton University and Tel Aviv University
MARK BUN, Boston University
ROI LIVNI, Tel Aviv University
MARYANTHE MALLIARIS, University of Chicago
SHAY MORAN, Technion

Let $\mathcal{H}$ be a binary-labeled concept class. We prove that $\mathcal{H}$ can be PAC learned by an (approximate) differentially private algorithm if and only if it has a finite Littlestone dimension. This implies a qualitative equivalence between online learnability and private PAC learnability.

CCS Concepts: • **Theory of computation** → **Sample complexity and generalization bounds**; **Online learning theory**; • **Security and privacy** → **Social aspects of security and privacy;**

Additional Key Words and Phrases: Differential privacy, PAC learning, online learning, Littlestone dimension

ACM Reference format:
Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. 2022. Private and Online Learnability Are Equivalent. *J. ACM* 69, 4, Article 28 (August 2022), 34 pages.
https://doi.org/10.1145/3526074

1 INTRODUCTION

This work studies the relationship between private PAC learning and online learning.

Differentially private learning. Statistical analyses and computer algorithms play significant roles in the decisions that shape modern society. The collection and analysis of individuals' data drives computer programs that determine many critical outcomes, including the allocation of community resources, decisions to give loans, and school admissions.

Although data-driven and automated approaches have obvious benefits in terms of efficiency, they also raise the possibility of unintended negative impacts, especially against marginalized

N. Alon was supported in part by NSF grant DMS-1855464 and BSF grant 2018267. M. Bun was partially supported by NSF grant CCF-1947889 and CAREER award CNS-2046425. R. Livni was supported by ISF grant 2188/20 and a grant from Tel Aviv University Center for AI and Data Science (TAD) in collaboration with Google, as part of the initiative of AI and DS for social good. M. Malliaris was partially supported by NSF CAREER award 1553653 and a Minerva Research Foundation membership at IAS. S. Moran was supported i part by ISF grant 1225/20, BSF grant 2018385, and the Azrieli Foundation.

Authors' addresses: N. Alon, Department of Mathematics Princeton University Washington Road Princeton, NJ 08544 USA, and Schools of Mathematics and Computer Science, Tel Aviv University; email: nalon@math.princeton.edu; M. Bun, Department of Computer Science, Boston University 111 Cummington Mall, Boston, MA, USA 02215; email: mbun@bu.edu; R. Livni, Tel Aviv University, Haim Levanon Street 30, 6997801 University; email: rlivni@tauex.tau.ac.il; M. Malliaris, Department of Mathematics University of Chicago 5734 S. University Avenue Chicago, IL 60637 USA; email: mem@math.uchicago.edu; S. Moran, Department of Mathematics, Technion, Haifa, Israel, 3200003; email: smoran@technion.ac.il.

0004-5411/2022/08-ART28 $15.00
https://doi.org/10.1145/3526074

groups. This possibility highlights the need for *responsible* algorithms that obey relevant ethical requirements (e.g., see [71]).

Differential privacy (DP) [33] plays a key role in this context. Its initial (and primary) purpose was to provide a formal framework for ensuring individuals' privacy in the statistical analysis of large datasets. But it has also found use in addressing other ethical issues such as *algorithmic fairness* (e.g., see [29, 30]).

There is extensive literature identifying differentially private algorithms and their limitations in a variety of contexts, including statistical query release, synthetic data generation, classification, clustering, graph analysis, hypothesis testing, and more. In general, the goal is to understand when and how privacy can be achieved in these tasks with a modest overhead in resources, such as data samples, computation time, or communication. Nevertheless, many basic questions remain regarding which tasks are compatible with DP whatsoever, especially in settings where the data are complex, high dimensional, or infinite.

We study these questions in the private PAC model [57, 78], which captures binary classification tasks under DP. This is the simplest and most extensively studied model of how sensitive data is analyzed in machine learning. In their work introducing this model, Kasiviswanathan et al. [57] showed that every finite class $\mathcal{H}$ is privately learnable using $O(\log|\mathcal{H}|)$ samples. However, this bound is loose for many specific concept classes of interest and says nothing when $\mathcal{H}$ is infinite. Several works gave improved bounds for specific classes [8, 9, 11, 12, 19, 22, 38, 54, 55, 72], but a general characterization of learnability in terms of the combinatorial structure of $\mathcal{H}$ remains elusive. This situation stands in stark contrast to the non-private case, where early results showed that the sample complexity of PAC learning is characterized, up to constant factors, by the VC dimension [14, 79].

In this article, we make progress toward characterizing PAC learnability by algorithms satisfying *approximate* DP. We prove a *qualitative* characterization: we show that a hypothesis class $\mathcal{H}$ is differentially privately learnable (with some finite number of samples) if and only if it is online learnable (with some finite mistake bound).

Online learning. Online learning is a well-studied branch of machine learning that addresses algorithms making real-time predictions on sequentially arriving data. Such tasks arise in contexts including recommendation systems and advertisement placement. The literature on this subject is vast and includes several works (e.g., [24, 46, 73]).

Online Prediction, or *Prediction with Expert Advice*, is a basic setting within online learning. Let $\mathcal{H} = \{h : X \to \{\pm 1\}\}$ be a class of predictors (also called *experts*) over a domain X. Consider an algorithm that observes examples $(x_1, y_1) \ldots (x_T, y_T) \in X \times \{\pm 1\}$ in a sequential manner. More specifically, in each timestep t, the algorithm first observes the instance x_t, then predicts a label $\hat{y}_t \in \{\pm 1\}$, and finally learns whether its prediction was correct. The goal is to minimize the *regret*, namely the number of mistakes compared to the best expert in $\mathcal{H}$:

$$\sum_{t=1}^{T} 1[y_t \neq \hat{y}_t] - \min_{h^* \in \mathcal{H}} \sum_{t=1}^{T} 1[y_t \neq h^*(x_t)].$$

In this context, a class $\mathcal{H}$ is said to be online learnable if for every T there is an algorithm that achieves sublinear regret $o(T)$ against any sequence of T examples. The *Littlestone dimension* is a combinatorial parameter associated to the class $\mathcal{H}$ that characterizes its online learnability [13, 61]: $\mathcal{H}$ is online learnable if and only if it has a finite Littlestone dimension $d < \infty$. Moreover, the best possible regret $R(T)$ for online learning of $\mathcal{H}$ satisfies

$$\Omega(\sqrt{dT}) \leq R(T) \leq O\left(\sqrt{dT \log T}\right).$$

Furthermore, if it is known that if one of the experts never errs (a.k.a. the realizable setting), then the optimal regret is exactly d.[1] (The regret is referred to as *mistake bound* in this context.)

Stability. Although at a first glance it may seem that online learning and differentially private learning have little to do with one another, a recent line of work has revealed a tight connection between the two [2, 3, 16, 44, 49, 69].

At a high level, this connection appears to boil down to the notion of stability, which plays a key role in both topics. On one hand, the definition of DP is itself a form of stability; it requires robustness of the output distribution of an algorithm when its input undergoes small changes. On the other hand, stability also arises as a central motif in online learning paradigms such as *Follow the Perturbed Leader* [51, 52] and *Follow the Regularized Leader* [1, 46, 75].

In their monograph, Dwork and Roth [34] identified stability as a common factor of learning and DP: "Differential privacy is enabled by stability and ensures stability . . . we observe a tantalizing moral equivalence between learnability, differential privacy, and stability." This insight has found formal manifestations in several works. For example, Abernethy et al. [2] used DP-inspired stability methodology to derive a unified framework for proving state-of-the-art bounds in online learning. In the opposite direction, Agarwal and Singh [3] showed that certain standard stabilization techniques in online learning imply DP.

Stability plays a key role in this work as well. The direction that any class with a finite Littlestone dimension can be privately learned hinges on the following form of stability: for $\eta > 0$ and $n \in \mathbb{N}$, a learning algorithm $\mathcal{A}$ is (n, η)-*globally stable*[2] with respect to a distribution $\mathcal{D}$ over examples if there exists an hypothesis h whose frequency as an output is at least η. Namely,

$$\Pr_{S \sim \mathcal{D}^n}[\mathcal{A}(S) = h] \geq \eta.$$

Our argument follows by showing that every $\mathcal{H}$ can be learned by a globally stable algorithm with parameters $\eta = 2^{-2^{O(d)}}, n = 2^{O(d)}$, where d is the Littlestone dimension of $\mathcal{H}$. As a corollary, we get an equivalence between global stability and DP (which can be viewed as a form of local stability). In other words, the existence of a globally stable learner for $\mathcal{H}$ is equivalent to the existence of a differentially private learner for it (and both are equivalent to having a finite Littlestone dimension).

Littlestone dimension and thresholds. The converse direction—that every DP-learnable class has a finite Littlestone dimension—utilizes an intimate relationship between thresholds and the Littlestone dimension: a class $\mathcal{H}$ has a finite Littlestone dimension if and only if it does not embed thresholds as a subclass (for a formal statement, see Theorem 10); this follows from a seminal result in model theory by Shelah [76]. As explained in the preliminaries (Section 3), Shelah's theorem is usually stated in terms of orders and ranks. Chase and Freitag [25] noticed[3] that the Littlestone dimension is the same as the model-theoretic rank. Meanwhile, order translates naturally to thresholds. To make Theorem 10 more accessible for readers with less background in model theory, we provide a combinatorial proof in the appendix.

[1] More precisely, there is a deterministic algorithm that makes no more than d mistakes, and for every deterministic algorithm there is a (realizable) input sequence on which it makes at least d mistakes. For randomized algorithms, a slightly weaker lower bound of $d/2$ holds with respect to the expected number of mistakes.

[2] The word *global* highlights a difference with other forms of algorithmic stability. Indeed, previous forms of stability such as DP and *uniform hypothesis stability* [15] are local in the sense that they require output robustness subject to *local* changes in the input. However, the property required by global stability captures stability with respect to resampling the entire input.

[3] Interestingly, although the Littlestone dimension is a basic parameter in machine learning, this result has not appeared in the machine learning literature.

Littlestone classes. It is natural to ask which classes have finite Littlestone dimension. First, note that every finite class $\mathcal{H}$ has Littlestone dimension $d \leq \log|\mathcal{H}|$. There are also many natural and interesting infinite classes with finite Littlestone dimension. For example, let $X = \mathbb{F}^n$ be an n-dimensional vector space over a field $\mathbb{F}$ and let $\mathcal{H} \subseteq \{\pm 1\}^X$ consist of all (indicators of) affine subspaces of dimension $\leq d$. The Littlestone dimension of $\mathcal{H}$ is d. More generally, any class of hypotheses that can be described by polynomial *equalities* of bounded degree has a bounded Littlestone dimension.[4] This can be generalized even further to classes that are definable in *stable theories*. This (different, still) notion of stability is deep and well explored in model theory. We refer the reader to Section 5.1 in the work of Chase and Freitag [26] for more examples of stable theories and the Littlestone classes they correspond to.

Organization. The rest of this article is organized as follows. In Section 2, we formally state our main results and discuss some implications and other related and subsequent work. We present the preliminaries in Section 3. Then, in Section 4, we prove the direction that differentially private learnable classes have a finite Littlestone dimension, and in Section 5, we prove the converse direction—that every Littlestone class is differentially private PAC learnable. Finally, Section 6 concludes the article with some suggestions for future work.

2 RESULTS

We next present our main results that yield an equivalence between private PAC learning and online learning. We note that the derived equivalence is *qualitative* in the sense that the gap between the best-known lower and upper bounds for learning a class $\mathcal{H}$ is incredibly large: the lower bound is proportional to $\log^*(d)$, whereas the upper bound is doubly exponential in d, where d is the Littlestone dimension of $\mathcal{H}$. Our upper bound has recently been reduced to $\tilde{O}(d^6)$ in subsequent work [40].

The rest of this section is organized as follows: Sections 2.1, 2.2, and 2.3 are dedicated to the relationship between differentially private learning, Littlestone dimension, and online learning, and in Section 2.4. we discuss an implication for private boosting. Throughout this section, some standard technical terms are used. For definitions of these terms, we refer the reader to Section 3.

2.1 Private Learning Implies Finite Littlestone Dimension

We begin by the following statement that resolves an open problem in the work of Feldman and Xiao [38] and Bun et al. [22].

Theorem 1 (Thresholds Are Not Privately Learnable). *Let $X \subseteq \mathbb{R}$ and let $\mathcal{A}$ be a $(\frac{1}{16}, \frac{1}{16})$-accurate learning algorithm for the class of thresholds over X with sample complexity n that satisfies (ε, δ)-DP with $\varepsilon = 0.1$ and $\delta = O(\frac{1}{n^2 \log n})$. Then,*

$$n \geq \Omega(\log^* |X|).$$

In particular, the class of thresholds over an infinite X cannot be learned privately.

We note that an upper bound which scales with $(\log^* |X|)^{\frac{3}{2}}$ on the private sample complexity of learning thresholds over a domain of size n is given by Kaplan et al. [53]. Thus, Theorem 1 is tight up to polynomial factors. A weaker version of Theorem 1 by Bun et al. [22] provides a similar lower bound but applies only to proper learning algorithms.

[4] Note that if one replaces “equalities” with “inequalities,” then the Littlestone dimension may become unbounded while the VC dimension remains bounded. This is demonstrated, for example, by halfspaces that are captured by polynomial inequalities of degree 1.

Theorems 1 and 10 (which is stated in Section 3) imply that any privately learnable class has a finite Littlestone dimension.

Theorem 2 (Private Learning Implies Finite Littlestone Dimension). *Let H be an hypothesis class with Littlestone dimension $d \in \mathbb{N} \cup \{\infty\}$ and let $\mathcal{A}$ be a $(\frac{1}{16}, \frac{1}{16})$-accurate learning algorithm for H with sample complexity n that satisfies (ε, δ)-differentially private with $\varepsilon = 0.1$ and $\delta = O(\frac{1}{n^2 \log n})$. Then,*

$$n \geq \Omega(\log^* d).$$

In particular, any class that is privately learnable has a finite Littlestone dimension.

2.1.1 On the Proof of Theorem 1. A common approach of proving impossibility results in computer science (and in machine learning in particular) exploits a Minmax principle, whereby one specifies a fixed hard distribution over inputs, and establishes the desired impossibility result for any algorithm with respect to random inputs from that distribution. As an example, consider the "No-Free-Lunch Theorem," which establishes that the VC dimension lower bounds the sample complexity of PAC learning a class $\mathcal{H}$. Here, the hard distribution is picked to be uniform on a shattered set of size $d = \mathsf{VC}(H)$, and the argument follows by showing that every learning algorithm must observe $\Omega(d)$ examples. (For example, see Theorem 5.1 in the work of Shalev-Shwartz and Singer [74].)

Such "Minmax" proofs establish a stronger assertion: they apply even to algorithms that "know" the input distribution. For example, the No-Free-Lunch Theorem applies even to learning algorithms that are designed given the knowledge that the marginal distribution is uniform over some shattered set.

Interestingly, such an approach is bound to fail in proving Theorem 1. The reason is that if the marginal distribution D_X is fixed, then one can pick an $\epsilon/2$-cover,[5] which we denote by $C_{\varepsilon/2}$, for the class thresholds over X of size $|C_{\varepsilon/2}| = O(1/\epsilon)$, and use the *exponential mechanism* [64] to DP-learn the finite class $C_{\varepsilon/2}$ with sample complexity that scales with $\log|C_{\varepsilon/2}| = O(\log(1/\varepsilon))$. Since $C_{\varepsilon/2}$ is an ε-cover for the class of thresholds, the obtained algorithm PAC learns the class of thresholds in a differentially private manner. To conclude, there is no single distribution that is "hard" for all DP algorithms that learn thresholds.

