Sample-efficient proper PAC learning with approximate differential privacy

Badih Ghazi* Noah Golowich† Ravi Kumar* Pasin Manurangsi*

December 4, 2020

Abstract

In this paper we prove that the sample complexity of properly learning a class of Littlestone dimension d with approximate differential privacy is $\tilde{O}(d^6)$, ignoring privacy and accuracy parameters. This result answers a question of Bun et al. (FOCS 2020) by improving upon their upper bound of $2^{O(d)}$ on the sample complexity. Prior to our work, finiteness of the sample complexity for privately learning a class of finite Littlestone dimension was only known for improper private learners, and the fact that our learner is proper answers another question of Bun et al., which was also asked by Bousquet et al. (NeurIPS 2020). Using machinery developed by Bousquet et al., we then show that the sample complexity of sanitizing a binary hypothesis class is at most polynomial in its Littlestone dimension and dual Littlestone dimension. This implies that a class is sanitizable if and only if it has finite Littlestone dimension. An important ingredient of our proofs is a new property of binary hypothesis classes that we call *irreducibility*, which may be of independent interest.

1 Introduction

Machine learning algorithms are often trained on datasets consisting of sensitive data, such as in medical or social network applications. Protecting the privacy of the users' data is of importance, both from an ethical perspective [RK19] and to maintain compliance with an increasing number of laws and regulations [Par14, NBW+18, CN20]. The notion of *differential privacy* [DMNS06, DR14, Vad17] provides a formal framework for controlling the privacy-accuracy tradeoff in numerous settings involving private data release, and it has played a central role in the development of privacy-preserving algorithms.

In the body of work on private learning algorithms, a significant amount of effort has gone into developing algorithms for the private PAC model [KLN+08], namely the setting of differentially private binary classification (see Section 2.1 for a formal definition). Some papers on this fundamental topic include [KLN+08, BBKN14, BNSV15, FX14, BNS14, BDRS18, BNS19, ALMM19, KLM+20, BLM20b, NRW19, Bun20]. A remarkable recent development [ALMM19, BLM20b] in this area is the result that a hypothesis class $\mathcal{F}$ of binary classifiers is learnable with approximate differential privacy (Definition 2.2) if and only if it is online learnable, i.e., has finite Littlestone dimension (Definition 2.5). Specifically, Alon et al. [ALMM19] showed that any differentially private learning algorithm with at most constant error for a class of Littlestone dimension d must

*Google Research, Mountain View, CA. badihghazi@gmail.com, ravi.k53@gmail.com, pasin@google.com.

†MIT EECS, Cambridge, MA. Supported at MIT by a Fannie & John Hertz Foundation Fellowship, an MIT Akamai Fellowship, and an NSF Graduate Fellowship. This work was done while interning at Google Research. nzg@mit.edu.

use at least $\Omega(\log^\star d)$ samples. Conversely, Bun et al. [BLM20b] showed that if $\mathcal{F}$ has Littlestone dimension d, then there is a differentially private learning algorithm for $\mathcal{F}$ with error $\alpha > 0$ using $2^{O(d)}/\alpha$ samples.[1]

1.1 Results

In this paper, we resolve two open questions posed by Bun et al. [BLM20b] and Bousquet et al. [BLM20a]: first, we introduce a new private learning algorithm with sample complexity *polynomial* in the Littlestone dimension d of the class $\mathcal{F}$, thus improving exponentially on the bound $2^{O(d)}$ from [BLM20b]. Answering a second question of [BLM20b], we show how to make our private learner *proper* (whereas the learner from [BLM20b] was improper). Whether privately properly learning classes of finite Littlestone dimension is possible was also asked by Bousquet et al. [BLM20a, Question 1]. Theorem 1.1 states our main result:

Theorem 1.1 (Private proper PAC learning; informal version of Theorem 6.4)**.** *Let $\mathcal{F}$ be a class of hypotheses $f : \mathcal{X} \to \{-1, 1\}$, of Littlestone dimension d. For any $\varepsilon, \delta, \alpha \in (0, 1)$, for some $n = \tilde{O}\left(\frac{d^6}{\varepsilon\alpha^2}\right)$, there is an (ε, δ)-differentially private algorithm which, given n i.i.d. samples from any realizable distribution P on $\mathcal{X} \times \{-1, 1\}$, with high probability outputs a classifier $\hat{f} \in \mathcal{F}$ with classification error over P at most α.*

The theorem statement above treats the case where the distribution P over $\mathcal{X} \times \{-1, 1\}$ is *realizable*, namely that there exists some $f^\star \in \mathcal{F}$ so that P is supported on pairs $(x, f^\star(x))$. A generic reduction of [ABMS20] allows us to show essentially the same sample complexity bound as in Theorem 1.1 for the non-realizable (i.e., *agnostic*) setting (see Corollary 6.5). We also remark that it is impossible to obtain a sample complexity bound better than $n = O(d)$ in the context of Theorem 1.1 if we insist that the bound depends on the class $\mathcal{F}$ only through the Littlestone dimension d. This follows because for any $d \in \mathbb{N}$, there are classes $\mathcal{F}$ whose Littlestone and VC dimensions are both equal to d (for instance, the class of all binary hypotheses on d points), and it is well-known that the VC dimension characterizes the sample complexity of learning a class (in the absence of privacy) [Vap98].

The question posed in [BLM20a, BLM20b] (and answered by Theorem 1.1) of whether classes of finite Littlestone dimension have private proper learners is motivated by a connection between proper private learning and *private query release* established in [BLM20a]. The problem of private query release, or *sanitization* [BLR08, BNS14], for a class $\mathcal{F}$ has an extensive history, described in Section 1.2. It is defined as follows: given $\alpha > 0$, a *sanitizer* with sample complexity $n \in \mathbb{N}$ is given as input a dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \{-1, 1\})^n$. The sanitizer must output a function $\mathrm{Est} : \mathcal{F} \to [0, 1]$, which is differentially private for the input S, so that with high probability, for each $f \in \mathcal{F}$, $|\mathrm{Est}(f) - \mathrm{err}_S(f)| \leqslant \alpha$, where $\mathrm{err}_S(f) := \frac{1}{n} \cdot |\{i \in [n] : f(x_i) \neq y_i\}|$. Bousquet et al. [BLM20a] showed that the existence of a private proper learner for a class $\mathcal{F}$ implies the existence of a sanitizer for $\mathcal{F}$; as a corollary of their result and of Theorem 1.1 we therefore obtain the following:

Corollary 1.2 (Private query release; informal version of Corollary 6.7)**.** *Let $\mathcal{F}$ be a class of hypotheses $f : \mathcal{X} \to \{-1, 1\}$ of Littlestone dimension d and dual Littlestone dimension $d^\star$. For any $\varepsilon, \delta, \alpha \in [0, 1]$, there is an (ε, δ)-differentially private algorithm that for some $n = \mathrm{poly}(d, d^\star, 1/\varepsilon, 1/\alpha, \log 1/\delta)$, takes as input a dataset S of size n and outputs a function $\mathrm{Est} : \mathcal{F} \to [0, 1]$ so that with high probability, for all $f \in \mathcal{F}$, $|\mathrm{Est}(f) - \mathrm{err}_S(f)| \leqslant \alpha$.*

[1] This bound ignores the dependence on the privacy parameters ε, δ. Moreover, it applies to the realizable setting; a slightly weaker bound was shown in [BLM20b] for the agnostic setting.

It is known that the dual Littlestone dimension $d^\star$ of a class $\mathcal{F}$ is finite if and only if the Littlestone dimension d is finite; in fact, we have $d^\star \leqslant 2^{2^{d+2}} - 2$ [Bha17, Corollary 3.6]. Thus, Corollary 1.2 implies that a class $\mathcal{F}$ is *sanitizable* (roughly, that it has a sanitizer with sample complexity $\mathrm{poly}(1/\alpha)$; see Definition 2.3 for a formal version) if it has finite Littlestone dimension. The converse, namely that any sanitizable class must have finite Littlestone dimension, follows as a consequence of a result of [BNSV15], as discussed in Section 6.2. Summarizing, we have the following:

Corollary 1.3. *A hypothesis class $\mathcal{F}$ is sanitizable if and only if it has finite Littlestone dimension.*

Techniques: irreducibility The main technique that allows us to both improve the exponential bound $2^{O(d)}$ on the sample complexity from [BLM20b] to a polynomial dependence, and to make the learner proper in Theorem 1.1 is a property of hypothesis classes we introduce, called *irreducibility* (Section 4). Roughly speaking, a binary hypothesis class $\mathcal{G}$ of Littlestone dimension d on domain $\mathcal{X}$ is irreducible if any binary tree of bounded depth labeled by elements of $\mathcal{X}$ has a leaf such that the restriction of $\mathcal{G}$ to that leaf still has Littlestone dimension d. The exponential sample complexity bound in [BLM20b] arises (in part) for the following reason: the main sub-procedure in their algorithm operates in a sequence of $d = \mathrm{Ldim}(\mathcal{F})$ steps, maintaining a class of candidate hypotheses; at the end of the d steps, this class will have Littlestone dimension 0 (i.e., consists of a single hypothesis), and will be the hypothesis output by the sub-procedure. Each of these d steps decreases the Littlestone dimension of the class of candidate hypotheses by 1 and increases the number of samples needed by a constant factor, leading to $2^{O(d)}$ samples overall. The notion of irreducibility allows us to show that certain intermediate classes of candidate hypotheses can be "sufficiently stable" to allow us to output a hypothesis associated with the intermediate class in a private way. This allows us to avoid the exponential blowup in d associated with decreasing the Littlestone dimension of the candidate hypotheses all the way to 0. We believe that the notion of irreducibility may be useful in other applications. We provide a more detailed overview of our proofs in Section 3.