To overcome this difficulty, one must come up with a method of assigning to any given algorithm A a "hard" distribution $D = D_A$ that is tailored to A and witnesses Theorem 1 with respect to A. The challenge is that A can be arbitrary—for example, it may be improper.[6] We refer the reader elsewhere [7, 67, 68] for a line of work that explores in detail a similar "failure" of the Minmax principle in the context of PAC learning with low mutual information.

The "method" we use to prove Theorem 1 exploits Ramsey theory. In a nutshell, Ramsey theory provides tools that allow to detect, for any learning algorithm, a "largish"' set $X' \subseteq X$ such that the behavior of A on input samples from X' is highly regular. Then, the uniform distribution over X' is the "hard" distribution that is used to derive Theorem 1.

We note that similar applications of Ramsey theory in computer science date back to the 1980s [65]. For more recent usages, see other works [23, 27, 28].

Finally, we note that in the proper case, Bun et al. [22] demonstrated an *ensemble*, namely a distribution over distributions, which is hard for every differentially private algorithm A: if one draws a random distribution D from the ensemble and runs A on an input sample from D, then the expected error of A will be large. It is plausible that such a statement also holds for a general (possibly improper) algorithm, and it would be interesting to find such a natural ensemble.

[5] In other words, $C_{\varepsilon/2}$ satisfies that for every threshold h there exists $c \in C_{\varepsilon/2}$ such that $\Pr_{x \sim D_X}(c(x) \neq h(x)) \leq \epsilon/2$.

[6] In other words, it may output hypotheses that are not thresholds.

2.2 Finite Littlestone Dimension Implies Private Learning

The following statement provides an upper bound on the sample complexity of DP-learning $\mathcal{H}$, which depends only on the Littlestone dimension of $\mathcal{H}$ and the privacy/utility parameters. In particular, it does not depend on $|\mathcal{H}|$.

Theorem 3 (Littlestone Classes are Privately Learnable). *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class with Littlestone dimension d, let $\varepsilon, \delta \in (0,1)$ be privacy parameters, and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For*

$$n = O\left(\frac{2^{\tilde{O}(2^d)} + \log 1/\beta\delta}{\alpha\epsilon}\right) = O_d\left(\frac{\log(1/\beta\delta)}{\alpha\varepsilon}\right)$$

there exists an (ε,δ)-DP learning algorithm such that for every realizable distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies $\operatorname{loss}_{\mathcal{D}}(f) \le \alpha$ with probability at least $1-\beta$, where the probability is taken over $S \sim \mathcal{D}^n$ as well as the internal randomness of $\mathcal{A}$.

A similar result holds in the agnostic setting.

Corollary 4 (Agnostic Learner for Littlestone Classes). *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class with Littlestone dimension d, let ε and $\delta \in (0,1)$ be privacy parameters, and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For*

$$n = O\left(\frac{2^{\tilde{O}(2^d)} + \log(1/\beta\delta)}{\alpha\epsilon} + \frac{VC(\mathcal{H}) + \log(1/\beta)}{\alpha^2\epsilon}\right)$$

there exists an (ε,δ)-DP learning algorithm such that for every distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies

$$\operatorname{loss}_{\mathcal{D}}(f) \le \min_{h\in\mathcal{H}} \operatorname{loss}_{\mathcal{D}}(h) + \alpha$$

with probability at least $1-\beta$, where the probability is taken over $S \sim \mathcal{D}^n$ as well as the internal randomness of $\mathcal{A}$.

Corollary 4 follows from Theorem 3 by Theorem 2.3 in [4] which provides a general mechanism to transform a learner in the realizable setting to a learner in the agnostic setting.[7] We note that formally the transformation in the work of Alon et al. [4] is stated for a constant $\varepsilon = O(1)$. Taking $\varepsilon = O(1)$ is without loss of generality, as a standard "secrecy-of-the-sample" argument can be used to convert this learner into one that is (ε,δ)-differentially private by increasing the sample size by a factor of roughly $1/\varepsilon$ and running the algorithm on a random subsample. See other works [57, 77] for further details.

2.3 Online Learning Versus Differentially Private PAC Learning

Since the Littlestone dimension characterizes online learnability [13, 61], Theorems 2 and 3 imply an equivalence between differentially private PAC learning and online learning.

Theorem 5 (Private PAC Learning ≡ Online Prediction). *The following statements are equivalent for a class $\mathcal{H} \subseteq \{\pm 1\}^X$:*

(1) *$\mathcal{H}$ is online learnable.*
(2) *$\mathcal{H}$ is approximate differentially privately PAC learnable.*

[7] Theorem 2.3 of Alon et al. [4] is based on a previous realizable-to-agnostic transformation from Beimel et al. [10] that applies to *proper* learners. Here we require the more general transformation from Alon et al. [4], as the learner implied by Theorem 3 may be improper.

Theorem 5 directly follows from Theorem 2 (which gives 2 → 1) and Theorem 3 (which gives 1 → 2). We comment that a quantitative relation between the learning rates and mistake/regret bounds is also implied—for example, in the agnostic setting, it is known that the optimal regret bound for $\mathcal{H}$ is $\tilde{\Theta}_d(\sqrt{T})$, where the $\tilde{\Theta}_d$ conceals a constant that depends on the Littlestone dimension of $\mathcal{H}$ [13]. Similarly, we get that the optimal sample complexity of agnostically privately learning $\mathcal{H}$ is $\Theta_d(\frac{\log(1/(\beta\delta))}{\alpha^2\varepsilon})$.

We remark, however, that the preceding equivalence is mostly interesting from a theoretical perspective, and should not be regarded as an efficient transformation between online and private learning. Indeed, the Littlestone dimension dependencies concealed by the $\tilde{\Theta}_d(\cdot)$ in the preceding bounds on the regret and sample complexities may be quite different from one another. For example, there are classes for which the $\Theta_d(\frac{\log(1/(\beta\delta))}{\alpha\varepsilon})$ bound hides a $\text{poly}(\log^*(d))$ dependence, and the $\tilde{\Theta}_d(\sqrt{T})$ bound hides a $\Theta(d)$ dependence. One example that attains both of these dependencies is the class of thresholds over a linearly ordered domain of size 2^d [53].

2.3.1 Global Stability. Our proof of Theorem 3 hinges on an intermediate property that we call *global stability.*

Definition 6 (Global Stability). Let $n \in \mathbb{N}$ be a sample size and $\eta > 0$ be a global stability parameter. An algorithm $\mathcal{A}$ is (n, η)-globally stable with respect to a distribution $\mathcal{D}$ if there exists an hypothesis h such that

$$\Pr_{S\sim\mathcal{D}^n}[A(S) = h] \geq \eta.$$

Although global stability is a rather strong property, it holds automatically for learning algorithms using a finite hypothesis class. By an averaging argument, every learner using n samples that produces a hypothesis in a finite hypothesis class $\mathcal{H}$ is $(n, 1/|\mathcal{H}|)$-globally stable. The following proposition generalizes "Occam's Razor"for finite hypothesis classes to show that global stability is enough to imply similar generalization bounds in the realizable setting.

Proposition 7 (Global Stability $\implies$ Generalization). *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class, and assume that $\mathcal{A}$ is a consistent learner for $\mathcal{H}$ (i.e., $\text{loss}_S(\mathcal{A}(S)) = 0$ for every realizable sample S). Let $\mathcal{D}$ be a realizable distribution such that $\mathcal{A}$ is (n, η)-globally stable with respect to $\mathcal{D}$, and let h be a hypothesis such that $\Pr_{S\sim\mathcal{D}^n}[A(S) = h] \geq \eta$, as guaranteed by the definition of global stability. Then,*

$$\text{loss}_{\mathcal{D}}(h) \leq \frac{\ln(1/\eta)}{n}.$$

Proof. Let α denote the loss of h (i.e., $\text{loss}_{\mathcal{D}}(h) = \alpha$), and let E_1 denote the event that h is consistent with the input sample S. Thus, $\Pr[E_1] = (1-\alpha)^n$. Let E_2 denote the event that $\mathcal{A}(S) = h$. By assumption, $\Pr[E_2] \geq \eta$. Now, since $\mathcal{A}$ is consistent we get that $E_2 \subseteq E_1$, and hence that $\eta \leq (1-\alpha)^n$. This finishes the proof (using the fact that $1 - \alpha \leq e^{-\alpha}$ and taking the logarithm of both sides). □

Another way to view global stability is in the context of *pseudo-deterministic algorithms* [39]. A pseudo-deterministic algorithm is a randomized algorithm that yields some fixed output with high probability. Thinking of a realizable distribution $\mathcal{D}$ as an instance on which the PAC-learning algorithm has oracle access, a globally stable learner is one that is "weakly" pseudo-deterministic in that it produces some fixed output with probability bounded away from zero. A different model of pseudo-deterministic learning, in the context of learning from membership queries, was defined and studied by Oliveira and Santhanam [70].

We prove Theorem 3 by constructing, for a given Littlestone class $\mathcal{H}$, an algorithm $\mathcal{A}$ that is globally stable with respect to *every* realizable distribution.

2.4 Boosting for Approximate DP

Our characterization of private learnability in terms of the Littlestone dimension has new consequences for boosting the privacy and accuracy guarantees of differentially private learners. Specifically, it shows that the existence of a learner with weak (but non-trivial) privacy and accuracy guarantees implies the existence of a learner with any desired privacy and accuracy parameters—in particular, one with $\delta(n) = \exp(-\Omega(n))$.

Theorem 8. *There exists a constant $c > 0$ for which the following holds. Suppose that for some sample size n_0 there is an $(\varepsilon_0, \delta_0)$-differentially private learner $\mathcal{W}$ for a class $\mathcal{H}$ satisfying the guarantee*

$$\Pr_{S\sim\mathcal{D}^{n_0}}[\mathrm{loss}_{\mathcal{D}}(\mathcal{W}(S)) > \alpha_0] < \beta_0$$

for $\varepsilon_0 = 0.1, \alpha_0 = \beta_0 = 1/16$, and $\delta_0 \le c/n_0^2 \log n_0$.

Then there exists a constant $C_{\mathcal{H}}$ such that for every $\alpha, \beta, \varepsilon, \delta \in (0,1)$ there exists an (ε, δ)-differentially private learner for $\mathcal{H}$ with

$$\Pr_{S\sim\mathcal{D}^{n}}[\mathrm{loss}_{\mathcal{D}}(\mathcal{A}(S)) > \alpha] < \beta$$

whenever $n \ge C_{\mathcal{H}} \cdot \log(1/\beta\delta)/\alpha\varepsilon$.

Given a weak learner $\mathcal{W}$ as in the statement of Theorem 8, Theorem 2 imply that $\mathrm{Ldim}(\mathcal{H})$ is finite. Hence, Theorem 3 allows us to construct a learner for $\mathcal{H}$ with arbitrarily small privacy and accuracy, yielding Theorem 8. The constant $C_{\mathcal{H}}$ in the last line of the theorem statement suppresses a factor depending on $\mathrm{Ldim}(\mathcal{H})$.

Prior to our work, it was open whether arbitrary learning algorithms satisfying approximate DP could be boosted in this strong a manner. We remark, however, that in the case of *pure* DP, such boosting can be done algorithmically and efficiently. Specifically, given an $(\varepsilon_0, 0)$-differentially private weak learner as in the statement of Theorem 8, one can first apply random sampling to improve the privacy guarantee to $(p\varepsilon_0, 0)$-DP at the expense of increasing its sample complexity to roughly n_0/p for any $p \in (0,1)$. The Boosting-for-People construction of Dwork et al. [36] (also see the work of Bun et al. [18]) then produces a strong learner by making roughly $T \approx \log(1/\beta)/\alpha^2$ calls to the weak learner. By composition of DP, this gives an $(\varepsilon, 0)$-differentially private strong learner with sample complexity roughly $n_0 \cdot \log(1/\beta)/\alpha^2\varepsilon$.

What goes wrong if we try to apply this argument using an $(\varepsilon_0, \delta_0)$-differentially private weak learner? Random sampling still gives a $(p\varepsilon_0, p\delta_0)$-differentially private weak learner with sample complexity n_0/p. However, this is not sufficient to improve the δ parameter of the learner *as a function of the number of samples n*. Thus, the strong learner one obtains using Boosting-for-People still at best guarantees $\delta(n) = \tilde{O}(1/n^2)$. Meanwhile, Theorem 8 shows that the existence of a $(0.1, \tilde{O}(1/n^2))$-differentially private learner for a given class implies the existence of a $(0.1, \exp(-\Omega(n))$-differentially private learner for that class.

We leave it as an interesting open question to determine whether this kind of boosting for approximate DP can be done algorithmically.

2.5 Related and Subsequent Work

In this work, we determine that the (approximately) differentially-privately learnable classes are exactly those that are online learnable. We note that PAC learnability under the much stronger constraint of *pure* differentially privacy has already been characterized by several natural parameters such as the *probabilistic representation dimension* [12] and *one-way communication complexity* [38]. These characterizations even imply nearly tight bounds on the optimal sample complexity. This is in contrast with the equivalence derived in this work whose implied upper and lower bounds on the sample complexity are extremely far away from each other.

Subsequent to our work, Ghazi et al. [40] gave a significantly improved upper bound of $\tilde{O}(d^6)$ on the sample complexity of learning any class with Littlestone dimension d. Moreover, their learning algorithm is proper. There is still an enormous gap between this and our lower bound of $\Omega(\log^* d)$, but both the upper and lower bound are within polynomial factors of the best possible sample complexity bounds that depend *only* on the Littlestone dimension. Thus, despite the fact that DP learnability is characterized by the finiteness of the Littlestone dimension, it remains wide open to find meaningful quantitative bounds on the sample complexity of DP learning. This is discussed in more detail in Section 5.4, where we suggest directions for future work.

Subsequent work has also extended the connection between online learning, global stability, and private learning to settings beyond binary classification. The private learnability of Littlestone classes has been studied in multiclass classification [20, 50], real-valued classification (regression) [42, 50], quantum state learning [6], and the online private learning model [43].

Ghazi et al. [41] used a generalization of global stability to derive private learning algorithms for datasets where each individual contributes multiple samples. Global stability is also related to a definition of reproducibility for machine learning algorithms put forth by Impagliazzo et al. [48].

Finally, several works have studied the question of whether computationally efficient reductions exist between online and private learning. Gonen et al. [44] gave an efficient compiler from low sample complexity pure private learners to online learners, whereas Bun [17] showed that under cryptographic assumptions, such a reduction cannot exist in general.

3 PRELIMINARIES

3.1 PAC Learning

We use standard notation from statistical learning (e.g., see [74]). Let X be any “domain” set and consider the “label” set $Y = \{\pm 1\}$. A *hypothesis* is a function $h : X \to Y$, which we alternatively write as an element of Y^X. An *example* is a pair $(x, y) \in X \times Y$. A *sample* S is a finite sequence of examples. We also use the following notation: for samples S, T, let $S \circ T$ denote the combined sample obtained by appending T to the end of S.

Definition 9 (Population and Empirical Loss). Let $\mathcal{D}$ be a distribution over $X \times \{\pm 1\}$. The population loss of a hypothesis $h : X \to \{\pm 1\}$ is defined by

$$\mathrm{loss}_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

Let $S = ((x_i, y_i))_{i=1}^n$ be a sample. The empirical loss of h with respect to S is defined by

$$\mathrm{loss}_S(h) = \frac{1}{n}\sum_{i=1}^{n} 1[h(x_i) \neq y_i].$$

Let $\mathcal{H} \subseteq Y^X$ be a *hypothesis class.* A sample S is said to be *realizable by* $\mathcal{H}$ if there is $h \in H$ such that $\mathrm{loss}_S(h) = 0$. A distribution $\mathcal{D}$ is said to be *realizable by* $\mathcal{H}$ if there is $h \in H$ such that $\mathrm{loss}_{\mathcal{D}}(h) = 0$. A *learning algorithm* A is a (possibly randomized) mapping taking input samples to output hypotheses. We denote by $A(S)$ the distribution over hypotheses induced by the algorithm when the input sample is S. We say that A *learns*[8] *a class* $\mathcal{H}$ with α*-error*, $(1-\beta)$*-confidence*, and *sample complexity* m if for every realizable distribution $\mathcal{D}$,

$$\Pr_{S\sim D^m,\ h\sim A(S)}[\mathrm{loss}_{\mathcal{D}}(h) > \alpha] \leq \beta.$$

[8] We focus on the realizable case.

For brevity, if A is a learning algorithm with α-error and $(1-\beta)$-confidence, we will say that A is an (α,β)-*accurate learner.*

3.2 Online Learning

Littlestone dimension. The Littlestone dimension is a combinatorial parameter that captures mistake and regret bounds in online learning [13, 61].[9] Its definition uses the notion of *mistake trees.* A mistake tree is a binary decision tree whose internal nodes are labeled by elements of X. Any root-to-leaf path in a mistake tree can be described as a sequence of examples $(x_1,y_1),\ldots,(x_d,y_d)$, where x_i is the label of the i'th internal node in the path, and $y_i=+1$ if the $(i+1)$'th node in the path is the right child of the i'th node and $y_i=-1$ otherwise. We say that a mistake tree T is *shattered* by $\mathcal{H}$ if for any root-to-leaf path $(x_1,y_1),\ldots,(x_d,y_d)$ in T there is an $h\in\mathcal{H}$ such that $h(x_i)=y_i$ for all $i\le d$ (Figure 1). The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{Ldim}(\mathcal{H})$, is the depth of largest complete tree that is shattered by $\mathcal{H}$. We say that $\mathcal{H}$ is a Littlestone class if it has finite Littlestone dimension.

Littlestone dimension and thresholds. Recently, Chase and Freitag [25] noticed that the Littlestone dimension coincides with a model-theoretic measure of complexity—Shelah's 2-rank.

A classical theorem of Shelah connects bounds on 2-rank (Littlestone dimension) to bounds on the so-called order property in model theory. The order property corresponds naturally to the concept of *thresholds.* Let $\mathcal{H}\subseteq\{\pm1\}^X$ be an hypothesis class. We say that $\mathcal{H}$ *contains k thresholds* if there are $x_1,\ldots,x_k\in X$ and $h_1,\ldots,h_k\in\mathcal{H}$ such that $h_i(x_j)=1$ if and only if $i\le j$ for all $i,j\le k$.

Shelah's result (part of the so-called Unstable Formula Theorem[10]) [47, 76], which we use in the following translated form, provides a simple and elegant connection between Littlestone dimension and thresholds.

Theorem 10 (Littlestone dimension and thresholds [47, 76]). *Let $\mathcal{H}$ be an hypothesis class, then*

(1) *If the* $\mathrm{Ldim}\,\mathcal{H}\ge d$, *then* $\mathcal{H}$ *contains* $\lfloor\log d\rfloor$ *thresholds.*
(2) *If* $\mathcal{H}$ *contains d thresholds, then its* $\mathrm{Ldim}\,\mathcal{H}\ge\lfloor\log d\rfloor$.

For completeness, we provide a combinatorial proof of Theorem 10 in Appendix A.

In the context of model theory, Theorem 10 is used to establish an equivalence between finite Littlestone dimension and *stable theories.* It is interesting to note that an analogous connection between theories that are called *NIP theories* and VC dimension has also been previously observed and was pointed out by Laskowski [59]; this in turn led to results in learning theory, particularly within the context of compression schemes [62] but also some of the first polynomial bounds for the VC dimension for sigmoidal neural networks [56].

Mistake bound and the Standard Optimal Algorithm. The simplest setting in which learnability is captured by the Littlestone dimension is called the *mistake-bound model* [61]. Let $\mathcal{H}\subseteq\{\pm1\}^X$ be a fixed hypothesis class known to the learner. The learning process takes place in a sequence of trials, where the order of events in each trial t is as follows:

(i) The learner receives an instance $x_t\in X$,
(ii) The learner responses with a prediction $\hat{y}_t\in\{\pm1\}$, and
(iii) The learner is told whether or not the responds was correct.

[9] It appears that the name "Littlestone dimension" was coined in the work of Ben-David et al. [13].

[10] Shelah [76] provides a qualitative statement, and a quantitative one that is more similar to Theorem 10 can be found in the work of Hodges [47].

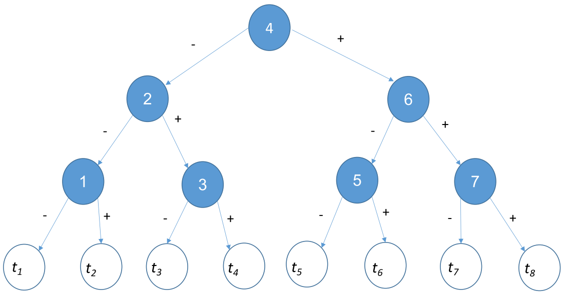

Fig. 1. A tree shattered by the class $\mathcal{H} \subseteq \{\pm 1\}^8$ that contains the threshold functions t_i, where $t_i(j) = +1$ if and only if $i \leq j$.

We assume that the examples given to the learner are realizable in the following sense: for the entire sequence of trials, there is a hypothesis $h \in \mathcal{H}$ such that $y_t = h(x_t)$ for every instance x_t and correct response y_t. An algorithm in this model learns $\mathcal{H}$ with mistake bound M if for every realizable sequence of examples presented to the learner it makes a total of at most M incorrect predictions.

Littlestone [61] showed that the minimum mistake bound achievable by any online learner is exactly $\mathrm{Ldim}(\mathcal{H})$. Furthermore, he described an explicit algorithm, called the **Standard Optimal Algorithm** (SOA), which achieves this optimal mistake bound.

Standard Optimal Algorithm (SOA)

```
(1) Initialize $\mathcal{H}_1 = \mathcal{H}$.
(2) For trials $t = 1, 2, \ldots$:
    (i) For each $b \in \{\pm 1\}$ and $x \in X$, let $\mathcal{H}_t^b(x) = \{h \in \mathcal{H}_t : h(x) = b\}$. Define $h : X \to \{\pm 1\}$ by $h_t(x) = \mathrm{argmax}_b \, \mathrm{Ldim}(\mathcal{H}_t^b(x))$.
    (ii) Receive instance $x_t$.
    (iii) Predict $\hat{y}_t = h_t(x_t)$.
    (iv) Receive correct response $y_t$.
    (v) Update $\mathcal{H}_{t+1} = \mathcal{H}_t^{y_t}(x_t)$.
```

Extending the SOA *to non-realizable sequences.* Our globally stable learner for Littlestone classes will make use of an optimal online learner in the mistake bound model. For concreteness, we pick the SOA (any other optimal algorithm will also work). It will be convenient to extend the SOA to sequences that are not necessarily realizable by a hypothesis in $\mathcal{H}$. We will use the following simple extension of the SOA to non-realizable samples.

Definition 11 (Extending the SOA *to Non-realizable Sequences).* Consider a run of the SOA on examples $(x_1, y_1), \ldots, (x_m, y_m)$, and let h_t denote the predictor used by the SOA after seeing the first t examples (i.e., h_t is the rule used by the SOA to predict in the $(t+1)$'st trial). Then, after observing both x_{t+1}, y_{t+1} do the following:

- If the sequence $(x_1, y_1), \ldots, (x_{t+1}, y_{t+1})$ is realizable by some $h \in \mathcal{H}$, then apply the usual update rule of the SOA to obtain h_{t+1}.
- Else, set h_{t+1} as follows: $h_{t+1}(x_{t+1}) = y_{t+1}$, and $h_{t+1}(x) = h_t(x)$ for every $x \neq x_{t+1}$.

Thus, upon observing a non-realizable sequence, this update rule locally updates the maintained predictor h_t to agree with the last example.

3.3 Differential Privacy

We use standard definitions and notation from the DP literature. For more background, see surveys found elsewhere [34, 77]. For $a, b, \varepsilon, \delta \in [0,1]$, let $a \approx_{\varepsilon,\delta} b$ denote the statement

$$a \leq e^{\varepsilon} b + \delta \text{ and } b \leq e^{\varepsilon} a + \delta.$$

We say that two probability distributions p, q are (ε, δ)*-indistinguishable* if $p(E) \approx_{\varepsilon,\delta} q(E)$ for every event E.

Definition 12 (Private Learning Algorithm). A randomized algorithm

$$A : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$$

is (ε, δ)-differentially private if for every two samples $S, S' \in (X \times \{\pm 1\})^n$ that disagree on a single example the output distributions $A(S)$ and $A(S')$ are (ε, δ)-indistinguishable.

We emphasize that (ε, δ)-indistinguishability must hold for every such pair of samples, even if they are not generated according to a (realizable) distribution.

The parameters ε, δ are usually treated as follows: ε is a small constant (say 0.1), and δ is negligible, $\delta = n^{-\omega(1)}$, where n is the input sample size. The case of $\delta = 0$ is also referred to as *pure DP*. Thus, a class $\mathcal{H}$ is privately learnable if it is PAC learnable by an algorithm A that is $(\varepsilon(n), \delta(n))$-differentially private with $\varepsilon(n) \leq 0.1$, and $\delta(n) \leq n^{-\omega(1)}$.

We will use the following corollary of the *Basic Composition Theorem* from DP (e.g., see Theorem 3.16 in the work of Dwork and Roth [35]).

Lemma 13 (Composition [31, 32]). *If p, q are (ε, δ)-indistinguishable, then for all $k \in \mathbb{N}$, p^k and q^k are $(k\varepsilon, k\delta)$-indistinguishable, where p^k, q^k are the k-fold products of p, q (i.e., corresponding to k independent samples).*

Private empirical learners. For the proof of Theorem 1, it will be convenient to consider the following task of minimizing the empirical loss.

Definition 14 (Empirical Learner). Algorithm A is (α, β)-accurate empirical learner for a hypothesis class $\mathcal{H}$ with sample complexity m if for every $h \in \mathcal{H}$ and for every sample $S = ((x_1, h(x_1), \ldots, (x_m, h(x_m))) \in (X \times \{\pm 1\})^m$ the algorithm A outputs a function f satisfying

$$\Pr_{f \sim A(S)} (\text{loss}_S(f) \leq \alpha) \geq 1 - \beta.$$

This task is simpler to handle than PAC learning, which is a distributional loss minimization task. Replacing PAC learning by this task does not lose generality; this is implied by the following result of Bun et al. [22].

Lemma 15 ([22], Lemma 5.9). *Suppose $\varepsilon < 1$ and A is an (ϵ, δ)-differentially private (α, β)-accurate learning algorithm for a hypothesis class $\mathcal{H}$ with sample complexity m. Then there exists an (ϵ, δ)–differentially private (α, β)-accurate empirical learner for $\mathcal{H}$ with sample complexity $9m$.*

3.4 Additional Notation

A sample S of an even length is called *balanced* if half of its labels are $+1$'s and half are -1's.

For a sample S, let S_X denote the underlying set of unlabeled examples: $S_X = \{x | (\exists y) : (x, y) \in S\}$. Let A be a randomized learning algorithm. It will be convenient to associate with A and S the function $A_S : X \to [0,1]$ defined by

$$A_S(x) = \Pr_{h \sim A(S)} \Big[h(x) = 1 \Big].$$

Intuitively, this function represents the average hypothesis outputted by A when the input sample is S.

For the next definitions assume that the domain X is linearly ordered. Let $S = ((x_i, y_i))_{i=1}^m$ be a sample. We say that S is *increasing* if $x_1 < x_2 < \cdots < x_m$. For $x \in X$, define $\mathrm{ord}_S(x)$ by $|\{i | x_i \leq x\}|$. Note that the set of points $x \in X$ with the same $\mathrm{ord}_S(x)$ form an interval whose endpoints are two consecutive examples in S (consecutive with respect to the order on X, i.e., there is no example x_i between them).

The *tower function* $\mathrm{twr}_k(x)$ is defined by the recursion

$$\mathrm{twr}^{(i)} x = \begin{cases} x & i = 1, \\ 2^{\mathrm{twr}(i-1)(x)} & i > 1. \end{cases}$$

The iterated logarithm, $\log^{(k)}(x)$ is defined by the recursion

$$\log^{(i)} x = \begin{cases} \log x & i = 0, \\ 1 + \log^{(i-1)} \log x & i > 0. \end{cases}$$

The function $\log^* x$ equals the number of times the iterated logarithm must be applied before the result is less than or equal to 1. It is defined by the recursion

$$\log^* x = \begin{cases} 0 & x \leq 1, \\ 1 + \log^* \log x & x > 1. \end{cases}$$

4 PRIVATE LEARNING IMPLIES FINITE LITTLESTONE DIMENSION

In this section, we prove that every class $\mathcal{H}$ that can be PAC learned by a DP algorithm has a finite Littlestone dimension. This is achieved by establishing a lower bound on the sample complexity of privately learning $\mathcal{H}$ that depends on its Littlestone dimension (Theorem 2). The crux of this lower bound lies in Theorem 1, which provides a lower bound for the task of privately learning 1-dimensional thresholds. This section is organized as follows. In Section 4.1, we provide an overview of the proof. Then, in Sections 4.2 and 4.3, we prove Theorems 1 and 2.

4.1 Proof Overview

The starting point of the proof is Theorem 10, which asserts that if $\mathcal{H}$ has Littlestone dimension d, then it contains, as a subclass, at least some $\log d$ thresholds. In other words, the class of thresholds is "complete" in the sense that a lower bound on the sample complexity of DP learning thresholds yields a lower bound for classes with a large Littlestone dimension.

Thus, consider an arbitrary differentially private algorithm A that learns the class of thresholds over an ordered domain X of size n. Our goal is to show a lower bound of $\Omega(\log^* n)$ on the sample complexity of A. A central challenge in the proof emerges because A may be improper and output arbitrary hypotheses (this is in contrast with proving impossibility results for proper algorithms where the structure of the learned class can be exploited).

The proof consists of two parts. The first part handles the preceding challenge by showing that for any algorithm (in fact, for any mapping that takes input samples to output hypotheses) there is a large subset of the domain that is *homogeneous with respect to the algorithm*. This notion of homogeneity places useful restrictions on the algorithm on input samples from the homogeneous

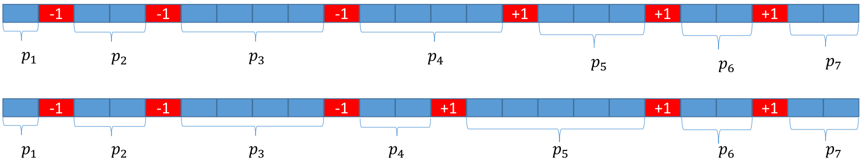

Fig. 2. Depiction of two possible outputs of an algorithm over an homogeneous set, given two input samples from the set (marked in red). The numbers p_i denote, for a given point x, the probability that $h(x) = 1$, where $h \sim A(S)$ is the hypothesis h outputted by the algorithm on input sample S. These probabilities depend (up to a small additive error) only on the interval that x belongs to. In the figure, we changed in the input the fourth example—this only affects the interval and not the values of the p_i's (again, up to a small additive error).

set. The second part of the argument utilizes the homogeneity of $X' \subseteq X$ to derive a lower bound on the sample complexity of the algorithm in terms of $|X'|$.

We note that the Ramsey argument in the first part is quite general: it does not use the definition of DP and could perhaps be useful in other sample complexity lower bounds. It is also worth noting that a Ramsey-based argument was used by Bun [23] in a weaker lower bound for DP learning thresholds in the proper case. In contrast to the first part, the second (and more technical) part of the proof is tailored specifically to the definition of DP. We next outline each of these two parts.