1.2 Related work

Sample complexity of differentially private learning The sample complexity of PAC learning with *pure* differential privacy (namely, $(\varepsilon, 0)$-differential privacy) is well-understood. The seminal work of Kasiviswanathan et al. [KLN+08] showed that a finite class $\mathcal{F}$ consisting of hypotheses $f : \mathcal{X} \to \{-1, 1\}$ can be learned with pure differential privacy with sample complexity $O(\log|\mathcal{F}|)$ (in this section we omit dependence on the privacy and accuracy parameters). By the Sauer–Shelah lemma, $\log(|\mathcal{F}|) \leqslant O(\mathrm{VCdim}(\mathcal{F}) \cdot \log(|\mathcal{X}|))$; moreover, the multiplicative gap between $\mathrm{VCdim}(\mathcal{F})$, which characterizes the sample complexity of *non-private* learning, and $\log|\mathcal{F}|$, can be as large as $\log|\mathcal{X}|$. To obtain a more precise result, Beimel et al. [BNS19] introduced a complexity measure for a class $\mathcal{F}$ of binary hypotheses, known as the *probabilistic representation dimension* of $\mathcal{F}$, which they showed to characterize the sample complexity of (improperly) learning $\mathcal{F}$ with pure differential privacy up to a constant factor (see also [BBKN14]). Feldman and Xiao [FX14] showed that, in turn, the probabilistic representation dimension is characterized, up to a constant factor, by the one-way public coin communication complexity of an evaluation problem associated to $\mathcal{F}$. As a corollary of this result, they established that the sample complexity of learning $\mathcal{F}$ with pure privacy is always at least $\Omega(\mathrm{Ldim}(\mathcal{F}))$, where $\mathrm{Ldim}(\mathcal{F})$ denotes the Littlestone dimension of $\mathcal{F}$.

The current understanding of the sample complexity of learning with *approximate* differential privacy (namely, (ε, δ)-differential privacy with δ negligible as a function of the number of users),

which is our focus in this paper, is much less complete. The class of threshold functions on a domain of size 2^d, which has Littlestone dimension d, is known to be learnable with approximate privacy with sample complexity $O((\log^\star d)^{1.5})$ [KLM+20], showing that the sample complexity of learning a class $\mathcal{F}$ with approximate privacy can be much less than its Littlestone dimension (see also [BNSV15, BNS14, BDRS18], which obtained weaker bounds). As mentioned previously, the best-known lower bound for the sample complexity of (improperly) privately learning a class of Littlestone dimension d is $\Omega(\log^\star d)$ [ALMM19]; our Theorem 1.1 gives the best known upper bound in terms of Littlestone dimension. In a different direction, some recent papers have investigated the sample complexity of privately learning halfspaces [KMST20, KSS20, BMNS19].

Differentially private query release The problem of private data release (also known as sanitization; see Section 2.2 for a formal definition) for a binary hypothesis class $\mathcal{F}$ dates back to Blum et al. [BLR08], who showed that the sample complexity of private sanitization is bounded above by $O(\mathrm{VCdim}(\mathcal{F}) \cdot \log|\mathcal{X}|) \leqslant O(\log|\mathcal{F}| \cdot \log|\mathcal{X}|)$. This bound was later improved to $\tilde{O}(\log|\mathcal{F}| \cdot \sqrt{\log|\mathcal{X}|})$ by Hardt and Rothblum [HR10],[2] which is known to be essentially the best possible dependence on $|\mathcal{F}|, |\mathcal{X}|$ attainable for a broad range of values of $|\mathcal{F}|, |\mathcal{X}|$ [BUV14]. Many works have developed more fine-grained bounds on the sample complexity of sanitization in terms of geometrical properties of $\mathcal{F}$ [BDKT12, BBNS19, ENU20, HT10, Nik15, NTZ12], and several have additionally studied computational considerations for this problem [DNR+09, DRV10, HLM12, RR10]. However, the upper bounds on the sample complexity of sanitization obtained by all of these works scale at least polynomially with either $\log|\mathcal{X}|$ or $\log|\mathcal{F}|$; thus, they implicitly assume that $\mathcal{X}$ or $\mathcal{F}$ (or both) is finite. In addition to being of purely theoretical interest, establishing sample complexity bounds with no explicit dependence on $|\mathcal{X}|, |\mathcal{F}|$ (and thus which can apply when $|\mathcal{X}|$ and $|\mathcal{F}|$ are infinite) could lead to significant gains even in cases when they are finite since in many natural settings, $|\mathcal{X}|, |\mathcal{F}|$ are exponentially large in parameters such as dimensionality of the data. The question of removing the $\mathrm{poly}\log|\mathcal{F}|$ factors in existing bounds has also been asked in [Vad17]: Questions 5.24 and 5.25 in [Vad17] ask for a characterization of the sample complexity of sanitization up to "small" approximation factors. In the proof of Corollary 1.3 it is established that the sample complexity of sanitizing a class of Littlestone dimension d is between $\Omega(\log^\star d)$ and $2^{O(2^d)}$. This gap is definitely not "small" by any means, but for infinite $|\mathcal{F}|, |\mathcal{X}|$, it is the first finite approximation factor to the best of our knowledge.

Online learning and Littlestone dimension The Littlestone dimension of a hypothesis class $\mathcal{F}$ is known to be equal to the optimal mistake bound in the realizable setting of online learning [Lit87, Sha12]. Moreover, it characterizes the optimal regret of an online learning algorithm in the agnostic setting up to a logarithmic factor: the optimal regret $\mathrm{Reg}(T)$ for an online learning algorithm with respect to a class of Littlestone dimension d satisfies $\Omega(\sqrt{dT}) \leqslant \mathrm{Reg}(T) \leqslant O(\sqrt{dT\log T})$ [BPS09, Sha12]. Therefore, Theorem 1.1 implies that the sample complexity of privately learning a binary hypothesis class $\mathcal{F}$ is bounded above by a polynomial in the sample complexity of online learning of $\mathcal{F}$ (in either the realizable or agnostic setting).

Many prior works have investigated the connection between online and private learnability in slightly different settings from ours. Inherent stability-type properties of private learning algorithms have been used to show that certain problems have online learning algorithms [GHM19, AJL+19, NRW19, ALMM19]. Bun [Bun20] shows that such a reduction is not possible in a generic sense if it is required to be computationally efficient. In the opposite direction, [AS17, BLM20b] develop differentially private algorithms to solve problems which are online learnable.

[2] The $\tilde{O}$ hides factors logarithmic in $\log|\mathcal{F}|$ and $\log|\mathcal{X}|$.

1.3 Organization of the paper

In Section 2, we review some preliminaries regarding private query release, private PAC learning, and online learning. In Section 3, we outline the proof of Theorem 1.1. In Section 4 we introduce a central notion used in our proof, namely that of *irreducibility*, and prove some basic properties of it. In Sections 5 and 6 we prove Theorem 1.1 and its corollaries for private query release. Concluding remarks are in Section 7.

2 Preliminaries

We will use the script notation (e.g., $\mathcal{F}, \mathcal{X}$) to denote sets (e.g., sets of data points or sets of binary hypotheses). For sets $\mathcal{S}, \mathcal{T}$, we write $\mathcal{S} \subset \mathcal{T}$ to mean that $\mathcal{S}$ is a (not necessarily proper) subset of $\mathcal{T}$.

2.1 PAC learning

We use standard notation and terminology regarding PAC learning (see, e.g., [SB14]). Let $\mathcal{X}$ be an arbitrary set and let $\{-1, 1\}$ be the label set. We suppose throughout the paper that $\mathcal{X} \times \{-1, 1\}$ is endowed with a σ-algebra Σ. For $x \in \mathcal{X}, y \in \{-1, 1\}$, let $\delta_{(x,y)}$ denote the point measure at (x, y), i.e., for $A \in \Sigma$, $\delta_{(x,y)}(A)$ is defined to be 1 if $(x, y) \in A$, and 0 otherwise.

A *hypothesis* is a function $f : \mathcal{X} \to \{-1, 1\}$. We write the set of all hypotheses on $\mathcal{X}$ as $\{-1, 1\}^{\mathcal{X}}$. An *example* is a pair $(x, y) \in \mathcal{X} \times \{-1, 1\}$, and for $n \in \mathbb{N}$, a *dataset* S_n is a set of n examples, $S_n := \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Given such a dataset, define the *empirical measure* $\hat{P}_{S_n} := \frac{1}{n} \sum_{i=1}^{n} \delta_{(x_i,y_i)}$ on $\mathcal{X} \times \{-1, 1\}$. For a distribution P on $\mathcal{X} \times \{-1, 1\}$, let P^n be the distribution of $S_n \in (\mathcal{X} \times \{-1, 1\})^n$ consisting of n i.i.d. draws from P.

Definition 2.1 (Error of a hypothesis)**.** Let P be a probability distribution on $\mathcal{X} \times \{-1, 1\}$. The *error* (or *loss*) of a hypothesis $f : \mathcal{X} \to \{-1, 1\}$ is defined as

$$\mathrm{err}_P(f) := \Pr_{(x,y)\sim P} [f(x) \neq y] .$$

The *empirical error* of a hypothesis f with respect to a dataset S_n is defined to be $\mathrm{err}_{\hat{P}_{S_n}}(f)$. At times we will abbreviate $\mathrm{err}_{\hat{P}_{S_n}}(f)$ by writing $\mathrm{err}_{S_n}(f)$ instead. In this paper we will consider hypothesis classes $\mathcal{F} \subset \{-1, 1\}^{\mathcal{X}}$; to avoid having to make technical measurability assumptions on $\mathcal{F}, \mathcal{X}$, we will assume throughout that $\mathcal{F}$ and $\mathcal{X}$ are countable. (We refer the reader to [Dud99, Chapter 5] for a discussion of such assumptions in the case that countability does not hold. We remark that it is necessary to make such measurability assumptions for standard arguments (e.g., regading uniform convergence) to hold even in the non-private case: without such assumptions, there are (uncountably) infinite classes of VC dimension 1, which empirical risk minimization fails to learn [Ben15].)

For any $x \in \mathcal{X}, b \in \{-1, 1\}$, write $\mathcal{F}|_{(x,b)} := \{f \in \mathcal{F} : f(x) = b\}$.

2.2 Differential privacy and sanitization

While our main focus in this paper is on PAC learning, we will additionally discuss implications of our results to differentially private data release. Therefore, in the below definition of differential privacy, we allow each user's example to belong to an arbitrary set $\mathcal{Z}$ (in PAC learning we have $\mathcal{Z} = \mathcal{X} \times \{-1, 1\}$).

Definition 2.2 (Differential privacy, [Dwo06])**.** Fix sets $\mathcal{Z}, \mathcal{W}$ and $n \in \mathbb{N}$, and suppose $\mathcal{W}$ is countable.[3] A randomized algorithm $A : \mathcal{Z}^n \to \mathcal{W}$ is (ε, δ)-*differentially private* if the following holds: for any datasets $S, S' \in \mathcal{Z}^n$ differing in a single example and for all subsets $\mathcal{T} \subset \mathcal{W}$,

$$\Pr[A(S) \in \mathcal{T}] \leqslant e^{\varepsilon} \cdot \Pr[A(S') \in \mathcal{T}] + \delta.$$

The *sanitization* (or *private query release*) problem was introduced in [BLR08] and has been central in many works in differential privacy:

Definition 2.3 (Sanitization, [BLR08, BNS14])**.** Fix $n \in \mathbb{N}$ and $\alpha, \beta, \varepsilon, \delta \in (0,1)$, and suppose $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ is a binary hypothesis class. A randomized algorithm $A : (\mathcal{X} \times \{-1,1\})^n \to [0,1]^{\mathcal{F}}$ is an $(n, \alpha, \beta, \varepsilon, \delta)$-*sanitizer* if A is (ε, δ)-differentially private and for all datasets $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1,1\})^n$, $A(S)$ outputs a function $\mathrm{Est} : \mathcal{F} \to [0,1]$ so that with probability at least $1 - \beta$, for all $f \in \mathcal{F}$,

$$\left| \mathrm{Est}(f) - \frac{|\{i \in [n] : f(x_i) = y_i\}|}{n} \right| \leqslant \alpha.$$

Following [BLM20a], we say that a class $\mathcal{F}$ is *sanitizable* if there exists a bound $n_{\mathcal{F}}(\alpha, \beta) = \mathrm{poly}(1/\alpha, 1/\beta)$ so that for every $\alpha, \beta > 0$, there exists an algorithm A on datasets of size $n = n_{\mathcal{F}}(\alpha, \beta)$ which is an $(n, \alpha, \beta, \varepsilon, \delta)$-sanitizer for some $\varepsilon = O(1)$ and δ negligible as a function of n.

2.3 VC dimension and uniform convergence

We will denote *hypothesis classes*, namely subsets of $\{-1,1\}^{\mathcal{X}}$, with the letters $\mathcal{F}, \mathcal{G}, \mathcal{H}$. A class $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ is said to *shatter* a set $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ if for each choice $(\varepsilon_1, \ldots, \varepsilon_n) \in \{-1,1\}^n$, there is some $f \in \mathcal{F}$ so that for all $i \in [n]$, $f(x_i) = \varepsilon_i$.

Definition 2.4 (VC dimension)**.** The *VC dimension* of the class $\mathcal{F}$, denoted $\mathrm{VCdim}(\mathcal{F})$, is the largest positive integer n so that $\mathcal{F}$ shatters a set of size n.

We need the following standard fact that finite VC dimension is a sufficient condition for uniform convergence with respect to arbitrary distributions:

Theorem 2.1 (e.g., [BM03], Theorems 5 & 6)**.** *Suppose that $\mathcal{F}$ is countable and $\mathrm{VCdim}(\mathcal{F}) = d_{\mathrm{V}} \geqslant 1$. Then there is a constant $C_0 \geqslant 1$ such that for any distribution P on $\mathcal{X} \times \{-1,1\}$ and any $\gamma \in (0,1)$, it holds that*

$$\Pr_{S_n \sim P^n} \left[\sup_{f \in \mathcal{F}} \left| \mathrm{err}_P(f) - \mathrm{err}_{\hat{P}_{S_n}}(f) \right| > C_0 \sqrt{\frac{d_{\mathrm{V}} + \log 1/\gamma}{n}} \right] \leqslant \gamma.$$

For a class $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ and a distribution P on $\mathcal{X} \times \{-1,1\}$, define

$$\mathcal{F}_{P,\alpha} := \{f \in \mathcal{F} : \mathrm{err}_P(f) \leqslant \alpha\}.$$

Note that for any $0 \leqslant \alpha \leqslant \beta \leqslant 1$, we have $\mathcal{F}_{P,\alpha} \subset \mathcal{F}_{P,\beta}$.

For any $\gamma > 0$ and $n \in \mathbb{N}$, write $\alpha(n, \gamma) := C_0 \sqrt{\frac{d_{\mathrm{V}} + \log 1/\gamma}{n}}$, so that by Theorem 2.1 we have that $\Pr_{S_n} \left[\sup_{f \in \mathcal{F}} \left| \mathrm{err}_P(f) - \mathrm{err}_{\hat{P}_{S_n}}(f) \right| > \alpha(n, \gamma) \right] \leqslant \gamma$.

[3] The restriction of countability may be readily removed by fixing a σ-algebra Σ on $\mathcal{W}$ and letting A be a mapping from $\mathcal{Z}^n$ to the space $\Delta(\mathcal{W})$ of probability measures on the measure space $(\mathcal{W}, \Sigma)$.

Note that, under the event $\sup_{f\in\mathcal{F}}\left|\mathrm{err}_P(f)-\mathrm{err}_{\hat{P}_{S_n}}(f)\right|\leqslant\alpha_0$, we have that, for each $\alpha\in[0,1]$,

$$\mathcal{F}_{\hat{P}_{S_n},\alpha-2\alpha_0}\subset\mathcal{F}_{P,\alpha-\alpha_0}\subset\mathcal{F}_{\hat{P}_{S_n},\alpha}. \tag{1}$$

Given a class $\mathcal{F}\subset\{-1,1\}^{\mathcal{X}}$, its *dual class*, denoted by $\mathcal{F}^\star$, is defined as follows: $\mathcal{F}^\star\subset\{-1,1\}^{\mathcal{F}}$ and is indexed by $\mathcal{X}$. For each $x\in\mathcal{X}$, the corresponding function in $\mathcal{F}^\star$ is the function $x:\mathcal{F}\to\{-1,1\}$, defined by $x(f):=f(x)$. The *dual VC dimension* of $\mathcal{F}$, denoted by $\mathrm{VCdim}^\star(\mathcal{F})$, is the VC dimension of $\mathcal{F}^\star$: i.e., $\mathrm{VCdim}^\star(\mathcal{F}):=\mathrm{VCdim}(\mathcal{F}^\star)$.

2.4 Littlestone dimension

To introduce the Littlestone dimension, we need some notation regarding binary trees. For a positive integer t and a sequence $b_1,b_2,\ldots,b_t,\ldots\in\{-1,1\}$, write $b_{1:t}:=(b_1,\ldots,b_t)$. As a convention, let $b_{1:0}$ denote the empty sequence. For $n\in\mathbb{N}$, an *$\mathcal{X}$-valued binary tree* $\mathbf{x}$ of *depth* n is a collection of partial functions $\mathbf{x}_t:\{-1,1\}^{t-1}\to\mathcal{X}$ for $1\leqslant t\leqslant n$, each with nonempty domain, so that for all $b_{1:t}$ in the domain of $\mathbf{x}_{t+1}$, $b_{1:t-1}$ is in the domain of $\mathbf{x}_t$ and $(b_1,\ldots,b_{t-1},-b_t)$ is in the domain of $\mathbf{x}_{t+1}$. If $\mathbf{x}_t$ is a total function for all t, then we say that $\mathbf{x}$ is a *complete* tree; otherwise, we say that $\mathbf{x}$ is *incomplete*. By default we will use the term "tree" to refer to complete binary trees; when we wish to refer to incomplete trees (or the notion of *generalized trees* in Definition 4.4), we will use the appropriate adjective.

Associated with each sequence $b_{1:t}\in\{-1,1\}^t$ so that either $t=0$ or $b_{1:t-1}$ is in the domain of $\mathbf{x}_t$, for some $1\leqslant t\leqslant n$, is a *node* of the (possibly incomplete) tree. We say that this node is a *leaf* if $b_{1:t}$ is not in the domain of $\mathbf{x}_{t+1}$; in particular, for complete trees, the nodes associated to each $b_{1:n}\in\{-1,1\}^n$ are the leaves. Suppose $b_{1:t-1}$ is in the domain of $\mathbf{x}_t$, for some t; then the node associated with $b_{1:t-1}$ is not a leaf, and we say that this node is *labeled* by $\mathbf{x}_t(b_{1:t-1})$. For any such non-leaf node v, the two nodes associated with $(b_1,\ldots,b_{t-1},-1)$ and $(b_1,\ldots,b_{t-1},1)$ are the *children* of v *corresponding to the bits* -1 *and* 1, respectively. Note that a node is a leaf if and only if it has no children. Note also that any non-leaf node has exactly 2 children.

A class $\mathcal{F}\subset\{-1,1\}^{\mathcal{X}}$ is said to *shatter* a (complete) tree $\mathbf{x}$ of depth n if for all sequences $(b_1,\ldots,b_n)\in\{-1,1\}^n$, there is some $f\in\mathcal{F}$ so that for each $t\in[n]$, $f(\mathbf{x}_t(b_{1:t-1}))=b_t$.