Reduction to homogeneous sets. As discussed earlier, the first step in the proof is about identifying a large homogeneous subset of the input domain X on which we can control the output of A. To define homogeneity, recall from Section 3.4 that a sample $S = ((x_i, y_i))_{i=1}^m$ of an even length is called *balanced* if half of its labels are +1's and half are −1's, and that S is said to be increasing if $x_1 < x_2 < \cdots < x_m$. Now, a subset $X' \subseteq X$ is called *homogeneous with respect to* A if there is a list of numbers $p_0, p_1, \ldots, p_m$ such that for every *increasing balanced sample* S of points from X' and for every x' from X' with $\operatorname{ord}_S(x') = i$,

$$|A_S(x') - p_i| \leq \gamma,$$

where γ is sufficiently small. For simplicity, in this proof overview we will assume that $\gamma = 0$. (In the proof, γ is some $O(1/m)$ (see Definition 16).) So, for example, if A is deterministic, then $h = A(S)$ is constant over each of the intervals defined by consecutive examples from S. Figure 2 presents an illustration.

The derivation of a large homogeneous set follows by a standard application of the Ramsey theorem for hypergraphs using an appropriate coloring (Lemma 17).

Lower bound for homogenous algorithms. We next assume that $X' = \{1, \ldots, k\}$ is a large homogeneous set with respect to A (with $\gamma = 0$). We will obtain a lower bound on the sample complexity of A, denoted by m, by constructing a family $\mathcal{P}$ of distributions such that (i) on the one hand, $|\mathcal{P}| \leq 2^{\tilde{O}(m^2)}$, and (ii) on the other hand, $|\mathcal{P}| \geq \Omega(k)$. Combining these inequalities yields a lower bound on m in terms of $|X'| = k$ and concludes the proof.

The construction of $\mathcal{P}$ proceeds as follows and is depicted in Figure 3: let S be an increasing balanced sample of points from X'. Using the fact that A learns thresholds, it is shown that for some $i_1 < i_2$ we have that $p_{i_1} \leq 1/3$ and $p_{i_2} \geq 2/3$. Thus, by a simple averaging argument, there is some $i_1 \leq i \leq i_2$ such that $p_i - p_{i-1} \geq \Omega(1/m)$.

The last step in the construction is done by picking an increasing sample S such that the interval (x_{i-1}, x_{i+1}) has size $n = \Omega(k)$. For $x \in (x_{i-1}, x_{i+1})$, let S_x denote the sample obtained by replacing x_i with x in S. By restricting the output hypothesis to the interval (x_{i-1}, x_{i+1}) (which is of size n),

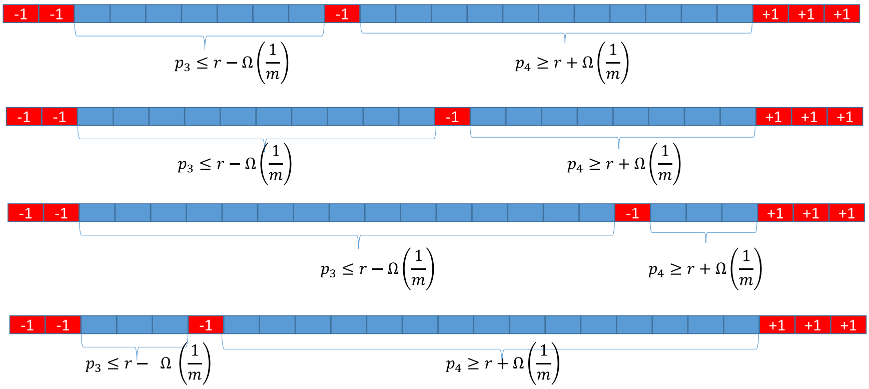
-1 -1
-1
+1 +1 +1
$p_3 \leq r - \Omega\left(\frac{1}{m}\right)$
$p_4 \geq r + \Omega\left(\frac{1}{m}\right)$

Fig. 3. An illustration of the definition of the family P. Given an homogeneous set and two consecutive intervals where there is a gap of at least $\Omega(1/m)$ between p_i and p_{i-1} (here, $i = 4$). The distributions in P correspond to the different positions of the i'th example, which separates between the $(i-1)$'th and the i'th intervals.

each output distribution $A(S_x)$ can be seen as a distribution over the cube $\{\pm 1\}^n$. Thus, the family of distributions $\mathcal{P}$ consists of all distributions $P_x = A(S_x)$ for $x \in (x_{i-1}, x_{i+1})$. Since A is private, it follows that $\mathcal{P}$ has the following two properties:

- $P_{x'}, P_{x''} \in \mathcal{P}$ are (ε, δ)-indistinguishable for all $x', x'' \in (x_{i-1}, x_{i+1})$, and
- Put $r = \frac{p_{i-1}+p_i}{2}$, then for all $P_x \in \mathcal{P}$,

$$(\forall x' \leq n) : \Pr_{h \sim P_x}\left[h(x') = 1\right] = \begin{cases} r - \Omega(1/m) & x' < x, \\ r + \Omega(1/m) & x' > x. \end{cases}$$

It remains to show that $\Omega(k) \leq |\mathcal{P}| \leq 2^{\tilde{O}(m^2)}$. The lower bound follows directly from the definition of $\mathcal{P}$. The upper bound requires a more subtle argument: it exploits the composition property for DP (see Lemma 13) via a privacy-breaching "attack" that is based on binary search. This argument appears in Lemma 21, whose proof is self-contained.

4.2 A Lower Bound for Privately Learning Thresholds

4.2.1 Proof of Theorem 1. The proof uses the following definition of homogeneous sets. Recall the definitions of balanced sample and of an increasing sample—in particular, that a sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ of an even size is realizable (by thresholds), balanced, and increasing if and only if $x_1 < x_2 < \cdots < x_m$ and the first half of the y_i's are -1 and the second half are $+1$.

Definition 16 (m-Homogeneous Set). A set $X' \subseteq X$ is *m-homogeneous* with respect to a learning algorithm A if there are numbers $p_i \in [0, 1]$, for $0 \leq i \leq m$ such that for every increasing balanced realizable sample $S \in (X' \times \{\pm 1\})^m$ and for every $x \in X' \setminus S_X$,

$$|A_S(x) - p_i| \leq \frac{1}{10^2 m},$$

where $i = \text{ord}_S(x)$. The list $(p_i)_{i=0}^m$ is referred to as the probabilities list of X' with respect to A.

Proof of Theorem 1. Let A be a $(1/16, 1/16)$-accurate learning algorithm that learns the class of thresholds over X with m examples and is (ε, δ)-differentially private with $\varepsilon = 0.1, \delta = \frac{1}{10^3 m^2 \log m}$. By Lemma 15, we may assume without loss of generality that A is an empirical learner with the same privacy and accuracy parameters and sample size that is at most nine times larger.

Theorem 1 follows from the next two lemmas, which we prove later.

Lemma 17 (Every Algorithm Has Large Homogeneous Sets). *Let A be a (possibly randomized) algorithm that is defined over input samples of size m over a domain $X \subseteq R$ with $|X| = n$. Then, there is a set $X' \subseteq X$ that is m-homogeneous with respect to A of size*

$$|X'| \geq \frac{\log^{(m)}(n)}{2^{O(m \log m)}}.$$

Lemma 17 allows us to focus on a large homogeneous set with respect to A. The next lemma implies a lower bound in terms of the size of a homogeneous set. For simplicity and without loss of generality, assume that the homogeneous set is $\{1, \ldots, k\}$.

Lemma 18 (Large Homogeneous Sets Imply Lower Bounds for Private Learning). *Let A be an $(0.1, \delta)$-differentially private algorithm with sample complexity m and $\delta \leq \frac{1}{10^3 m^2 \log m}$. Let $X = \{1, \ldots, k\}$ be m-homogeneous with respect to A. Then, if A empirically learns the class of thresholds over X with $(1/16, 1/16)$-accuracy, then*

$$k \leq 2^{O(m^2 \log^2 m)}$$

(i.e., $m \geq \Omega(\frac{\sqrt{\log k}}{\log \log k})$).

With these lemmas in hand, Theorem 1 follows by a short calculation: indeed, Lemma 17 implies the existence of an homogeneous set X' with respect to A of size $k \geq \log^{(m)}(n)/2^{O(m \log m)}$. We then restrict A to input samples from the set X', and by relabeling the elements of X' assume that $X' = \{1, \ldots, k\}$. Lemma 18 then implies that $k = 2^{O(m^2 \log^2 m)}$. Together, we obtain that

$$\log^{(m)}(n) \leq 2^{c \cdot m^2 \log m}$$

for some constant $c > 0$. Applying the iterated logarithm $t = \log^*(2^{c \cdot m^2 \log m}) = \log^*(m) + O(1)$ times on the inequality yields that

$$\log^{(m+t)}(n) = \log^{(m + \log^*(m) + O(1))}(n) \leq 1,$$

and therefore $\log^*(n) \leq \log^*(m) + m + O(1)$, which implies that $m \geq \Omega(\log^* n)$ as required. □

4.2.2 Proof of Lemma 17. We next prove that every learning algorithm has a large homogeneous set. We will use the following quantitative version of the Ramsey theorem due to Erdös and Rado [37] (see also the book by Graham et al. [45] or Theorem 10.1 in the survey by Mubayi and Suk [66]).

Theorem 19 ([37]). *Let $s > t \geq 2$ and q be integers, and let*

$$N \geq \mathrm{twr}_t(3sq \log q).$$

Then for every coloring of the subsets of size t of a universe of size N using q colors, there is a homogeneous subset[11] *of size s.*

Proof of Lemma 17. Define a coloring on the $(m+1)$-subsets of X as follows. Let $D = \{x_1 < x_2 < \cdots < x_{m+1}\}$ be an $(m+1)$-subset of X. For each $i \leq m+1$, let $D^{-i} = D \setminus \{x_i\}$, and let S^{-i} denote the balanced increasing sample on D^{-i}. Set p_i to be the fraction of the form $\frac{t}{10^2 m}$ that is closest to $A_{S^{-i}}(x_i)$ (in case of ties, pick the smallest such fraction). The coloring assigned to A is the list $(p_1, p_2, \ldots, p_{m+1})$.

[11] A subset of the universe is homogeneous if all of its t-subsets have the same color.

Thus, the total number of colors is $(10^2m+1)^{(m+1)}$. By applying Theorem 19 with $t := m+1, q := (10^2m + 1)^{(m+1)}$, and $N := n$, there is a set $X' \subseteq X$ of size

$$|X'| \geq \frac{\log^{(m)}(n)}{3(10^2m + 1)^{m+1}(m + 1)\log(10^2m + 1)} = \frac{\log^{(m)}(N)}{2^{O(m\log m)}}$$

such that all $m + 1$-subsets of X' have the same color. One can verify that X' is indeed m-homogeneous with respect to A. □

4.2.3 Proof of Lemma 18. The lower bound is proven by using the algorithm A to construct a family of distributions $\mathcal{P}$ with certain properties, and use these properties to derive that $\Omega(k) \leq \mathcal{P} \leq 2^{O(m^2\log^2 m)}$, which implies the desired lower bound.

Lemma 20. *Let A, X', m, k as in Lemma 18, and set $n = k - m$. Then there exists a family $\mathcal{P} = \{P_i : i \leq n\}$ of distributions over $\{\pm 1\}^n$ with the following properties:*

(1) *Every $P_i, P_j \in \mathcal{P}$ are $(0.1, \delta)$-indistinguishable.*
(2) *There exists $r \in [0, 1]$ such that for all $i, j \leq n$:*

$$\Pr_{v\sim P_i}[v(j) = 1] = \begin{cases} \leq r - \frac{1}{10m} & j < i, \\ \geq r + \frac{1}{10m} & j > i. \end{cases}$$

Lemma 21. *Let $\mathcal{P}, n, m, r$ as in Lemma 20. Then, $n \leq 2^{10^3m^2\log^2 m}$.*

By the preceding lemmas, $k - m = |\mathcal{P}| \leq 2^{10^3m^2\log^2 m}$, which implies that $k = 2^{O(m^2\log^2 m)}$ as required. Thus, it remains to prove these lemmas, which we do next.

For the proof of Lemma 20, we will need the following claim.

Claim 22. *Let $(p_i)_{i=0}^m$ denote the probabilities list of X' with respect to A. Then for some $0 < i \leq m$,*

$$p_i - p_{i-1} \geq \frac{1}{4m}.$$

Proof. The proof of this claim uses the assumption that A empirically learns thresholds. Let S be a balanced increasing realizable sample such that $S_X = \{x_1 < \cdots < x_m\} \subseteq X'$ are evenly spaced points on K (so, $S = (x_i, y_i)_{i=1}^m$, where $y_i = -1$ for $i \leq m/2$ and $y_i = +1$ for $i > m_2$).

A is an $(\alpha = 1/16, \beta = 1/16)$-empirical learner, and therefore its expected empirical loss on S is at most $(1 - \beta)\cdot\alpha + \beta\cdot 1 \leq \alpha + \beta = 1/8$, and so

$$\begin{aligned} \frac{7}{8} &\leq \mathop{\mathbb{E}}_{h\sim A(S)}(1 - \mathrm{loss}_S(h)) \\ &= \frac{1}{m}\sum_{i=1}^{m/2}[1 - A_S(x_i)] + \frac{1}{m}\sum_{i=m/2+1}^{m}[A_S(x_i)]. && \text{(since } S \text{ is balanced)} \end{aligned}$$

This implies that there is $m/2 \leq m_1 \leq m$ such that $A_S(x_{m_1}) \geq 3/4$. Next, by privacy, if we consider S' the sample where we replace x_{m_1} by $x_{m_1} + 1$ (with the same label), we have that

$$A_{S'}(x_{m_1}) \geq \left(\frac{3}{4} - \delta\right)e^{-0.1} \geq \frac{2}{3}.$$

Note that $\mathrm{ord}_{S'}(x_{m_1}) = m_1 - 1$, hence by homogeneity: $p_{m_1-1} \geq \frac{2}{3} - \frac{1}{10^2m}$. Similarly, we can show that for some $1 \leq m_2 \leq \frac{m}{2}$, we have $p_{m_2-1} \leq \frac{1}{3} + \frac{1}{10^2m}$. This implies that for some $m_2 - 1 \leq i \leq m_1 - 1$,

$$p_i - p_{i-1} \geq \frac{1/3}{m} - \frac{1}{50m^2} \geq \frac{1}{4m},$$

as required. □

Proof of Lemma 20. Let i be the index guaranteed by Claim 22 such that $p_i - p_{i-1} \geq 1/4m$. Pick an increasing realizable sample $S \in (X' \times \{\pm 1\})^m$ so that the interval $J \subseteq X'$ between x_{i-1} and x_{i+1},

$$J = \Big\{x \in \{1, \ldots, k\} : x_{i-1} < x < x_{i+1}\Big\},$$

is of size $k - m$. For every $x \in J$, let S_x be the neighboring sample of S that is obtained by replacing x with x_i. This yields family of neighboring samples $\{S_x : x \in (x_{i-1}, x_{i+1})\}$ such that

- Every two output distributions $A(S_{x'}), A(S_{x''})$ are (ε, δ)-indistinguishable (because A satisfies (ε, δ) DP).
- Set $r = \frac{p_{i+1}+p_i}{2}$. Then for all $x, x' \in J$,

$$\Pr_{h \sim A(S_x)}\Big[h(x') = 1\Big] = \begin{cases} \leq r - \frac{1}{10m} & x' < x, \\ \geq r + \frac{1}{10m} & x' > x. \end{cases}$$

  The proof is concluded by restricting the output of A to J, and identifying J with $[n]$ and each output distributions $A(S_x)$ with a distribution over $\{\pm 1\}^n$. □

Proof of Lemma 21. Set $T = 10^3 m^2 \log^2 m - 1$, and $D = 10^2 m^2 \log T$. We want to show that $n \leq 2^{T+1}$. Assume toward contradiction that $n > 2^{T+1}$. Consider the family of distributions $Q_i = P_i^D$ for $i = 1, \ldots, n$. By Lemma 13, each Q_i, Q_j is $(0.1D, \delta D)$-indistinguishable.

We next define a set of mutually disjoint events E_i for $i \leq 2^T$ that are measurable with respect to each of the Q_i's. For a sequence of vectors $\mathbf{v} = (v_1, \ldots, v_D)$ in $\{\pm 1\}^n$, we let $\bar{\mathbf{v}} \in \{\pm 1\}^n$ be the threshold vector defined by

$$\bar{\mathbf{v}}(j) = \begin{cases} -1 & \frac{1}{D}\sum_{i=1}^{D} v_i(j) \leq r, \\ +1 & \frac{1}{D}\sum_{i=1}^{D} v_i(j) \geq r. \end{cases}$$

Given a point in the support of any of the Q_i's, namely a sequence $\mathbf{v} = (v_1, \ldots, v_D)$ of D vectors in $\{\pm 1\}^n$, define a mapping B according to the outcome of T steps of binary search on $\bar{\mathbf{v}}$ as follows: probe the $\frac{n}{2}$'th entry of $\bar{\mathbf{v}}$; if it is $+1$, then continue recursively with the first half of $\bar{\mathbf{v}}$. Else, continue recursively with the second half of $\bar{\mathbf{v}}$. Define the mapping $B = B(\mathbf{v})$ to be the entry that was probed at the T'th step. The events E_j correspond to the 2^T different outcomes of B. These events are mutually disjoint by the assumption that $n > 2^{T+1}$.

Notice that for any possible i in the image of B, applying the binary search on a sufficiently large independent and identically distributed sample $\mathbf{v}$ from P_i would yield $B(\mathbf{v}) = i$ with high probability. Quantitatively, a standard application of Chernoff inequality and a union bound imply that the event $E_i = \{\mathbf{v} : B(\bar{\mathbf{v}}) = i\}$ for $\mathbf{v} \sim Q_i$, has probability at least

$$1 - T\exp\Big(-2\frac{1}{10^2 m^2}D\Big) = 1 - T\exp(-2\log T) \geq \frac{2}{3}.$$

We claim that for all $j \leq n$, and i in the image of B,

$$Q_j(E_i) \geq \frac{1}{2}\exp(-0.1D). \tag{1}$$

This will finish the proof since the 2^T events are mutually disjoint, and therefore

$$\begin{aligned} 1 &\geq Q_j(\cup_i E_i) \\ &= \sum_i Q_j(E_i) \\ &\geq 2^T \cdot \frac{1}{2}e^{-0.1D} \\ &= 2^{T-1}e^{-0.1D}. \end{aligned}$$

However, $2^{T-1}e^{-0.1D} > 1$ by the choice of T, D, which is a contradiction.

Thus, it remains to prove Equation (1). This follows since Q_i, Q_j are $(0.1D, D\delta)$-indistinguishable:

$$\frac{2}{3} \leq Q_i(E_i) \leq \exp(0.1D)Q_j(E_i) + D\delta,$$

and by the choice of δ, which implies that $\frac{2}{3} - D\delta \geq \frac{1}{2}$. □

4.3 Privately Learnable Classes Have Finite Littlestone Dimension

We conclude this section by deriving Theorem 2, which gives a lower bound of $\Omega(\log^* d)$ on the sample complexity of privately learning a class with Littlestone dimension d.

Proof of Theorem 2. The proof is a direct corollary of Theorems 10 and 1. Indeed, let H be a class with Littlestone dimension d, and let $c = \lfloor \log d \rfloor$. By Item 1 of Theorem 10, there are $x_1, \ldots, x_c$ and $h_1, \ldots, h_c \in H$ such that $h_i(x_j) = +1$ if and only if $j \geq i$. Theorem 1 implies a lower bound of $m \geq \Omega(\log^* c) = \Omega(\log^* d)$ for any algorithm that learns $\{h_i : i \leq c\}$ with accuracy $(1/16, 1/16)$ and privacy $(0.1, O(1/m^2 \log m))$. □

5 FINITE LITTLESTONE DIMENSION IMPLIES PRIVATE LEARNING

In this section, we prove that every Littlestone class $\mathcal{H}$ is PAC learnable by a DP algorithm (Theorem 3). We begin by providing a proof overview in Section 5.1. Then, in Section 5.2, we prove that every Littlestone class can be learned by a globally stable algorithm, and in Section 5.3 that globally stable algorithms can be transformed to DP algorithms. Finally, in Section 5.4, we wrap up by proving Theorem 3.

5.1 Proof Overview

We next give an overview of the main arguments used in the proof of Theorem 3. The proof consist of two parts: (i) we first show that every class with a finite Littlestone dimension can be learned by a globally stable algorithm, and (ii) we then show how to generically obtain a differentially private learner from any globally stable learner.

5.1.1 Step 1: Finite Littlestone Dimension $\implies$ Globally Stable Learning. Let $\mathcal{H}$ be a concept class with Littlestone dimension d. Our goal is to design a globally stable learning algorithm for $\mathcal{H}$ with stability parameter $\eta = 2^{-2^{O(d)}}$ and sample complexity $n = 2^{2^{O(d)}}$. We will sketch here a weaker variant of our construction that uses the same ideas but is simpler to describe.

The property of $\mathcal{H}$ that we will use is that it can be online learned in the realizable setting with at most d mistakes (see Section 3.2 for a brief overview of this setting). Let $\mathcal{D}$ denote a realizable distribution with respect to which we wish to learn in a globally stable manner. In other words, $\mathcal{D}$ is a distribution over examples $(x, c(x))$, where $c \in \mathcal{H}$ is an unknown target concept. Let $\mathcal{A}$ be a learning algorithm that makes at most d mistakes while learning an unknown concept from $\mathcal{H}$ in the online model. Consider applying $\mathcal{A}$ on a sequence $S = ((x_1, c(x_1)) \ldots (x_n, c(x_n))) \sim \mathcal{D}^n$, and denote by M the random variable counting the number of mistakes $\mathcal{A}$ makes in this process. The mistake bound guarantee on $\mathcal{A}$ guarantees that $M \leq d$ always. Consequently, there is $0 \leq i \leq d$ such that

$$\Pr[M = i] \geq \frac{1}{d+1}.$$

Note that we can identify, with high probability, an i such that $\Pr[M = i] \geq 1/2d$ by running $\mathcal{A}$ on $O(d)$ samples from $\mathcal{D}^n$. We next describe how to handle each of the $d + 1$ possibilities for i. Let us first assume that $i = d$, namely that

$$\Pr[M = d] \geq \frac{1}{2d}.$$

We claim that in this case we are done: indeed, after making d mistakes, it must be the case that $\mathcal{A}$ has completely identified the target concept c (or else $\mathcal{A}$ could be presented with another example that forces it to make $d + 1$ mistakes). Thus, in this case, it holds with probability at least $1/2d$ that $\mathcal{A}(S) = c$ and we are done. Let us next assume that $i = d - 1$, namely that

$$\Pr[M = d - 1] \geq \frac{1}{2d}.$$

The issue with applying the previous argument here is that before making the d'th mistake, $\mathcal{A}$ can output many different hypotheses (depending on the input sample S). We use the following idea: draw two samples $S_1, S_2 \sim \mathcal{D}^n$ independently, and set $f_1 = \mathcal{A}(S_1)$ and $f_2 = \mathcal{A}(S_2)$. Condition on the event that the number of mistakes made by $\mathcal{A}$ on each of S_1, S_2 is exactly $d - 1$ (by assumption, this event occurs with probability at least $(1/2d)^2$) and consider the following two possibilities:

(i) $\Pr[f_1 = f_2] \geq \frac{1}{4}$,
(ii) $\Pr[f_1 = f_2] < \frac{1}{4}$.

If (i) holds, then using a simple calculation one can show that there is h such that $\Pr[A(S) = h] \geq \frac{1}{(2d)^2} \cdot \frac{1}{4}$ and we are done. If (ii) holds, then we apply the following *random contest* between S_1, S_2:

(1) Pick x such that $f_1(x) \neq f_2(x)$ and draw $y \sim \{\pm 1\}$ uniformly at random.
(2) If $f_1(x) \neq y$, then the output is $\mathcal{A}(S_1 \circ (x, y))$, where $S_1 \circ (x, y)$ denotes the sample obtained by appending (x, y) to the end of S. In this case, we say that S_1 "won the contest."
(3) Else, $f_2(x) \neq y$, then the output is $\mathcal{A}(S_2 \circ (x, y))$. In this case, we that S_2 "won the contest."

Note that adding the auxiliary example (x, y) forces $\mathcal{A}$ to make exactly d mistakes on $S_i \circ (x, y)$. Now, if $y \sim \{\pm 1\}$ satisfies $y = c(x)$, then by the mistake bound argument it holds that $\mathcal{A}(S_i \circ (x, y)) = c$. Therefore, since $\Pr_{y \sim \{\pm 1\}}[c(x) = y] = 1/2$, it follows that

$$\Pr_{S_1, S_2, y}[\mathcal{A}(S_i \circ (x, y)) = c] \geq \frac{1}{(2d)^2} \cdot \frac{3}{4} \cdot \frac{1}{2} = \Omega(1/d^2),$$

and we are done.

Similar reasoning can be used by induction to handle the remaining cases (the next one would be that $\Pr[M = d - 2] \geq \frac{1}{2d}$, and so on). As the number of mistakes reduces, we need to guess more labels, to enforce mistakes on the algorithm. As we guess more labels, the success rate reduces; nevertheless, we never need to make more than 2^d such guesses. (Note that the random contests performed by the algorithm can naturally be presented using the internal nodes of a binary tree of depth $\leq d$.)he proof we present in Section 5.2 is based on a similar idea of performing random contests, although the construction becomes more complex to handle other issues, such as generalization, which were not addressed here. For more details, we refer the reader to the complete argument in Section 5.2.

5.1.2 Step 2: Globally Stable Learning $\Longrightarrow$ Differentially Private Learning. Given a globally stable learner $\mathcal{A}$ for a concept class $\mathcal{H}$, we can obtain a differentially private learner using standard techniques in the literature on private learning and query release. If $\mathcal{A}$ is a (η, m)-globally stable learner with respect to a distribution $\mathcal{D}$, we obtain a differentially private learner using roughly m/η samples from that distribution as follows. We first run $\mathcal{A}$ on $k \approx 1/\eta$ independent samples, non-privately producing a list of k hypotheses. We then apply a differentially private "Stable Histograms" algorithm [21, 58] to this list that allows us to privately publish a short list of hypotheses that appear with frequency $\Omega(\eta)$. Global stability of the learner $\mathcal{A}$ guarantees that with high probability, this list contains *some* hypothesis h with small population loss. We can then apply a generic differentially private learner (based on the exponential mechanism) on a fresh set of examples to identify such an accurate hypothesis from the short list.

5.2 Globally Stable Learning of Littlestone Classes

5.2.1 Theorem Statement. The following theorem states that any class $\mathcal{H}$ with a bounded Littlestone dimension can be learned by a globally stable algorithm.

Theorem 23. *Let $\mathcal{H}$ be a hypothesis class with Littlestone dimension $d \geq 1$, let $\alpha > 0$, and set*

$$m = 2^{2^{d+2}+1} 4^{d+1} \cdot \left\lceil \frac{2^{d+2}}{\alpha} \right\rceil.$$

Then there exists a randomized algorithm $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ with the following properties. Let $\mathcal{D}$ be a realizable distribution, and let $S \sim \mathcal{D}^m$ be an input sample. Then there exists a hypothesis f such that

$$\Pr[G(S) = f] \geq \frac{1}{(d+1)2^{2^d+1}} \text{ and } \mathrm{loss}_{\mathcal{D}}(f) \leq \alpha.$$

5.2.2 The Distributions $\mathcal{D}_k$. Algorithm G is obtained by running the SOA on a sample drawn from a carefully tailored distribution. This distribution belongs to a family of distributions that we define next. Each of these distributions can be sampled from using black-box access to independent and identically distributed samples from $\mathcal{D}$. Recall that for a pair of samples S, T, we denote by $S \circ T$ the sample obtained by appending T to the end of S. Define a sequence of distributions $\mathcal{D}_k$ for $k \geq 0$ as shown in the boxed text.

Distributions $\mathcal{D}_k$

Let n denote an "auxiliary sample" size (to be fixed later), and let $\mathcal{D}$ denote the target realizable distribution over examples. The distributions $\mathcal{D}_k = \mathcal{D}_k(\mathcal{D}, n)$ are defined by induction on k as follows:

(1) $\mathcal{D}_0$: output the empty sample $\emptyset$ with probability 1.
(2) Let $k \geq 1$. If there exists an f such that

$$\Pr_{S \sim \mathcal{D}_{k-1}, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = f] \geq 2^{-2^{d+2}},$$

or if $\mathcal{D}_{k-1}$ is undefined, then $\mathcal{D}_k$ is undefined.
(3) Else, $\mathcal{D}_k$ is defined recursively by the following process:
  (i) Draw $S_0, S_1 \sim \mathcal{D}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$ independently.
  (ii) Let $f_0 = \mathrm{SOA}(S_0 \circ T_0)$, $f_1 = \mathrm{SOA}(S_1 \circ T_1)$.
  (iii) If $f_0 = f_1$, then go back to step (i).
  (iv) Else, pick $x \in \{x : f_0(x) \neq f_1(x)\}$ and sample $y \sim \{\pm 1\}$ uniformly.
  (v) If $f_0(x) \neq y$, then output $S_0 \circ T_0 \circ ((x, y))$ and else output $S_1 \circ T_1 \circ ((x, y))$.

Please see Figure 4 for an illustration of sampling $S \sim \mathcal{D}_k$ for $k = 3$.

We next observe some basic facts regarding these distributions. First, note that whenever $\mathcal{D}_k$ is well defined, the process in Item 3 terminates with probability 1.

Let k be such that $\mathcal{D}_k$ is well defined and consider a sample S drawn from $\mathcal{D}_k$. The size of S is $|S| = k \cdot (n+1)$. Among these $k \cdot (n+1)$ examples, there are $k \cdot n$ examples drawn from $\mathcal{D}$ and k examples that are generated in Item 3(iv). We will refer to these k examples as *tournament*

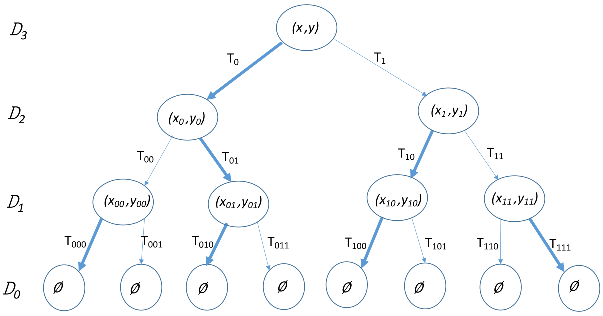

Fig. 4. An illustration of the process of generating a sample $S \sim \mathcal{D}_3$. The edge labels are the samples T_b drawn in Item 3(i). The node labels are the tournament examples (x_b, y_b) generated in Item 3(iv). The bold edges indicate which of the samples T_{b0}, T_{b1} was appended to S in step 3(v) along with the corresponding tournament example. The sample S generated in this illustration is $T_{010} \circ (x_{01}, y_{01}) \circ T_{01} \circ (x_0, y_0) \circ T_0 \circ (x, y)$.

examples. *Note that during the generation of $S \sim \mathcal{D}_k$, there are examples drawn from $\mathcal{D}$ that do not actually appear in S. In fact, the number of such examples may be unbounded, depending on how many times Items 3(i) through 3(iii) were repeated. In Section 5.2.3, we will define a "Monte Carlo" variant of $\mathcal{D}_k$ in which the number of examples drawn from $\mathcal{D}$ is always bounded. This Monte Carlo variant is what we actually use to define our globally stable learning algorithm, but we introduce the simpler distributions $\mathcal{D}_k$ to clarify our analysis.*

The k tournament examples satisfy the following important properties.