Definition 2.5 (Littlestone dimension)**.** The *Littlestone dimension* of a class $\mathcal{F}\subset\{-1,1\}^{\mathcal{X}}$ is the largest positive integer n so that there exists a tree $\mathbf{x}$ of depth n that is shattered by $\mathcal{F}$.

The Littlestone dimension is known to exactly characterize the optimal mistake bound for online learnability of the class $\mathcal{F}$ in the realizable setting [Lit87], as well as to characterize the optimal regret bound for online learnability of $\mathcal{F}$ in the agnostic setting up to a logarithmic factor [BPS09].

Similar to the case for VC dimension, the dual Littlestone dimension of a class $\mathcal{F}$, denoted by $\mathrm{Ldim}^\star(\mathcal{F})$, is the Littlestone dimension of $\mathcal{F}^\star$: i.e., $\mathrm{Ldim}^\star(\mathcal{F}):=\mathrm{Ldim}(\mathcal{F}^\star)$.

3 Proof overview

In this section we overview the proof of Theorem 1.1. The proof is in two parts:

1. The first part is a private *improper* learner, `PolyPriLearn` (Algorithm 2), with sample complexity $\tilde{O}\left(\frac{d^6}{\varepsilon\alpha^2}\right)$. The hypothesis $\hat{f}\in\{-1,1\}^{\mathcal{X}}$ output by `PolyPriLearn` also satisfies an additional property, namely, it is associated with an *irreducible* subclass of $\mathcal{F}$ (a notion that we introduce and explain below), with high probability.

2. The second part is a technique, `PolyPriPropLearn` (Algorithm 3), to convert the improper learner from the first part to a proper learner using the irreducibility property of the hypothesis $\hat{f}$.

We now elaborate further on the two parts of the proof.

Part 1: Improper learner and irreducibility Besides allowing us to convert an improper learner to a proper one, the notion of *irreducibility* is central in allowing us to find a private improper learner for $\mathcal{F}$ with sample complexity polynomial in $\mathrm{Ldim}(\mathcal{F})$. Before defining irreducibility and explaining how it is useful, we first outline the overall approach. Given a dataset $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn i.i.d. from some distribution P over $\mathcal{X} \times \{-1, 1\}$, we will find several subclasses $\hat{\mathcal{G}}_1, \ldots, \hat{\mathcal{G}}_J \subset \mathcal{F}$,[4] for some $J \in \mathbb{N}$ so that for each $1 \leqslant j \leqslant J$, $\hat{\mathcal{G}}_j$ consists entirely of functions with low empirical error on the dataset S_n (this task is performed by the sub-routine `ReduceTree` (Algorithm 1) of `PolyPriLearn`). We will then consider the *SOA classifier*[5] for each subclass $\hat{\mathcal{G}}_j$; the SOA classifier for a class $\mathcal{G}$, denoted by $\mathrm{SOA}_{\mathcal{G}} \in \{-1, 1\}^{\mathcal{X}}$, is defined as follows: for $x \in \mathcal{X}$, $\mathrm{SOA}_{\mathcal{G}}(x) = 1$ if $\mathrm{Ldim}(\mathcal{G}|_{(x,1)}) \geqslant \mathrm{Ldim}(\mathcal{G}|_{(x,-1)})$, and $\mathrm{SOA}_{\mathcal{G}}(x) = -1$ otherwise. The crux of the proof rests on two facts:

(a) There are $d+1$ "special" classifiers $\sigma^\star_1, \ldots, \sigma^\star_{d+1} \in \{-1, 1\}^{\mathcal{X}}$ (which depend on P but not any particular dataset) so that with high probability, at least one of $\mathrm{SOA}_{\hat{\mathcal{G}}_1}, \ldots, \mathrm{SOA}_{\hat{\mathcal{G}}_J}$ is equal to one of $\sigma^\star_1, \ldots, \sigma^\star_{d+1}$.

(b) For each class $\hat{\mathcal{G}}_j$ that is found in the sub-routine `ReduceTree`, with high probability it holds that $\mathrm{SOA}_{\hat{\mathcal{G}}_j}$ has low population error (i.e., $\mathrm{err}_P(\mathrm{SOA}_{\hat{\mathcal{G}}_j})$ is small).

If properties (a) and (b) are given, then the construction of a private learner is fairly straightforward: if J were a constant, then we could draw $m = \tilde{O}(d)$ independent datasets $S_n^{(1)}, \ldots, S_n^{(m)}$ and use the private stable histogram of [BNS16, Proposition 2.20] together with property (a) to privately output some $\mathrm{SOA}_{\mathcal{G}} \in \{-1, 1\}^{\mathcal{X}}$ that belongs to $\{\mathrm{SOA}_{\hat{\mathcal{G}}_1^{(i)}}, \ldots, \mathrm{SOA}_{\hat{\mathcal{G}}_J^{(i)}}\}$ for many of the independent datasets $S_n^{(i)}$ (we denote the subclasses corresponding to the ith dataset, $i \in [m]$, by $\hat{\mathcal{G}}_j^{(i)}$, for $j \in [J]$). By property (b), such $\mathrm{SOA}_{\mathcal{G}}$ would then have low population error. As it turns out, we will only be able to guarantee that $J = 2^{\tilde{O}(d^2)}$; we can still guarantee sample complexity polynomial in d, though, by using a variant of the stable histogram based on the exponential mechanism [GKM20] in Algorithm 2. This will necessitate an increase in m by a factor of $\log(2^{\tilde{O}(d^2)})$, so that we draw a total of $m = \tilde{O}(d^3)$ independent datasets; each will be of size $\tilde{O}(d^3)$, leading to the overall sample complexity bound of $\tilde{O}(d^6)$.

Next we discuss the proofs of properties (a) and (b) of the subclasses $\hat{\mathcal{G}}_1, \ldots, \hat{\mathcal{G}}_J$ that the sub-routine `ReduceTree` outputs. The proofs of both of these properties depend on irreducibility, which we now define. We say that a hypothesis class $\mathcal{G} \subset \{-1, 1\}^{\mathcal{X}}$ is *irreducible* if for any $x \in \mathcal{X}$, it holds that for some $b \in \{-1, 1\}$, we have $\mathrm{Ldim}(\mathcal{F}|_{(x,b)}) = \mathrm{Ldim}(\mathcal{F})$. Definition 4.1 introduces the generalization of *k-irreducibility* for all $k \in \mathbb{N}$ (irreducibility corresponds to 1-irreducibility), but in this section we exhibit the main ideas behind the proof using $k = 1$. To explain how we obtain property (a), first suppose that the following holds, for some fixed $\alpha_\Delta < \alpha_0$:

$$\begin{gathered}\textit{With high probability over the sample } S_n\textit{, it holds that} \\ \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}) \textit{ and } \mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta} \textit{ is irreducible.}\end{gathered} \tag{A}$$

By Theorem 2.1 and (1) with $\alpha_0 = \alpha_\Delta/2$, as long as $n \geqslant \tilde{\Omega}\left(\frac{d}{\alpha_\Delta^2}\right)$, then with high probability we

[4] As a general convention we use a hat for quantities that depend on the dataset.

[5] As an aside, the SOA classifier achieves the optimal mistake bound in the realizable setting of online learning [Lit87, Sha12].

have $\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta} \subset \mathcal{F}_{P,\alpha-\alpha_\Delta/2} \subset \mathcal{F}_{\hat{P}_{S_n},\alpha}$, and so

$$\mathrm{Ldim}(\mathcal{F}_{P,\alpha-\alpha_\Delta/2}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}) \tag{2}$$

by (A). Using irreducibility of $\mathcal{F}_{P,\alpha-\alpha_\Delta}$ and (2), it is straightforward to show (Lemma 4.3) that

$$\mathrm{SOA}_{\mathcal{F}_{P,\alpha-\alpha_\Delta/2}} = \mathrm{SOA}_{\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}}. \tag{3}$$

Thus, we have shown, assuming (A), that a quantity that can be computed from the empirical data, namely $\mathrm{SOA}_{\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}}$, is equal with high probability to a fixed quantity, namely $\mathrm{SOA}_{\mathcal{F}_{P,\alpha-\alpha_\Delta/2}}$, which we may take to be, say $\sigma_1^\star$, in property (a).

Of course, we must also deal with the case where (A) does not hold. There are two possible reasons for this: the first is that $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}) < \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha})$. In this case, as long as α_Δ is sufficiently small, we may replace α with $\alpha - \alpha_\Delta$ and recurse (i.e., check if (A) holds with the new value of α, and act accordingly). Since $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha}) \leqslant \mathrm{Ldim}(\mathcal{F}) \leqslant d$, the Littlestone dimension can decrease at most d times and therefore it is sufficient to choose $\alpha_\Delta \approx \alpha/d$ (and so we may take $n = \tilde{O}(d^3)$).

The other reason that (A) may fail to hold is that $\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}$ is not irreducible. In such a case, by definition of irreducibility, there exists some $x \in \mathcal{X}$ so that

$$\max\{\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}|_{(x,1)}), \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha-\alpha_\Delta}|_{(x,-1)})\} < \mathrm{Ldim}(\mathcal{F}).$$

The idea is to now make *two* recursive calls, replacing α with $\alpha - \alpha_\Delta$ (as before) and using each of the classes $\mathcal{F}|_{(x,1)}$ and $\mathcal{F}|_{(x,-1)}$ in place of $\mathcal{F}$. A clear issue with this approach is that x may depend on the dataset S_n, and so the crucial "stability" property of (3) may fail to hold in the recursive call, even if (A) holds with the new α and for the class $\mathcal{F}|_{(x,\pm1)}$. It turns out that we can amend this issue by replacing irreducibility in (A) with the stronger property of k-irreducibility for $k > 1$; the details can be found in Sections 4 and 5.1.