Observation 24. *Let k be such that $\mathcal{D}_k$ is well defined and consider running the* SOA *on the concatenated sample $S \circ T$, where $S \sim \mathcal{D}_k$ and $T \sim \mathcal{D}^n$. Then*

(1) *Each tournament example forces a mistake on the* SOA. *Consequently, the number of mistakes made by the* SOA *when run on $S \circ T$ is at least k.*
(2) SOA$(S \circ T)$ *is consistent with T.*

The first item follows directly from the definition of x in Item 3(iv) and the definition of S in Item 3(v). The second item clearly holds when $S \circ T$ is realizable by $\mathcal{H}$ (because the SOA *is consistent). For non-realizable $S \circ T$, Item 2 holds by our extension of the* SOA *in Definition 11.*

The existence of frequent hypotheses. The following lemma is the main step in establishing global stability.

Lemma 25. *There exists $k \leq d$ and an hypothesis $f : X \to \{\pm 1\}$ such that*

$$\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = f] \geq 2^{-2^{d+2}}.$$

Proof. Suppose for the sake of contradiction that this is not the case. In particular, this means that $\mathcal{D}_d$ is well defined and that for every f,

$$\Pr_{S \sim \mathcal{D}_d, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = f] < 2^{-2^{d+2}}. \tag{2}$$

We show that this cannot be the case when $f = c$ is the target concept (i.e., for $c \in \mathcal{H}$, which satisfies $\mathrm{loss}_{\mathcal{D}}(c) = 0$). Toward this end, we first show that with probability $2^{-2^{d+2}}$ over $S \sim \mathcal{D}_d$, we have that all d tournament examples are consistent with c: for $k \leq d$, let ρ_k denote the probability that all k tournament examples in $S \sim \mathcal{D}_k$ are consistent with c. We claim that ρ_k

satisfies the recursion $\rho_k \geq \frac{1}{2}(\rho_{k-1}^2 - 8 \cdot 2^{-2^{d+2}})$. Indeed, consider the event E_k that (i) in each of $S_0, S_1 \sim \mathcal{D}_{k-1}$, all $k-1$ tournament examples are consistent with c, and (ii) that $f_0 \neq f_1$. Since $f_0 = f_1$ occurs with probability at most $2^{-2^{d+2}} < 8 \cdot 2^{-2^{d+2}}$, it follows that $\Pr[E_k] \geq \rho_{k-1}^2 - 8 \cdot 2^{-2^{d+2}}$. Further, since $y \in \{\pm 1\}$ is chosen uniformly at random and independently of S_0 and S_1, we have that conditioned on E_k, $c(x) = y$ with probability $1/2$. Taken together, we have that $\rho_k \geq \frac{1}{2}\Pr[E_k] \geq \frac{1}{2}(\rho_{k-1}^2 - 8 \cdot 2^{-2^{d+2}})$. Since $\rho_0 = 1$, we get the recursive relation

$$\rho_k \geq \frac{\rho_{k-1}^2 - 8 \cdot 2^{-2^{d+2}}}{2}, \text{ and } \rho_0 = 1.$$

Thus, it follows by induction that for $k \leq d$, $\rho_k \geq 4 \cdot 2^{-2^{k+1}}$: the base case is verified readily, and the induction step is as follows:

$$\begin{aligned} \rho_k &\geq \frac{\rho_{k-1}^2 - 8 \cdot 2^{-2^{d+2}}}{2} \\ &\geq \frac{(4 \cdot 2^{-2^k})^2 - 8 \cdot 2^{-2^{d+2}}}{2} && \text{(by induction)} \\ &= 8 \cdot 2^{-2^{k+1}} - 4 \cdot 2^{-2^{d+2}} \\ &\geq 4 \cdot 2^{-2^{k+1}}. && (k \leq d \text{ and therefore } 2^{-2^{d+2}} \leq 2^{-2^{k+1}}) \end{aligned}$$

Therefore, with probability $2^{-2^{d+2}}$, we have that $S \circ T$ is consistent with c (because all examples in $S \circ T$ that are drawn from $\mathcal{D}$ are also consistent with c). Now, since each tournament example forces a mistake on the SOA (Observation 24), and since the SOA does not make more than d mistakes on realizable samples, it follows that if all tournament examples in $S \sim \mathcal{D}_d$ are consistent with c, then $\mathrm{SOA}(S) = \mathrm{SOA}(S \circ T) = c$. Thus,

$$\Pr_{S \sim \mathcal{D}_d, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = c] \geq 2^{-2^{d+2}},$$

which contradicts Equation (2) and finishes the proof. □

Generalization. The next lemma shows that only hypotheses f that generalize well satisfy the conclusion of Lemma 25 (note the similarity of this proof with the proof of Proposition 7).

Lemma 26 (Generalization). *Let k be such that $\mathcal{D}_k$ is well defined. Then every f such that*

$$\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = f] \geq 2^{-2^{d+2}}$$

satisfies $\mathrm{loss}_{\mathcal{D}}(f) \leq \frac{2^{d+2}}{n}$.

Proof. Let f be a hypothesis such that $\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathrm{SOA}(S \circ T) = f] \geq 2^{-2^{d+2}}$ and let $\alpha = \mathrm{loss}_{\mathcal{D}}(h)$. We will argue that

$$2^{-2^{d+2}} \leq (1-\alpha)^n. \tag{3}$$

Define the events A, B as follows:

(1) A is the event that $\mathrm{SOA}(S \circ T) = f$. By assumption, $\Pr[A] \geq 2^{-2^{d+2}}$.
(2) B is the event that f is consistent with T. Since $|T| = n$, we have that $\Pr[B] = (1-\alpha)^n$.

Note that $A \subseteq B$: indeed, $\mathrm{SOA}(S \circ T)$ is consistent with T by the second item of Observation 24. Thus, whenever $\mathrm{SOA}(S \circ T) = f$, it must be the case that f is consistent with T. Hence,

$\Pr[A] \leq \Pr[B]$, which implies Inequality (3) and finishes the proof (using the fact that $1 - \alpha \leq 2^{-\alpha}$ and taking logarithms on both sides). □

5.2.3 The Algorithm G.

A Monte Carlo variant of $\mathcal{D}_k$. Consider the following first attempt of defining a globally stable learner G: (i) draw $i \in \{0 \dots d\}$ uniformly at random, (ii) sample $S \sim \mathcal{D}_i$, and (iii) output $\mathrm{SOA}(S \circ T)$, where $T \sim \mathcal{D}^n$. The idea is that with probability $1/(d+1)$, the sampled i will be equal to a number k satisfying the conditions of Lemma 25, and so the desired hypothesis f guaranteed by this lemma (which also has low population loss by Lemma 26) will be outputted with probability at least $2^{-2^d}/(d+1)$.

The issue here is that sampling $f \sim \mathcal{D}_i$ may require an unbounded number of samples from the target distribution $\mathcal{D}$ (in fact, $\mathcal{D}_i$ may even be undefined). To circumvent this possibility, we define a Monte Carlo variant of $\mathcal{D}_k$ in which the number of examples drawn from $\mathcal{D}$ is always bounded.

The Distributions $\tilde{\mathcal{D}}_k$ (a Monte Carlo Variant of $\mathcal{D}_k$)

1. Let n be the auxiliary sample size and N be an upper bound on the number of examples drawn from $\mathcal{D}$.
2. $\tilde{\mathcal{D}}_0$: output the empty sample $\emptyset$ with probability 1.
3. For $k > 0$, define $\tilde{\mathcal{D}}_k$ recursively by the following process:
   - (*) Throughout the process, if more than N examples from $\mathcal{D}$ are drawn (including examples drawn in the recursive calls), then output "Fail."
   - (i) Draw $S_0, S_1 \sim \tilde{\mathcal{D}}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$ independently.
   - (ii) Let $f_0 = \mathrm{SOA}(S_0 \circ T_0)$, $f_1 = \mathrm{SOA}(S_1 \circ T_1)$.
   - (iii) If $f_0 = f_1$, then go back to step (i).
   - (iv) Else, pick $x \in \{x : f_0(x) \neq f_1(x)\}$ and sample $y \sim \{\pm 1\}$ uniformly.
   - (v) If $f_0(x) \neq y$, then output $S_0 \circ T_0 \circ ((x, y))$ and else output $S_1 \circ T_1 \circ ((x, y))$.

Note that $\tilde{\mathcal{D}}_k$ is well defined for every k, even for k such that $\mathcal{D}_k$ is undefined (however, for such k's, the probability of outputting "Fail" may be large).

It remains to specify the upper bound N on the number of examples drawn from $\mathcal{D}$ in $\tilde{\mathcal{D}}_k$. Toward this end, we prove the following bound on the expected number of examples from $\mathcal{D}$ that are drawn during generating $S \sim \mathcal{D}_k$.

Lemma 27 (Expected Sample Complexity of Sampling from $\mathcal{D}_k$). *Let k be such that $\mathcal{D}_k$ is well defined, and let M_k denote the number of examples from $\mathcal{D}$ that are drawn in the process of generating $S \sim \mathcal{D}_k$. Then,*

$$\mathbb{E}[M_k] \leq 4^{k+1} \cdot n.$$

Proof. Note that $\mathbb{E}[M_0] = 0$ as $\mathcal{D}_0$ deterministically produces the empty sample. We first show that for all $0 < i < k$,

$$\mathbb{E}[M_{i+1}] \leq 4\mathbb{E}[M_i] + 4n, \tag{4}$$

and then conclude the desired inequality by induction.

To see why Inequality (4) holds, let the random variable R denote the number of times Item 3(i) was executed during the generation of $S \sim \mathcal{D}_{i+1}$. In other words, R is the number of times a pair $S_0, S_1 \sim \mathcal{D}_i$ and a pair $T_0, T_1 \sim \mathcal{D}^n$ were drawn. Observe that R is distributed geometrically with success probability θ, where

$$\begin{aligned}\theta &= 1 - \Pr_{S_0, S_1, T_0, T_1}\Big[\mathrm{SOA}(S_0 \circ T_0) = \mathrm{SOA}(S_1 \circ T_1)\Big] \\ &= 1 - \sum_{h} \Pr_{S,T}\Big[\mathrm{SOA}(S \circ T) = h\Big]^2 \\ &\geq 1 - 2^{-2^{d+2}},\end{aligned}$$

where the last inequality follows because $i < k$ and hence $\mathcal{D}_i$ is well defined, which implies that $\Pr_{S,T}[\mathrm{SOA}(S \circ T) = h] \leq 2^{-2^{d+2}}$ for all h.

Now, the random variable M_{i+1} can be expressed as follows:

$$M_{i+1} = \sum_{j=1}^{\infty} M_{i+1}^{(j)},$$

where

$$M_{i+1}^{(j)} = \begin{cases} 0 & \text{if } R < j, \\ \text{\# of examples drawn from } \mathcal{D} \text{ in the } j\text{'th execution of Item 3(i)} & \text{if } R \geq j. \end{cases}$$

Thus, $\mathbb{E}[M_{i+1}] = \sum_{j=1}^{\infty} \mathbb{E}[M_{i+1}^{(j)}]$. We claim that

$$\mathbb{E}[M_{i+1}^{(j)}] = (1-\theta)^{j-1} \cdot (2\mathbb{E}[M_i] + 2n).$$

Indeed, the probability that $R \geq j$ is $(1-\theta)^{j-1}$ and conditioned on $R \geq j$, in the j'th execution of Item 3(i) two samples from $\mathcal{D}_i$ are drawn and two samples from $\mathcal{D}^n$ are drawn. Thus,

$$\mathbb{E}[M_{i+1}] = \sum_{j=1}^{\infty} (1-\theta)^{j-1} \cdot (2\mathbb{E}[M_i] + 2n) = \frac{1}{\theta} \cdot (2\mathbb{E}[M_i] + 2n) \leq 4\mathbb{E}[M_i] + 4n,$$

where the last inequality is true because $\theta \geq 1 - 2^{-2^{d+2}} \geq 1/2$.

This gives Inequality (4). Next, using that $\mathbb{E}[M_0] = 0$, a simple induction gives

$$\mathbb{E}[M_{i+1}] \leq (4 + 4^2 + \cdots + 4^{i+1})n \leq 4^{i+2} n,$$

and the lemma follows by taking $i + 1 = k$. □

Proof of Theorem 23. Our globally stable learning algorithm G is defined as shown in the boxed text.

Algorithm G

(1) Consider the distribution $\tilde{\mathcal{D}}_k$, where the auxiliary sample size is set to $n = \lceil \frac{2^{d+2}}{\alpha} \rceil$ and the sample complexity upper bound is set to $N = 2^{2^{d+2}+1} 4^{d+1} \cdot n$.
(2) Draw $k \in \{0, 1, \ldots, d\}$ uniformly at random.
(3) Output $h = \mathrm{SOA}(S \circ T)$, where $T \sim \mathcal{D}^n$ and $S \sim \tilde{\mathcal{D}}_k$.

First note that the sample complexity of G is $|S| + |T| \leq N + n = (2^{2^{d+2}+1} 4^{d+1} + 1) \cdot \lceil \frac{2^{d+2}}{\alpha} \rceil$, as required. It remains to show that there exists a hypothesis f such that

$$\Pr[G(S) = f] \geq \frac{2^{-2^{d+2}}}{d+1} \text{ and } \mathrm{loss}_{\mathcal{D}}(f) \leq \alpha.$$

By Lemma 25, there exists $k^* \le d$ and f^* such that

$$\Pr_{S\sim\mathcal{D}_{k^*},T\sim\mathcal{D}^n}[\mathrm{SOA}(S\circ T) = f^*] \ge 2^{-2^{d+2}}.$$

We assume k^* is minimal—in particular, $\mathcal{D}_k$ is well defined for $k \le k^*$. By Lemma 26,

$$\mathrm{loss}_{\mathcal{D}}(f^*) \le \frac{2^{d+2}}{n} \le \alpha.$$

We claim that G outputs f^* with probability at least $2^{-2^{d+2}-1}$. To see this, let M_{k^*} denote the number of examples drawn from $\mathcal{D}$ during the generation of $S \sim \mathcal{D}_{k^*}$. Lemma 27 and an application of Markov's inequality yield

$$\begin{aligned}\Pr\Big[M_{k^*} > 2^{2^{d+2}+1}\cdot 4^{d+1}\cdot n\Big] &\le \Pr\Big[M_{k^*} > 2^{2^{d+2}+1}\cdot 4^{k^*+1}\cdot n\Big] && \text{(because } k^* \le d\text{)}\\ &\le 2^{-2^{d+2}-1}. && \text{(by Markov's inequality, since } \mathbb{E}[M_{k^*}] \le 4^{k^*+1}\cdot n\text{)}\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr_{S\sim\tilde{\mathcal{D}}_{k^*},T\sim\mathcal{D}^n}[\mathrm{SOA}(S\circ T) = f^*] &= \Pr_{S\sim\mathcal{D}_{k^*},T\sim\mathcal{D}^n}\Big[\mathrm{SOA}(S\circ T) = f^* \text{ and } M_{k^*} \le 2^{2^d+2}4^{d+1}\cdot n\Big]\\ &\ge 2^{-2^{d+2}} - 2^{-2^{d+2}-1} = 2^{-2^d-1}.\end{aligned}$$

Thus, since $k = k^*$ with probability $1/(d+1)$, it follows that G outputs f^* with probability at least $\frac{2^{-2^{d+2}-1}}{d+1}$ as required. □

5.3 Globally Stable Learning Implies Private Learning

In this section, we prove that any globally stable learning algorithm yields a differentially private learning algorithm with finite sample complexity.

5.3.1 Tools from DP. We begin by stating a few standard tools from the DP literature that underlie our construction of a learning algorithm.

Let X be a data domain, and let $S \in X^n$. For an element $x \in X$, define $\mathrm{freq}_S(x) = \frac{1}{n}\cdot\#\{i \in [n] : x_i = x\}$—that is, the fraction of the elements in S which are equal to x.

Lemma 28 (Stable Histograms [21, 58]). *Let X be any data domain. For*

$$n \ge O\left(\frac{\log(1/\eta\beta\delta)}{\eta\varepsilon}\right),$$

there exists an (ε,δ)-differentially private algorithm Hist *that, with probability at least $1-\beta$, on input $S = (x_1,\ldots,x_n)$ outputs a list $L \subseteq X$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that*

- *Every x with $\mathrm{freq}_S(x) \ge \eta$ appears in L, and*
- *For every $x \in L$, the estimate a_x satisfies $|a_x - \mathrm{freq}_S(x)| \le \eta$.*

Using the Exponential Mechanism of McSherry and Talwar [64], Kasiviswanathan et al. [57] described a generic differentially private learner based on approximate empirical risk minimization.

Lemma 29 (Generic Private Learner [57]). *Let $H \subseteq \{\pm 1\}^X$ be a collection of hypotheses. For*

$$n = O\left(\frac{\log|H| + \log(1/\beta)}{\alpha\varepsilon}\right),$$

there exists an ε-differentially private algorithm GenericLearner $: (X\times\{\pm1\})^n \to H$ *such that the following holds. Let $\mathcal{D}$ be a distribution over $(X\times\{\pm1\})$ such that there exists $h^* \in H$ with*

$$\mathrm{loss}_{\mathcal{D}}(h^*) \le \alpha.$$

Then on input $S \sim \mathcal{D}^n$, algorithm GenericLearner *outputs, with probability at least* $1-\beta$*, a hypothesis* $\hat{h} \in H$ *such that*

$$\mathrm{loss}_{\mathcal{D}}(\hat{h}) \le 2\alpha.$$

Our formulation of the guarantees of this algorithm differ slightly from those of Kasiviswanathan et al. [57], so we give its standard proof for completeness.

Proof of Lemma 29. The algorithm GenericLearner(S) samples a hypothesis $h \in H$ with probability proportional to $\exp(-\varepsilon n \, \mathrm{loss}_S(h)/2)$. This algorithm can be seen as an instantiation of the Exponential Mechanism [64]; the fact that changing one sample changes the value of $\mathrm{loss}_S(h)$ by at most 1 implies that GenericLearner is ε-differentially private.

We now argue that GenericLearner is an accurate learner. Let E denote the event that the sample S satisfies the following conditions:

(1) For every $h \in H$ such that $\mathrm{loss}_{\mathcal{D}}(h) > 2\alpha$, it also holds that $\mathrm{loss}_S(h) > 5\alpha/3$, and
(2) For the hypothesis $h^* \in H$ satisfying $\mathrm{loss}_{\mathcal{D}}(h^*) \le \alpha$, it also holds that $\mathrm{loss}_S(h^*) \le 4\alpha/3$.

We claim that $\Pr[E] \ge 1-\beta/2$ as long as $n \ge O(\log(|H|/\beta)/\alpha)$. To see this, let $h \in H$ be an arbitrary hypothesis with $\mathrm{loss}_D(h) > 2\alpha$. By a multiplicative Chernoff bound,[12] we have $\mathrm{loss}_S(h) > 7\alpha/4$ with probability at least $1-\beta/(4|H|)$ as long as $n \ge O(\log(|H|/\beta)/\alpha)$. Taking a union bound over all $h \in H$ shows that condition (1) holds with probability at least $1-\beta/4$. Similarly, a multiplicative Chernoff bound ensures that condition (2) holds with probability at least $1-\beta/4$, so E holds with probability at least $1-\beta/2$.

Now we show that conditioned on E, the algorithm GenericLearner(S) indeed produces a hypothesis h with $\mathrm{loss}_D(\hat{h}) \le 2\alpha$. This follows the standard analysis of the accuracy guarantees of the Exponential Mechanism. Condition 2 of the definition of event E guarantees that $\mathrm{loss}_S(h^*) \le 4\alpha/3$. This ensures that the normalization factor in the definition of the Exponential Mechanism is at least $\exp(-2\varepsilon\alpha n/3)$. Hence, by a union bound,

$$\Pr[\mathrm{loss}_S(\hat{h}) > 5\alpha/3] \le |H| \cdot \frac{\exp(-5\varepsilon\alpha n/6)}{\exp(-2\varepsilon\alpha n/3)} = |H| e^{-\varepsilon\alpha n/6}.$$

Taking $n \ge O(\log(|H|/\beta)/\alpha\varepsilon)$ ensures that this probability is at most $\beta/2$. Given that $\mathrm{loss}(\hat{h}) \le 5\alpha/3$, condition (1) of the definition of event E ensures that $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \le 2\alpha$. Thus, for n sufficiently large as described, we have overall that $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \le 2\alpha$ with probability at least $1-\beta$. □

5.3.2 Construction of a Private Learner.

We now describe how to combine the Stable Histograms algorithm with the Generic Private Learner to convert any globally stable learning algorithm into a differentially private one.

Theorem 30. *Let $\mathcal{H}$ be a concept class over data domain X. Let $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ be a randomized algorithm such that for $\mathcal{D}$ a realizable distribution and $S \sim \mathcal{D}^m$, there exists a hypothesis h such that* $\Pr[G(S) = h] \ge \eta$ *and* $\mathrm{loss}_{\mathcal{D}}(h) \le \alpha/2$.

Then for some

$$n = \tilde{O}\left(\frac{m \cdot \log(1/\eta\beta\delta)}{\eta\varepsilon} + \frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right),$$

there exists an (ε, δ)-differentially private algorithm $M : (X \times \{\pm 1\})^n \to \{\pm 1\}^X$ that, given n independent and identically distributed samples from $\mathcal{D}$, produces a hypothesis $\hat{h}$ such that $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \le \alpha$ *with probability at least* $1-\beta$.

[12] In other words, for independent random variables $Z_1, \ldots, Z_n$ whose sum Z satisfies $\mathbb{E}[Z] = \mu$, we have for every $\delta \in (0,1)$ that $\Pr[Z \le (1-\delta)\mu] \le \exp(-\delta^2\mu/2)$ and $\Pr[Z \ge (1+\delta)\mu] \le \exp(-\delta^2\mu/3)$.

Theorem 30 is realized via the learning algorithm M described in the following. Here, the parameter

$$k = \tilde{O}\left(\frac{\log(1/\eta\beta\delta)}{\eta\varepsilon}\right)$$

is chosen so that Lemma 28 guarantees the algorithm Hist succeeds with the stated accuracy parameters. The parameter

$$n' = \tilde{O}\left(\frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right)$$

is chosen so that Lemma 29 guarantees that GenericLearner succeeds on a list L of size $|L| \leq 2/\eta$ with the given accuracy and confidence parameters.

Differentially Private Learner M

(1) Let $S_1, \ldots, S_k$ each consist of m independent and identically distributed samples from $\mathcal{D}$. Run G on each batch of samples producing $h_1 = G(S_1), \ldots, h_k = G(S_k)$.
(2) Run the Stable Histogram algorithm Hist on input $H = (h_1, \ldots, h_k)$ using privacy parameters $(\varepsilon/2, \delta)$ and accuracy parameters $(\eta/8, \beta/3)$, producing a list L of frequent hypotheses.
(3) Remove from L all hypotheses with estimated frequency $a_h < 3\eta/4$.
(4) Let S' consist of n' independent and identically distributed samples from $\mathcal{D}$. Run GenericLearner(S') using the collection of hypotheses L with privacy parameter $\varepsilon/2$ and accuracy parameters $(\alpha/2, \beta/3)$ to output a hypothesis $\hat{h}$.

Proof of Theorem 30. We first argue that the algorithm M is differentially private. The outcome L of step 2 is generated in a $(\varepsilon/2, \delta)$-differentially private manner as it inherits its privacy guarantee from Hist. For every fixed choice of the coin tosses of G during the executions $G(S_1), \ldots, G(S_k)$, a change to one entry of some S_i changes at most one outcome $h_i \in H$. DP for step 2 follows by taking expectations over the coin tosses of all executions of G, and for the algorithm as a whole by simple composition.

We now argue that the algorithm is accurate. Using the fact that $k \geq \tilde{O}(\log(1/\beta)/\eta)$, standard generalization arguments (e.g., see Theorem A3.1 in the work of Blumer et al. [14]) imply that with probability at least $1 - \beta/3$, every h such that $\Pr_{S \sim \mathcal{D}^m}[G(S) = h] > \eta$ satisfies

$$\mathrm{freq}_H(h) \geq \frac{7\eta}{8}.$$

Let us condition on this event. Then by the accuracy of the algorithm Hist, with probability at least $1 - \beta/2$ it produces a list L containing h^* together with a sequence of estimates that are accurate to within additive error $\eta/8$. In particular, h^* appears in L with an estimate $a_{h^*} \geq 7\eta/8 - \eta/8 \geq 3\eta/4$.

Now remove from L every item h with estimate $a_h < 3\eta/4$. Since every estimate is accurate to within $\eta/8$, this leaves a list with $|L| \leq 2/\eta$ that contains h^* with $\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha$. Hence, with probability at least $1 - \beta/3$, step 4 succeeds in identifying h^* with $\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha/2$.

The total sample complexity of the algorithm is $k \cdot m + n'$, which matches the asserted bound. □

5.4 Wrapping up

We now combine Theorem 23 (finite Littlestone dimension $\Longrightarrow$ global stability) with Theorem 30 (global stability $\Longrightarrow$ private learnability) to prove Theorem 3.

Proof of Theorem 3. Let $\mathcal{H}$ be a hypothesis class with Littlestone dimension d, and let $\mathcal{D}$ be any realizable distribution. Then, Theorem 23 guarantees, for $m = O(2^{2^{d+2}+1}4^{d+1} \cdot d/\alpha)$, the existence of a randomized algorithm $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ and a hypothesis f such that

$$\Pr[G(S) = f] \geq \frac{1}{(d+1)2^{2^{d+2}+1}} \text{ and } \operatorname{loss}_{\mathcal{D}}(f) \leq \alpha/2.$$

Taking $\eta = 1/(d+1)2^{2^{d+2}+1}$, Theorem 30 gives an (ε, δ)-differentially private learner with sample complexity

$$n = O\left(\frac{m \cdot \log(1/\eta\beta\delta)}{\eta\varepsilon} + \frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right) = O\left(\frac{2^{\tilde{O}(2^d)} + \log 1/\beta\delta}{\alpha\epsilon}\right). \qquad \square$$

6 CONCLUSION

We conclude this article with a few suggestions for future work:

(1) *Sharper quantitative bounds*: Our upper bound on the differentially private sample complexity of a class $\mathcal{H}$ has a double exponential dependence on the Littlestone dimension $\operatorname{Ldim}(\mathcal{H})$, whereas the lower bound by Alon et al. [5] depends on $\log^*(\operatorname{Ldim}(\mathcal{H}))$. The work by Kaplan et al. [53] shows that for thresholds, the lower bound is nearly tight (up to a polynomial factor). In a follow-up work to this article, Ghazi et al. [40] improved the upper bound to $\operatorname{poly}(\operatorname{Ldim}(\mathcal{H}))$ (roughly, with an exponent of 6). This is also tight up to polynomial factors for some classes, particularly those with maximal Littlestone dimension equal to $\log|\mathcal{H}|$. However the tower-of-exponents gap between the upper bound and the lower bound remains essentially the same (with two fewer levels). We thus pose the following question:

Can every class $\mathcal{H}$ be privately learned with sample complexity $\operatorname{poly}(\operatorname{VC}(\mathcal{H}), \log^*(\operatorname{Ldim}(\mathcal{H})))$?

(2) *Characterizing private query release*: Another fundamental problem in differentially private data analysis is the query release, or equivalently, the data sanitization problem: given a class $\mathcal{H}$ and a sensitive dataset S, output a synthetic dataset $\hat{S}$ such that $h(S) \approx h(\hat{S})$ for every $h \in \mathcal{H}$. In earlier versions of this work, we asked whether a finite Littlestone dimension characterizes when this task is possible. This was shown to be true by Bousquet et al. [16] and Ghazi et al. [40]. (Bousquet et al. [16] showed how to transform a *proper* private learner to a sanitizer, and Ghazi et al. [40] proved that every Littlestone class can be learned properly.) However, as with private classification, massive quantitative gaps between the known upper and lower bounds remain.

(3) *Oracle-efficient learning*: Neel et al. [69] recently began a systematic study of oracle-efficient learning algorithms: differentially private algorithms that are computationally efficient when given oracle access to their non-private counterparts. The main open question left by their work is whether *every* privately learnable concept class can be learned in an oracle-efficient manner. Our characterization shows that this is possible if and only if Littlestone classes admit oracle-efficient learners.

(4) *General loss functions*: It is natural to explore whether the equivalence between online and private learning extends beyond binary classification (which corresponds to the 0-1 loss) to regression and other real-valued losses. These more general loss functions have been studied in subsequent work [6, 20, 42, 50], although the problem of exactly characterizing private learnability in the regression setting remains open.

(5) *Global stability*: It would be interesting to perform a thorough investigation of global stability and to explore potential connections to other forms of stability in learning theory, including uniform hypothesis stability [15], PAC-Bayes [63], local statistical stability [60], and others.

(6) *Differentially private boosting*: Can the type of private boosting presented in Section 2.4 be done algorithmically, and ideally, efficiently?

APPENDIX

A PROOF OF THEOREM 10

In this appendix, we prove Theorem 10. Throughout the proof, a labeled binary tree means a full binary tree whose internal vertices are labeled by instances.

The second part of the theorem is easy. If $\mathcal{H}$ contains 2^t thresholds, then there are $h_i \in \mathcal{H}$ for $0 \le i < 2^t$ and x_j for $0 \le j < 2^t - 1$ such that $h_i(x_j) = 0$ for $j < i$ and $h_i(x_j) = 1$ for $j \ge i$. Define a labeled binary tree of height t corresponding to the binary search process. In other words, the root is labeled by $x_{2^{t-1}-1}$, its left child by $x_{2^{t-1}+2^{t-2}-1}$ and its right child by $x_{2^{t-1}-2^{t-2}-1}$ and so on. If the label of an internal vertex of distance q from the root, where $0 \le q \le t-1$, is x_p, then the label of its left child is $x_{p+2^{t-q-1}}$ and the label of its right child is $x_{p-2^{t-q-1}}$. It is easy to check that the root-to-leaf path corresponding to each of the functions h_i leads to leaf number i from the right among the leaves of the tree (counting from 0 to $2^t - 1$).

To prove the first part of the theorem, we first define the notion of a subtree T' of depth h of a labeled binary tree T by induction on h. Any leaf of T is a subtree of height 0. For $h \ge 1$, a subtree of height h is obtained from an internal vertex of T together with a subtree of height $h-1$ of the tree rooted at its left child and a subtree of height $h-1$ of the tree rooted at its right child. Note that if T is a labeled tree and it is shattered by the class $\mathcal{H}$, then any subtree T' of it with the same labeling of its internal vertices is shattered by the class $\mathcal{H}$. With this definition, we prove the following simple lemma.