This process of decreasing the Littlestone dimension by at least 1 and then making some number of "recursive" calls results in a tree with at most $2^{\tilde{O}(d^2)}$ leaves (Definition 4.4 describes the specific tree structure). Each of these leaves determines a class $\hat{\mathcal{G}}_j$, and using a generalization of (3), we can ensure that the classes $\hat{\mathcal{G}}_j$ satisfy property (a). Moreover, we will be able to ensure that for a sufficiently large integer k, each $\hat{\mathcal{G}}_j$ is k-irreducible; this will be enough to show that property (b) above holds via a fairly straightforward argument (carried out in Lemma 4.4 and Claim 5.9).

Part 2: Making the improper learner proper Let $\mathrm{SOA}_{\hat{\mathcal{G}}} \in \{-1,1\}^{\mathcal{X}}$ be the classifier output by the private improper learner `PolyPriLearn` described above. The idea to make this learner proper is to find a small set $\hat{\mathcal{H}} \subset \mathcal{F}$ (in particular, of size bounded by $O(\mathrm{VCdim}^\star(\mathcal{F})/\alpha^2)$), such that for any distribution Q over $\mathcal{X}$, there is some $\hat{h} \in \hat{\mathcal{H}}$ such that $\Pr_{x\sim Q}[\mathrm{SOA}_{\hat{\mathcal{G}}}(x) \neq \hat{h}(x)] \leqslant \alpha$. In particular, this holds for $Q = P$, the true population distribution. Thus, since the improper learner from above guarantees that $\mathrm{SOA}_{\hat{\mathcal{G}}}$ has low population error under P with high probability, we can choose some $\hat{h} \in \hat{\mathcal{H}}$ with not much higher error using the exponential mechanism on a fresh set of samples of size roughly $\log|\hat{\mathcal{H}}| \leqslant \tilde{O}(\log \mathrm{VCdim}^\star(\mathcal{F})) \leqslant \tilde{O}(\mathrm{VCdim}(\mathcal{F}))$ (this is explained in detail in `PolyPriPropLearn`, Algorithm 3).

It remains to show the existence of a small $\hat{\mathcal{H}} \subset \mathcal{F}$. To do so, we consider the zero-sum game with action spaces $\mathcal{F}$ and $\mathcal{X}$, where the row player chooses $h \in \mathcal{F}$, the column player chooses $x \in \mathcal{X}$,

and the value of the game is $\mathbb{1}[h(x) \neq \mathrm{SOA}_{\hat{\mathcal{G}}}(x)]$. By von Neumann's minimax theorem[6], we have

$$\inf_{D\in\Delta(\mathcal{F})} \sup_{P\in\Delta(\mathcal{X})} \mathbb{E}_{x\sim P,h\sim D}[\mathbb{1}[\mathrm{SOA}_{\hat{\mathcal{G}}}(x) \neq h(x)]] = \sup_{P\in\Delta(\mathcal{X})} \inf_{D\in\Delta(\mathcal{F})} \mathbb{E}_{x\sim P,h\sim D}[\mathbb{1}[\mathrm{SOA}_{\hat{\mathcal{G}}}(x) \neq h(x)]]. \tag{4}$$

Using the fact that the class $\hat{\mathcal{G}}$ corresponding to the classifier $\mathrm{SOA}_{\hat{\mathcal{G}}}$ output by `PolyPriLearn` is k-irreducible for a sufficiently large integer k, we show in Lemma 6.1 that the right-hand side of (4) is bounded above by the desired accuracy α. Thus the same holds for the left-hand side of (4). Now take a distribution $\hat{D} \in \Delta(\mathcal{F})$ attaining the infimum on the left-hand side of (4); using a uniform convergence argument applied to the *dual class* of $\mathcal{F}$ (Lemma 6.2), we may choose a multiset $\hat{\mathcal{H}} \subset \mathcal{F}$ of size $O(\mathrm{VCdim}^\star(\mathcal{F})/\alpha^2)$ so that the uniform distribution over $\hat{\mathcal{H}}$ comes close to the infimum on the left-hand side of (4). Such an $\hat{\mathcal{H}}$ satisfies the property we desired.

4 Irreducibility

In this section we make a definition which is central to our algorithm and its analysis, namely that of *irreducibility* of a hypothesis class. We then prove some basic properties of irreducible classes.

Fix some set $\mathcal{X}$ and a space $\mathcal{G}$ of hypotheses on $\mathcal{X}$. For any $x \in \mathcal{X}, b \in \{-1, 1\}$, set

$$\mathcal{G}|_{(x,b)} := \{f \in \mathcal{G} : f(x) = b\}.$$

For a set $S = \{(x_1, b_1), \ldots, (x_n, b_n)\}$, similarly set

$$\mathcal{G}|_S := \bigcap_{i\in[n]} \mathcal{G}|_{(x_i,b_i)} = \{f \in \mathcal{G} : f(x_i) = b_i \ \forall i \in [n]\}.$$

For S as above, we will at times abuse notation slightly and write $\mathcal{G}|_S = \mathcal{G}|_{(x_1,b_1),\ldots,(x_n,b_n)}$.

Definition 4.1 (Irreducibility)**.** A class $\mathcal{G}$ is defined to be *k-irreducible* if for any depth-k tree $\mathbf{x}$, there is some choice of bits $b_1, \ldots, b_k \in \{-1, 1\}$ such that

$$\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),(\mathbf{x}_2(b_1),b_2)\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)}) = \mathrm{Ldim}(\mathcal{G}).$$

We say that the class $\mathcal{G}$ is *irreducible* if it is 1-irreducible.

Note that k-irreducibility implies k'-irreducibility for $k' < k$. The following lemma shows that the choice of bits $b_1, \ldots, b_k$ in Definition 4.1 is unique:

Lemma 4.1. *Suppose $\mathcal{G}$ is k-irreducible. Then for any depth-k (possibly incomplete) tree $\mathbf{x}$, there is a unique $t \in [k]$ and leaf associated to some $(b_1, \ldots, b_t) \in \{-1, 1\}^t$ so that* $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),(\mathbf{x}_2(b_1),b_2)\ldots,(\mathbf{x}_t(b_{1:t-1}),b_t)}) = \mathrm{Ldim}(\mathcal{G})$.

Proof. Since $\mathcal{G}$ is k-irreducible, there is some $t \in [k]$ and leaf associated to some $b_{1:t} \in \{-1, 1\}^t$ so that $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),(\mathbf{x}_2(b_1),b_2)\ldots,(\mathbf{x}_t(b_{1:t-1}),b_t)}) = \mathrm{Ldim}(\mathcal{G})$. Suppose there were some other pair $(t', b'_{1:t'})$ so that $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b'_1),(\mathbf{x}_2(b'_1),b'_2)\ldots,(\mathbf{x}_t(b'_{1:t'-1}),b'_{t'})}) = \mathrm{Ldim}(\mathcal{G})$. Then $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1)}) = \mathrm{Ldim}(\mathcal{G}) = \mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b'_1)})$, and thus $b_1 = b'_1$. We proceed by induction: for $1 \leqslant s \leqslant \min\{t, t'\}$, if $b_1 = b'_1, \ldots, b_s = b'_s$, then

$$\begin{aligned}\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_{s+1}(b_{1:s}),b_{s+1})}) &= \mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_s(b_{1:s-1}),b_s)}) \\ &= \mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_s(b_{1:s-1}),b_s),(\mathbf{x}_{s+1}(b_{1:s}),b'_{s+1})}),\end{aligned}$$

[6]The application of von Neumann's minimax theorem assumes that $\mathcal{F}, \mathcal{X}$ are finite; the infinite (countable) case is handled in Appendix A.1 using basic ideas from topology.

and hence $b_{s+1} = b'_{s+1}$. Since $b_{1:t}$ and $b'_{1:t'}$ both are associated to leaves of the tree $\mathbf{x}$, we must have $t = t'$ and $b_{1:t} = b'_{1:t'}$. □

The following lemma shows that k-irreducibility satisfies a sort of "monotonicity" property among classes of the same Littlestone dimension.

Lemma 4.2. *Suppose $\mathcal{H} \subset \mathcal{G}$, and $\mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{G})$. If $\mathcal{H}$ is k-irreducible, then so is $\mathcal{G}$.*

Proof. If $\mathcal{H}$ is irreducible, then for any depth-k $\mathcal{X}$-valued tree $\mathbf{x}$, we have that for some $b_1, \ldots, b_k \in \{-1, 1\}$,

$$\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)}) \geqslant \mathrm{Ldim}(\mathcal{H}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)}) = \mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{G}),$$

where the first inequality follows since any $f \in \mathcal{H}$ with $f(\mathbf{x}_i(b_{1:i-1})) = b_i$ for $1 \leqslant i \leqslant k$ is also in $\mathcal{G}$. But since $\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)} \subset \mathcal{G}$, we have $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)}) \leqslant \mathrm{Ldim}(\mathcal{G})$, and so equality holds above, i.e., $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),\ldots,(\mathbf{x}_k(b_{1:k-1}),b_k)}) = \mathrm{Ldim}(\mathcal{G})$. □

We next define the *SOA classifier* associated with a function class $\mathcal{G}$; the choice of name is due to its similarity to the classifiers used in the standard optimal algorithm (SOA) in online learning [Lit87, BPS09].

Definition 4.2 (SOA classifier)**.** For a class $\mathcal{G}$, define the function $\mathrm{SOA}_{\mathcal{G}} : \mathcal{X} \to \{-1, 1\}$ as follows:

$$\mathrm{SOA}_{\mathcal{G}}(x) := \begin{cases} 1 & : \mathrm{Ldim}(\mathcal{G}|_{(x,1)}) \geqslant \mathrm{Ldim}(\mathcal{G}|_{(x,-1)}) \\ -1 & : \mathrm{Ldim}(\mathcal{G}|_{(x,1)}) < \mathrm{Ldim}(\mathcal{G}|_{(x,-1)}). \end{cases}$$

Lemma 4.3 establishes an important "stability-type" property satisfied by SOA classifiers of irreducible classes.

Lemma 4.3. *Suppose $\mathcal{H} \subset \mathcal{G}, \mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{G})$, and that $\mathcal{H}$ is irreducible. Then for all $x \in \mathcal{X}$, $\mathrm{SOA}_{\mathcal{H}}(x) = \mathrm{SOA}_{\mathcal{G}}(x)$.*

Proof. Fix any $x \in \mathcal{X}$. First suppose that $\mathrm{SOA}_{\mathcal{H}}(x) = 1$, i.e., $\mathrm{Ldim}(\mathcal{H}|_{(x,1)}) \geqslant \mathrm{Ldim}(\mathcal{H}|_{(x,-1)})$. Then since $\mathcal{H}$ is irreducible and $\mathcal{H} \subset \mathcal{G}$,

$$\mathrm{Ldim}(\mathcal{G}|_{(x,1)}) \geqslant \mathrm{Ldim}(\mathcal{H}|_{(x,1)}) = \mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{G}),$$

which means that $\mathrm{Ldim}(\mathcal{G}|_{(x,1)}) = \mathrm{Ldim}(\mathcal{G})$, and thus $\mathrm{SOA}_{\mathcal{G}}(x) = 1$.

Next suppose that $\mathrm{SOA}_{\mathcal{H}}(x) = -1$, i.e., $\mathrm{Ldim}(\mathcal{H}|_{(x,1)}) < \mathrm{Ldim}(\mathcal{H}|_{(x,-1)})$. Again using irreducibility of $\mathcal{G}$, we see that

$$\mathrm{Ldim}(\mathcal{G}|_{(x,-1)}) \geqslant \mathrm{Ldim}(\mathcal{H}|_{(x,-1)}) = \mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{G}),$$

which means that $\mathrm{Ldim}(\mathcal{G}|_{(x,-1)}) = \mathrm{Ldim}(\mathcal{G})$. We must have $\mathrm{Ldim}(\mathcal{G}|_{(x,1)}) \leqslant \mathrm{Ldim}(\mathcal{G}) - 1$, else it would be the case that $\mathrm{Ldim}(\mathcal{G}) \geqslant 1 + \mathrm{Ldim}(\mathcal{G}|_{(x,-1)})$. Hence $\mathrm{SOA}_{\mathcal{G}}(x) = -1$. □

The below lemma implies generalization bounds for the family of hypotheses $\mathrm{SOA}_{\mathcal{G}}$, for $\mathcal{G} \subset \mathcal{F}$ that are irreducible of sufficiently high order.

Lemma 4.4. *For a class $\mathcal{F}$ with $\mathrm{Ldim}(\mathcal{F}) = d$, set*

$$\tilde{\mathcal{F}}_{d+1} := \{\mathrm{SOA}_{\mathcal{G}} : \mathcal{G} \subset \mathcal{F}, \mathcal{G} \text{ is nonempty and } (d+1)\text{-irreducible.}\} \tag{5}$$

Then $\mathrm{Ldim}(\tilde{\mathcal{F}}_{d+1}) = d$ as well.

Note that $\mathcal{F} \subset \tilde{\mathcal{F}}_{d+1}$, since for any $f \in \mathcal{F}$, $\{f\}$ is k-irreducible for all $k \in \mathbb{N}$, and $\mathrm{SOA}_{\{f\}} = f$. It is natural to wonder whether one can upper-bound $\mathrm{Ldim}(\tilde{\mathcal{F}}_{d+1})$ if one drops the requirement that $\mathcal{G}$ is $(d+1)$-irreducible in (5); in Appendix B, we show that this is not possible.

Proof of Lemma 4.4. That $\mathrm{Ldim}(\tilde{\mathcal{F}}_{d+1}) \geqslant d$ follows from $\mathcal{F} \subset \tilde{\mathcal{F}}_{d+1}$ To see the upper bound on $\mathrm{Ldim}(\tilde{\mathcal{F}}_{d+1})$, suppose for the purpose of contradiction that $\tilde{\mathcal{F}}_{d+1}$ shatters an $\mathcal{X}$-valued tree $\mathbf{x}$ of depth $d+1$. We will show that $\mathcal{F}$ also shatters $\mathbf{x}$, which leads to the desired contradiction.

Fix any sequence $b = (b_1, \ldots, b_{d+1}) \in \{-1,1\}^{d+1}$. There must be some $\mathcal{G} \subset \mathcal{F}$ that is $(d+1)$-irreducible so that for $1 \leqslant t \leqslant d+1$, $\mathrm{SOA}_{\mathcal{G}}(\mathbf{x}_t(b_{1:t-1})) = b_t$, which, by irreducibility of $\mathcal{G}$, implies that $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_t(b_{1:t-1}),b_t)}) = \mathrm{Ldim}(\mathcal{G})$. Since $\mathcal{G}$ is $(d+1)$-irreducible, it follows that

$$\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b_1),(\mathbf{x}_2(b_1),b_2),\ldots,(\mathbf{x}_{d+1}(b_{1:d}),b_{d+1})}) = \mathrm{Ldim}(\mathcal{G}).$$

(Indeed, by $(d+1)$-irreducibility of $\mathcal{G}$, there must be *some* sequence $(b'_1, \ldots, b'_{d+1}) \in \{-1,1\}^{d+1}$ for which $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_1,b'_1),(\mathbf{x}_2(b'_1),b'_2),\ldots,(\mathbf{x}_{d+1}(b'_{1:d}),b'_{d+1})}) = \mathrm{Ldim}(\mathcal{G})$; the smallest t so that $b_t \neq b'_t$ satisfies $\mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_t(b_{1:t-1}),b_t)}) = \mathrm{Ldim}(\mathcal{G}|_{(\mathbf{x}_t(b_{1:t-1}),-b_t)}) = \mathrm{Ldim}(\mathcal{G})$, which is impossible. Thus $b_t = b'_t$ for all t.) Since $\mathcal{G}$ is nonempty, there must be some $f \in \mathcal{G} \subset \mathcal{F}$ such that for $1 \leqslant t \leqslant d+1$, $f(\mathbf{x}_t(b_{1:t-1})) = b_t$. It follows that $\mathcal{F}$ shatters the tree $\mathbf{x}$, as desired. □

In Definitions 4.3 and 4.4 below, we generalize the notion of tree to include those in which each node may have more than 2 children. The scheme by which we label nodes is somewhat non-standard so as to more closely correspond to the types of trees constructed in Algorithm 1 in the following section.

Definition 4.3 (Reducing arrays)**.** A *reducing array* of *depth* k is a collection of $k+1$ tuples $b^{(j)} := (b_1^{(j)}, \ldots, b_{j\wedge k}^{(j)}) \in \{-1,1\}^{j\wedge k}$ for $1 \leqslant j \leqslant k+1$, which satisfy the following property: $b_{j'}^{(j+1)} = b_{j'}^{(j)}$ for all $j' < j \leqslant k$, and $b_j^{(j+1)} = -b_j^{(j)}$ for $j \leqslant k$.[7]

Definition 4.4 (Generalized trees)**.** A *generalized tree* $\mathbf{x}$ with *values in* $\mathcal{X}$ of *depth* d and *branching factor* $k \in \mathbb{N}$ is a rooted tree of depth at most d[8] in which each node has at most $k+1$ children. Nodes of the tree without children are called its *leaves*. Moreover, the nodes and edges of the tree are labeled as follows:

1. Each non-leaf node v is labeled with an ordered tuple of some number $k_v \leqslant k$ of points in $\mathcal{X}$, denoted by $(\mathbf{x}(v)_1, \ldots, \mathbf{x}(v)_{k_v}) \in \mathcal{X}^{k_v}$.
2. The non-leaf node v has $k_v + 1$ children; the edge between v and the jth child, $1 \leqslant j \leqslant k_v + 1$, is labeled by a tuple $b^{(j)}$, where the tuples $b^{(j)} \in \{-1,1\}^{j\wedge k_v}$ form a reducing array of depth k_v (Definition 4.3).

Moreover, for any node v (perhaps a leaf), define $\mathbf{a}(v) \in (\mathcal{X} \times \{-1,1\})^\star$ (called the *ancestor set* of v) as follows: let $v^{(1)}, \ldots, v^{(t-1)}$ be the root-to-leaf path for the node v and $v^{(t)} := v$. For each $1 \leqslant i \leqslant t-1$, let $b^{(i)} \in \{-1,1\}^{k^{(i)}}$ be the label of the edge between $v^{(i)}$ and $v^{(i+1)}$, where $k^{(i)} \leqslant k_{v^{(i)}}$ is some positive integer. Then

$$\mathbf{a}(v) := \{(\mathbf{x}(v^{(1)})_1, b_1^{(1)}), \ldots, (\mathbf{x}(v^{(1)})_{k^{(1)}}, b_{k^{(1)}}^{(1)})\} \cup \cdots \cup \{(\mathbf{x}(v^{(t-1)})_1, b_1^{(t-1)}), \ldots, (\mathbf{x}(v^{(t-1)})_{k^{(t-1)}}, b_{k^{(t-1)}}^{(t-1)})\}.$$