Lemma 31. *Let p, q be positive integers and let T be a labeled binary tree of height $p + q - 1$ whose internal vertices are colored by two colors: red and blue. Then T contains either a subtree of height p in which all internal vertices are red (a red subtree) or a subtree of height q in which all vertices are blue (a blue subtree).*

Proof. We apply induction on $p + q$. The result is trivial for $p = q = 1$ as the root of T is either red or blue. Assuming the assertion holds for $p' + q' < p + q$, let T be of height $p + q - 1$. Without loss of generality, assume the root of t is red. If $p = 1$ we are done, as the root together with a leaf in the subtree of its left child and one in the subtree of its right child form a red subtree of height p. If $p > 1$, then by the induction hypothesis, the tree rooted at the left child of the root of T contains either a red subtree of height $p - 1$ or a blue subtree of height q, and the same applies to the tree rooted at the right child of the root. If at least one of them contains a blue subtree as earlier, we are done; otherwise, the two red subtrees together with the root provide the required red subtree. □

We can now prove the first part of the theorem, showing that if the Littlestone dimension of $\mathcal{H}$ is at least $2^{t+1} - 1$, then $\mathcal{H}$ contains $t + 2$ thresholds. We apply induction on t. If $t = 0$, we have a tree of height 1 shattered by $\mathcal{H}$. Its root is labeled by some variable x_0 and as it is shattered there are two functions $h_0, h_1 \in \mathcal{H}$ so that $h_0(x_0) = 1, h_1(x_0) = 0$, meaning that $\mathcal{H}$ contains two thresholds, as needed. Assuming the desired result holds for $t - 1$, we prove it for t, $t \ge 1$. Let T be a labeled binary tree of height $2^{t+1} - 1$ shattered by $\mathcal{H}$. Let h be an arbitrary member of $\mathcal{H}$ and define a two coloring of the internal vertices of T as follows. If an internal vertex is labeled by x and $h(x) = 1$, then color it red, else color it blue. Since $2^{t+1} - 1 = 2 \cdot 2^t - 1$, Lemma 31 with $p = q = 2^t$ implies that T contains either a red or a blue subtree T' of height 2^t. In the first case, define $h_0 = h$ and let X be the set of all variables x so that $h(x) = 1$. Let x_0 be the root of T' and let T'' be the subtree of T' rooted at the left child of T'. Let $\mathcal{H}'$ be the set of all $h' \in \mathcal{H}$ so that $h'(x_0) = 0$. Note that $\mathcal{H}'$ shatters the tree T'', and that the depth of T'' is $2^t - 1$. We can thus apply the induction hypothesis and get a set of $t+1$ thresholds $h_1, h_2, \ldots, h_{t+1} \in \mathcal{H}'$ and variables $x_1, x_2, \ldots, x_t \in X$ so that $h_i(x_j) = 1$ iff $j \geq i$. Adding h_0 and x_0 to these we get the desired $t+2$ thresholds.

Similarly, if T contains a blue subtree T', define $h_{t+1} = h$ and let X be the set of all variables x so that $h(x) = 0$. In this case denote the root of T' by x_t and let T'' be the subtree of T' rooted at the right child of T'. Let $\mathcal{H}'$ be the set of all $h' \in \mathcal{H}$ so that $h'(x_t) = 1$. As before, $\mathcal{H}'$ shatters the tree T'' whose depth is $2^t - 1$. By the induction hypothesis we get $t+1$ thresholds $h_0, h_1, \ldots, h_t$ and variables $x_0, x_1, \ldots, x_{t-1} \in X$ so that $h_i(x_j) = 1$ if and only if $j \geq i$, and the desired result follows by appending to them h_{t+1} and x_t. This completes the proof. □

ACKNOWLEDGMENTS

We would like to thank Amos Beimel and Uri Stemmer for pointing out and helping to fix a mistake in the derivation of Corollary 4 in a previous version. We also thank Raef Bassily and Yuval Dagan for providing useful comments and for insightful conversations. Last but not least, we thank the two anonymous reviewers for their comments and suggestions, which helped us to improve the presentation of this article.

REFERENCES

[1] Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. 2008. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT'08).* 263–274.

[2] Jacob D. Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. 2017. Online learning via differential privacy. *CoRR* abs/1711.10019 (2017).

[3] Naman Agarwal and Karan Singh. 2017. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017 (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.). Vol. 70. PMLR, 32–40. http://proceedings.mlr.press/v70/agarwal17a.html.

[4] Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. 2020. Closure properties for private classification and online prediction. *arXiv preprint arXiv:2003.04509* (2020).

[5] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. 2019. Private PAC learning implies finite Littlestone dimension. In *Proceedings of the 51st Annual ACM Symposium on the Theory of Computing (STOC'19).* ACM, New York, NY.

[6] Srinivasan Arunachalam, Yihui Quek, and John A. Smolin. 2021. Private learning implies quantum stability. *CoRR* abs/2102.07171 (2021). https://arxiv.org/abs/2102.07171.

[7] Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. 2018. Learners that use little information. In *Algorithmic Learning Theory, ALT 2018, 7–9 April 2018, Lanzarote, Canary Islands, Spain (Proceedings of Machine Learning Research)*, Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan (Eds.). Vol. 83. PMLR, 25–55. http://proceedings.mlr.press/v83/bassily18a.html.

[8] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2014. Bounds on the sample complexity for private learning and private data release. *Machine Learning* 94, 3 (2014), 401–437.

[9] Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. 2019. Private center points and learning of halfspaces. In *Conference on Learning Theory, COLT 2019, 25–28 June 2019, Phoenix, AZ, USA (Proceedings of Machine Learning Research)*, Alina Beygelzimer and Daniel Hsu (Eds.). Vol. 99. PMLR, 269–282.

[10] Amos Beimel, Kobbi Nissim, and Uri Stemmer. 2015. Learning privately with labeled and unlabeled examples. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms.* 461–477.

[11] Amos Beimel, Kobbi Nissim, and Uri Stemmer. 2016. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory of Computing* 12, 1 (2016), 1–61.

[12] Amos Beimel, Kobbi Nissim, and Uri Stemmer. 2019. Characterizing the sample complexity of pure private learners. *Journal of Machine Learning Research* 20, 146 (2019), 1–33. http://jmlr.org/papers/v20/18-269.html.

[13] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. 2009. Agnostic online learning. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT'09).* 1–11.

[14] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* 36, 4 (1989), 929–965.

[15] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *Journal of Machine Learning Research* 2 (2002), 499–526. http://jmlr.org/papers/v2/bousquet02a.html.

[16] Olivier Bousquet, Roi Livni, and Shay Moran. 2019. Passing tests without memorizing: Two models for fooling discriminators. arXiv:cs.LG/1902.03468 *(2019)*.

[17] Mark Bun. 2020. A computational separation between private learning and online learning. In *Proceedings of the 34th Joint Conference on Neural Information Processing Systems (NeurIPS'20)*.

[18] Mark Bun, Marco L. Carmosino, and Jessica Sorrell. 2020. Efficient, noise-tolerant, and private learning via boosting. *CoRR* abs/2002.01100 (2020).

[19] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. 2018. Composable and versatile privacy via truncated CDP. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing (STOC'18)*. ACM, New York, NY, 74–86.

[20] Mark Bun, Marco Gaboardi, and Satchit Sivakumar. 2021. Multiclass versus binary differentially private PAC learning. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21)*. 1–12.

[21] Mark Bun, Kobbi Nissim, and Uri Stemmer. 2016. Simultaneous private learning of multiple concepts. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science (ITCS'16)*. ACM, New York, NY, 369–380.

[22] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. 2015. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS'15)*. IEEE, Los Alamitos, CA, 634–649.

[23] Mark Mar Bun. 2016. *New Separations in the Complexity of Differential Privacy*. Ph.D. Dissertation. Graduate School of Arts & Sciences, Harvard University.

[24] Nicolò Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press. https://doi.org/10.1017/CBO9780511546921

[25] Hunter Chase and James Freitag. 2018. Model theory and machine learning. *arXiv preprint arXiv:1801.06566* (2018).

[26] Hunter Chase and James Freitag. 2019. Model theory and machine learning. *Bulletin of Symbolic Logic* 25, 03 (Feb. 2019), 319–332. https://doi.org/10.1017/bsl.2018.71

[27] Alon Cohen, Avinatan Hassidim, Haim Kaplan, Yishay Mansour, and Shay Moran. 2019. Learning to screen. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS'19)*. 8612–8621. http://papers.nips.cc/paper/9067-learning-to-screen.

[28] José R. Correa, Paul Dütting, Felix A. Fischer, and Kevin Schewior. 2019. Prophet inequalities for I.I.D. random variables from an unknown distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24–28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, New York, NY, 3–17. https://doi.org/10.1145/3328526.3329627

[29] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. 2019. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization (UMAP'19 Adjunct)*. ACM, New York, NY, 309–315. https://doi.org/10.1145/3314183.3323847

[30] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8–10, 2012*, Shafi Goldwasser (Ed.). ACM, New York, NY, 214–226.

[31] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*. Lecture Notes in Computer Science, Vol. 4004. Springer, 486–503.

[32] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*. 371–380.

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

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

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

[36] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*. IEEE, Los Alamitos, CA, 51–60.

[37] P. Erdős and R. Rado. 1952. Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London Mathematical Society* s3-2, 1 (1952), 417–439. https://doi.org/10.1112/plms/s3-2.1.417

[38] Vitaly Feldman and David Xiao. 2015. Sample complexity bounds on differentially private learning via communication complexity. *SIAM Journal on Computing* 44, 6 (2015), 1740–1764.

[39] Eran Gat and Shafi Goldwasser. 2011. Probabilistic search algorithms with unique answers and their cryptographic applications. *Electronic Colloquium on Computational Complexity* 18 (2011), 136.

[40] Badih Ghazi, Noah Golowich, Ravi Kumar, and Pasin Manurangsi. 2020. Sample-efficient proper PAC learning with approximate differential privacy. *CoRR* abs/2012.03893 (2020).

[41] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. 2021. User-level private learning via correlated sampling. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21).* 1–13.

[42] Noah Golowich. 2021. Differentially private nonparametric regression under a growth condition. In *Conference on Learning Theory, COLT 2021, 15–19 August 2021, Boulder, Colorado, USA (Proceedings of Machine Learning Research)*, Mikhail Belkin and Samory Kpotufe (Eds.). Vol. 134. PMLR, 2149–2192.

[43] Noah Golowich and Roi Livni. 2021. Littlestone classes are privately online learnable. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21).* 1–12.

[44] Alon Gonen, Elad Hazan, and Shay Moran. 2019. Private learning implies online learning: An efficient reduction. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS'19).*

[45] R. L. Graham, B. L. Rothschild, and J. H. Spencer. 1990. *Ramsey Theory.* Wiley. https://books.google.com/books?id=55oXT60dC54C.

[46] Elad Hazan. 2016. Introduction to online convex optimization. *Foundations and Trends in Optimization* 2, 3–4 (Aug. 2016), 157–325. https://doi.org/10.1561/2400000013

[47] Wilfrid Hodges. 1997. *A Shorter Model Theory.* Cambridge University Press, New York, NY.

[48] Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. 2022. Reproducibility in learning. arXiv:cs.LG/2201.08430 (2022).

[49] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. 2019. The role of interactivity in local differential privacy. In *Proceedings of the 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS'19).*

[50] Young Hun Jung, Baekjin Kim, and Ambuj Tewari. 2020. On the equivalence between online and private learnability beyond binary classification. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20).* 1–10.

[51] Adam Kalai and Santosh Vempala. 2002. *Geometric Algorithms for Online Optimization.* Technical Report MIT-LCS-TR-861. Massachusetts Institute of Technology, Cambridge, MA.

[52] Adam Kalai and Santosh Vempala. 2005. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences* 71, 3 (Oct. 2005), 291–307. https://doi.org/10.1016/j.jcss.2004.10.016

[53] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. 2019. Privately learning thresholds: Closing the exponential gap. arXiv:cs.DS/1911.10137 (2019).

[54] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. 2020. Privately learning thresholds: Closing the exponential gap. In *Conference on Learning Theory, COLT 2020, 9–12 July 2020, Virtual Event [Graz, Austria] (Proceedings of Machine Learning Research)*, Jacob D. Abernethy and Shivani Agarwal (Eds.). Vol. 125. PMLR, 2263–2285.

[55] Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. 2020. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20).* 1–10.

[56] Marek Karpinski and Angus Macintyre. 1997. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. *Journal of Computer and System Sciences* 54, 1 (1997), 169–176.

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

[58] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. 2009. Releasing search queries and clicks privately. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09).* ACM, New York, NY, 171–180.

[59] Michael C. Laskowski. 1992. Vapnik-Chervonenkis classes of definable sets. *Journal of the London Mathematical Society* 2, 2 (1992), 377–384.

[60] Katrina Ligett and Moshe Shenfeld. 2019. A necessary and sufficient stability notion for adaptive generalization. *CoRR* abs/1906.00930 (2019).

[61] Nick Littlestone. 1987. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2, 4 (1987), 285–318.

[62] Roi Livni and Pierre Simon. 2013. Honest compressions and their application to compression schemes. In *Proceedings of the Conference on Learning Theory.* 77–92.

[63] David A. McAllester. 1999. Some PAC-Bayesian theorems. *Machine Learning* 37, 3 (1999), 355–363.

[64] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07).* IEEE, Los Alamitos, CA, 94–103.

[65] Shlomo Moran, Marc Snir, and Udi Manber. 1985. Applications of Ramsey's theorem to decision tree complexity. *Journal of the ACM* 32, 4 (1985), 938–949.

[66] D. Mubayi and A. Suk. 2017. A survey of quantitative bounds for hypergraph Ramsey problems. *ArXiv e-prints* arXiv:math.CO/1707.04229 *(2017).*

[67] Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. 2018. A direct sum result for the information complexity of learning. In *Conference on Learning Theory, COLT 2018, Stockholm, Sweden, 6–9 July 2018 (Proceedings of Machine Learning Research)*, Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (Eds.). Vol. 75. PMLR, 1547–1568. http://proceedings.mlr.press/v75/nachum18a.html.

[68] Ido Nachum and Amir Yehudayoff. 2019. Average-case information complexity of learning. In *Algorithmic Learning Theory, ALT 2019, 22–24 March 2019, Chicago, Illinois, USA (Proceedings of Machine Learning Research)*, Aurélien Garivier and Satyen Kale (Eds.). Vol. 98. PMLR, 633–646. http://proceedings.mlr.press/v98/nachum19a.html.

[69] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2019. How to use heuristics for differential privacy. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)*. 72–93.

[70] Igor Carboni Oliveira and Rahul Santhanam. 2018. Pseudo-derandomizing learning and approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'18)*, E. Blais, J. D. P. Rolim, and D. Steurer (Eds.). Leibniz International Proceedings in Informatics. Schloss Dagstuhl–Leibniz-Zentrum fur Informatik, Dagstuhl Publishing, Dagstuhl, Germany, Article 55, 19 pages.

[71] Cathy O'Neil. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Crown, New York, NY.

[72] Menachem Sadigurschi and Uri Stemmer. 2021. On the sample complexity of privately learning axis-aligned rectangles. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS'21)*. 1–14.

[73] Shai Shalev-Shwartz. 2012. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4, 2 (Feb. 2012), 107–194. https://doi.org/10.1561/2200000018

[74] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, New York, NY.

[75] Shai Shalev-Shwartz and Yoram Singer. 2007. A primal-dual perspective of online learning algorithms. *Machine Learning* 69, 2 (2007), 115–142. https://doi.org/10.1007/s10994-007-5014-x

[76] Saharon Shelah. 1978. *Classification Theory and the Number of Non-isomorphic Models.* North-Holland Publishing, Amsterdam, Netherlands.

[77] Salil Vadhan. 2017. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, Yehuda Lindell (Ed.). Springer International Publishing AG, Cham, Switzerland, 347–450.

[78] Leslie G. Valiant. 1984. A theory of the learnable. *Communications of the ACM* 27, 11 (1984), 1134–1142.

[79] Vladimir Vapnik and Alexey Chervonenkis. 1974. *Theory of Pattern Recognition.* Nauka.

Received September 2020; revised January 2022; accepted March 2022