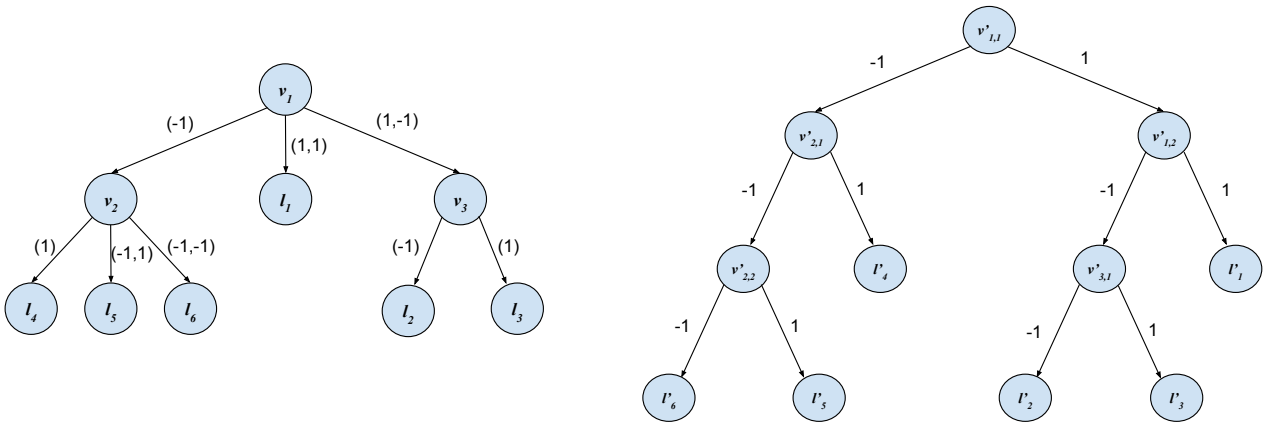
The *height* of the node v is defined to be $k^{(1)} + \cdots + k^{(t-1)}$, where $k^{(1)}, \ldots, k^{(t-1)}$ are defined given v as above. Note that the height of v is at least the size of (i.e., number of tuples in) the ancestor

[7] For real numbers a, b, we use the notation $a \wedge b$ and $a \vee b$ to denote $\min\{a,b\}$ and $\max\{a,b\}$ respectively.

[8] By depth at most d, we mean that the number of edges in the path from the root to any leaf is at most d; this aligns with the meaning of depth for binary trees in Section 2.4.

set $\mathbf{a}(v)$; the height may be even greater if there are duplicates in $\mathbf{a}(v)$. The *height* of the tree $\mathbf{x}$, denoted by $\mathrm{ht}(\mathbf{x})$, is the maximum height of any node v of $\mathbf{x}$. Note that we must have $\mathrm{ht}(\mathbf{x}) \geqslant d$ if the depth of $\mathbf{x}$ is d. To avoid ambiguity, when we wish to refer to a generalized tree, we will always use the adjective "generalized"; "tree" will continue to mean "complete binary tree".

Figure 1a shows an example of a generalized tree.



(a) Example of a generalized tree

(b) Transformed tree

Figure 1: (a) Example of a generalized tree $\mathbf{x}$ of depth 2 and height 3. The tree $\mathbf{x}$ has 3 non-leaf nodes, v_1, v_2, v_3, and 6 leaves, $\ell_1, \ldots, \ell_6$. We have $k_{v_1} = 2, k_{v_2} = 2, k_{v_3} = 1$. The tuples labeling the edges are the tuples of the reducing array corresponding to each non-leaf node v_i. For a few examples of ancestor sets, note that $\mathbf{a}(\ell_5) = \{(\mathbf{x}_1(v_1), -1), (\mathbf{x}_1(v_2), -1), (\mathbf{x}_2(v_2), 1)\}$, $\mathbf{a}(\ell_1) = \{(\mathbf{x}_1(v_1), 1), (\mathbf{x}_2(v_1), 1)\}$, and $\mathbf{a}(\ell_3) = \{(\mathbf{x}_1(v_1), 1), (\mathbf{x}_2(v_1), -1), (\mathbf{x}_1(v_3), 1)\}$. (b) The tree $\mathbf{x}'$ resulting from applying the transformation in Lemma 4.6 to $\mathbf{x}$. For each non-leaf node v_i, the k_{v_i} nodes constructed for v_i in $\mathbf{x}'$ are denoted by $v'_{i,1}, \ldots, v'_{i,k_{v_i}}$. For each leaf ℓ_i, the corresponding leaf in $\mathbf{x}'$ is denoted by ℓ'_i.

Lemma 4.5 below explains the choice of name "reducing array": it shows that if a class $\mathcal{G}$ is not k-irreducible, then there is a reducing array which can be used to "reduce the Littlestone dimension of $\mathcal{G}$" in a certain sense. As a matter of convention, if the class $\mathcal{G}$ is empty we write $\mathrm{Ldim}(\mathcal{G}) = -1$. Note also that $\mathrm{Ldim}(\mathcal{G}) = 0$ if and only if $\mathcal{G}$ contains a single hypothesis.

Lemma 4.5. *Suppose that $\mathcal{G}$ is not k-irreducible but is $(k-1)$-irreducible. Then there is a reducing array of depth k, denoted by $b^{(1)}, \ldots, b^{(k+1)}$ and a sequence $x_1, \ldots, x_k \in \mathcal{X}$ so that for all $j \in [k+1]$, we have*

$$0 \leqslant \mathrm{Ldim}(\mathcal{G}|_{(x_1, b_1^{(j)}), \ldots, (x_{j \wedge k}, b_{j \wedge k}^{(j)})}) < \mathrm{Ldim}(\mathcal{G}). \tag{6}$$

Proof. We use induction on k. For the base case $k = 1$, note that if $\mathcal{G}$ is not 1-irreducible, there must be some $x_1 \in \mathcal{X}$ so that $\max\{\mathrm{Ldim}(\mathcal{G}|_{(x,1)}), \mathrm{Ldim}(\mathcal{G}|_{(x,-1)})\} < \mathrm{Ldim}(\mathcal{G})$. Note that as a consequence we must have that $0 \leqslant \mathrm{Ldim}(\mathcal{G}|_{(x,b)})$ for each $b \in \{-1, 1\}$; indeed, if for some b it were the case that $\mathcal{G}|_{(x,b)}$ were empty, then $\mathcal{G} = \mathcal{G}|_{(x,-b)}$, in which case $\mathrm{Ldim}(\mathcal{G}) = \mathrm{Ldim}(\mathcal{G}|_{(x,-b)})$.

Now assume that the statement of the lemma is true for $k-1$. If, for each $x_1 \in \mathcal{X}$, there were some $b_1 \in \{-1, 1\}$ so that $\mathrm{Ldim}(\mathcal{G}|_{(x_1,b_1)}) = \mathrm{Ldim}(\mathcal{G})$ and $\mathcal{G}|_{(x_1,b_1)}$ were $(k-1)$-irreducible, then we would have that $\mathcal{G}$ is k-irreducible, a contradiction. Thus, there is some $x_1 \in \mathcal{X}$ so that one of the two conditions below holds:

- $0 \leqslant \mathrm{Ldim}(\mathcal{G}|_{(x_1,1)}) \vee \mathrm{Ldim}(\mathcal{G}|_{(x_1,-1)}) < \mathrm{Ldim}(\mathcal{G})$. In this case we may choose $x_2 = \cdots = x_k = x_1$ and the unique reducing array $b^{(1)}, \ldots, b^{(k+1)}$ satisfying $b_1^{(1)} = b_2^{(2)} = \cdots = b_k^{(k)} = 1, b_k^{(k+1)} = -1$ and obtain that (6) holds.

- For some $b_1 \in \{-1,1\}$, $\mathrm{Ldim}(\mathcal{G}|_{(x_1,b_1)}) = \mathrm{Ldim}(\mathcal{G})$ but $\mathcal{G}|_{(x_1,b_1)}$ is not $(k-1)$-irreducible (and is $(k-2)$-irreducible). In this case we must have that $\mathrm{Ldim}(\mathcal{G}|_{(x_1,-b_1)}) \geqslant 0$, i.e., $\mathcal{G}|_{(x_1,-b_1)}$ is nonempty; otherwise we would have that $\mathcal{G} = \mathcal{G}|_{(x_1,b_1)}$, which contradicts the fact that $\mathcal{G}$ is $(k-1)$-irreducible. Now we apply the inductive hypothesis, which guarantees a sequence $\tilde{x}_2, \ldots, \tilde{x}_k \in \mathcal{X}$ together with a reducing array of depth $k-1$, denoted $\tilde{b}^{(2)}, \ldots, \tilde{b}^{(k+1)}$, so that for $2 \leqslant j \leqslant k+1$, we have

$$0 \leqslant \mathrm{Ldim}(\mathcal{G}|_{(x_1,b_1),(\tilde{x}_2,b_1^{(j)}),\ldots,(\tilde{x}_{j\wedge k},b_{j\wedge k}^{(j)})}) < \mathrm{Ldim}(\mathcal{G}|_{(x_1,b_1)}) = \mathrm{Ldim}(\mathcal{G}). \tag{7}$$

  Now set $x_2 := \tilde{x}_2, \ldots, x_k := \tilde{x}_k$, and define the reducing array $b^{(1)}, \ldots, b^{(k+1)}$ of depth k by $b^{(1)} := -b_1$, and for $2 \leqslant j \leqslant k+1$, $b^{(j)} := (b_1, \tilde{b}_2^{(j)}, \ldots, \tilde{b}_{j\wedge k}^{(j)})$. Now $0 \leqslant \mathrm{Ldim}(\mathcal{G}|_{(x_1,-b_1)}) < \mathrm{Ldim}(\mathcal{G})$ together with (7) establishes (6). □

The next lemma extends Lemma 4.1 to generalized trees:

Lemma 4.6. *Suppose that* $\mathbf{x}$ *is a generalized tree so that* $\mathrm{ht}(\mathbf{x}) \leqslant k$*, and that* $\mathcal{G}$ *is* k*-irreducible. Then* $\mathbf{x}$ *has a unique leaf* ℓ *so that* $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\ell)}) = \mathrm{Ldim}(\mathcal{G})$*.*

Proof. We define a (possibly incomplete) binary tree $\mathbf{x}'$ of depth $\mathrm{ht}(\mathbf{x})$, as follows: for each non-leaf node v of $\mathbf{x}$ whose corresponding reducing array is denoted $b^{(1)}, \ldots, b^{(k_v+1)}$, we create k_v nodes of $\mathbf{x}'$, labeled by $\mathbf{x}(v)_1, \ldots, \mathbf{x}(v)_{k_v}$; we will refer to these nodes by $v'_1, \ldots, v'_{k_v}$. For $1 \leqslant k < k_v$, the node v'_{k+1} is a child of v'_k, corresponding to the bit $-b_k^{(k)}$. For $1 \leqslant k \leqslant k_v$, the child of v'_k corresponding to the bit $b_k^{(k)}$ is the child of v in $\mathbf{x}$ labeled by the tuple $b^{(k)}$. Finally, the child of v'_{k_v} corresponding to the bit $-b_{k_v}^{(k_v)}$ is the child of v in $\mathbf{x}$ labeled by the tuple $b^{(k_v+1)}$. An example of the construction of the tree $\mathbf{x}'$ is shown in Figure 1b.

It is evident that this construction induces a one-to-one mapping between leaves ℓ of $\mathbf{x}$ and corresponding leaves ℓ' of $\mathbf{x}'$. For any leaf ℓ of $\mathbf{x}$, notice that its ancestor set $\mathbf{a}(\ell)$ is equal to the ancestor set of the corresponding leaf ℓ' of $\mathbf{x}'$.[9] Lemma 4.1 implies that there is a unique leaf ℓ' of $\mathbf{x}'$ so that $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\ell')}) = \mathrm{Ldim}(\mathcal{G})$. Thus there is a unique leaf ℓ of $\mathbf{x}$ so that $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\ell)}) = \mathrm{Ldim}(\mathcal{G})$. □

Lemma 4.7 is a key part of the proof that the `ReduceTree` algorithm presented in Section 5.1 can be used together with the sparse selection protocol of Section 5.2 to generate an (improper) private learner. Roughly speaking, it gives sufficient conditions for a generalized tree $\mathbf{x}$ (which will depend on the input dataset) to have some leaf $\hat{v}$ so that for any hypothesis class $\mathcal{J}$ in a certain family of hypothesis classes, it holds that $\mathrm{SOA}_{\mathcal{J}|_{\mathbf{a}(\hat{v})}} = \mathrm{SOA}_{\mathcal{J}|_{S^\star}}$, where $S^\star \in (\mathcal{X} \times \{-1,1\})^\star$ is a collection of (x,y) pairs which will *not* depend on the input dataset. The statement of Lemma 4.7 is in fact slightly more general (so that the preceding statement corresponds to the case $\mathcal{J} = \mathcal{J}'$ in Lemma 4.7).

Lemma 4.7. *Fix some* $k, k' \in \mathbb{N}$ *with* $k > k'$ *and hypothesis classes* $\mathcal{H} \subset \mathcal{G} \subset \{-1,1\}^{\mathcal{X}}$*. Suppose we are given* $S^\star \in (\mathcal{X} \times \{-1,1\})^{k-k'}$ *so that* $\mathcal{H}|_{S^\star}$ *is* k*-irreducible, and that*

$$\mathrm{Ldim}(\mathcal{G}|_{S^\star}) = \mathrm{Ldim}(\mathcal{H}|_{S^\star}) =: \ell^\star \geqslant 0. \tag{8}$$

Suppose that $\mathbf{x}$ *is a generalized tree so that* $\mathrm{ht}(\mathbf{x}) \leqslant k-k'$ *and for all leaves* v *of* $\mathbf{x}$*,* $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(v)}) \leqslant \ell^\star$*. Then there is some leaf* $\hat{v}$ *of* $\mathbf{x}$ *so that* $\mathrm{SOA}_{\mathcal{J}|_{S^\star}} = \mathrm{SOA}_{\mathcal{J}'|_{\mathbf{a}(\hat{v})}}$ *for all hypothesis classes* $\mathcal{J}', \mathcal{J}$ *satisfying* $\mathcal{H} \subset \mathcal{J}' \subset \mathcal{G}$ *and* $\mathcal{H} \subset \mathcal{J} \subset \mathcal{G}$*.*

Moreover, the leaf $\hat{v}$ *satisfies:*

[9] Since incomplete binary trees are a special case of generalized trees, the definition of ancestor set in Definition 4.4 applies to the tree $\mathbf{x}'$.

1. $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\hat{v})}) = \mathrm{Ldim}(\mathcal{H}|_{\mathbf{a}(\hat{v})}) = \ell^\star$.
2. $\mathcal{H}|_{\mathbf{a}(\hat{v})}$ *is* k'*-irreducible.*

Proof. The k-irreducibility of $\mathcal{H}|_{S^\star}$, (8), and Lemma 4.2 gives that $\mathcal{G}|_{S^\star}$ and $\mathcal{J}|_{S^\star}$ are also k-irreducible for any $\mathcal{J} \supset \mathcal{H}$.

As a consequence of the k-irreducibility of $\mathcal{H}|_{S^\star}$ and the fact that $\mathrm{ht}(\mathbf{x}) \leqslant k - k'$, the following holds: there is some leaf $\hat{v}$ of $\mathbf{x}$ so that for any $\mathcal{X}$-valued tree $\mathbf{y}$ of depth k', there are some $(b_1, \ldots, b_{k'}) \in \{-1, 1\}^{k'}$ such that

$$\mathrm{Ldim}(\mathcal{H}|_{S^\star \cup \mathbf{a}(\hat{v}) \cup \{(\mathbf{y}_1, b_1), \ldots, (\mathbf{y}_{k'}(b_{1:k'-1}), b_{k'})\}}) = \mathrm{Ldim}(\mathcal{H}|_{S^\star}). \tag{9}$$

(That such a $\hat{v}$ exists is an immediate consequence of Lemma 4.6; that $\hat{v}$ does not depend on $\mathbf{y}$ follows from the fact that the leaf guaranteed by Lemma 4.6 is unique.)

Using the assumption that $\mathrm{Ldim}(\mathcal{H}|_{S^\star}) = \ell^\star \geqslant \mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\hat{v})})$, we see that for any $\mathbf{y}$ as above, there exist $b_1, \ldots, b_{k'}$ so that

$$\begin{aligned} \mathrm{Ldim}(\mathcal{H}|_{\mathbf{a}(\hat{v}) \cup \{(\mathbf{y}_1, b_1), \ldots, (\mathbf{y}_{k'}(b_{1:k'-1}), b_{k'})\}}) &\geqslant \mathrm{Ldim}(\mathcal{H}|_{S^\star \cup \mathbf{a}(\hat{v}) \cup \{(\mathbf{y}_1, b_1), \ldots, (\mathbf{y}_{k'}(b_{1:k'-1}), b_{k'})\}}) && (10) \\ &\overset{(9)}{=} \mathrm{Ldim}(\mathcal{H}|_{S^\star}) \\ &\geqslant \mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\hat{v})}) \geqslant \mathrm{Ldim}(\mathcal{H}|_{\mathbf{a}(\hat{v})}). && (11) \end{aligned}$$

It then follows that the inequalities in (10) and (11) are in fact equalities. For any $x \in \mathcal{X}$, set $\mathbf{y}$ to be the tree all of whose nodes are labeled by x. Then the tuple $(b_1, \ldots, b_{k'})$ making (9) true (which must be unique) is of the form $(b(x), \ldots, b(x))$ for some $b(x) \in \{-1, 1\}$. It follows from (10) and (11) that $\mathrm{SOA}_{\mathcal{H}|_{\mathbf{a}(\hat{v})}}(x) = b(x)$.

From (9) (again with all nodes of the tree $\mathbf{y}$ labeled by x) we see also that for any $x \in \mathcal{X}$,

$$\mathrm{Ldim}(\mathcal{H}|_{S^\star}) = \mathrm{Ldim}(\mathcal{H}|_{S^\star \cup \{(x, b(x))\}}), \tag{12}$$

which implies that $\mathrm{SOA}_{\mathcal{H}|_{S^\star}}(x) = b(x)$ for all $x \in \mathcal{X}$. By irreducibility of $\mathcal{H}|_{S^\star}$ and Lemma 4.3, we have that, for all $x \in \mathcal{X}$ and $\mathcal{J}$ satisfying $\mathcal{H} \subset \mathcal{J} \subset \mathcal{G}$,

$$\mathrm{SOA}_{\mathcal{H}|_{S^\star}}(x) = \mathrm{SOA}_{\mathcal{J}|_{S^\star}}(x). \tag{13}$$

Hence, for all $x \in \mathcal{X}$, $\mathrm{SOA}_{\mathcal{J}|_{S^\star}}(x) = b(x) = \mathrm{SOA}_{\mathcal{H}|_{\mathbf{a}(\hat{v})}}(x)$, which establishes the desired equality of SOA hypotheses for $\mathcal{J}' = \mathcal{H}$. Before establishing this for all $\mathcal{J}'$ satisfying $\mathcal{H} \subset \mathcal{J}' \subset \mathcal{G}$, we first show items 1 and 2.

Using the equalities of (10) and (11) gives that $\mathrm{Ldim}(\mathcal{G}|_{\mathbf{a}(\hat{v})}) = \mathrm{Ldim}(\mathcal{H}|_{\mathbf{a}(\hat{v})}) = \ell^\star$, establishing item 1. Item 2 is a direct consequence of the equalities of (10) and (11), since the depth-k' tree $\mathbf{y}$ is arbitrary.

Items 1 and 2 together with Lemma 4.3 imply that for any hypothesis class $\mathcal{J}'$ satisfying $\mathcal{H} \subset \mathcal{J}' \subset \mathcal{G}$, we have that

$$\mathrm{SOA}_{\mathcal{J}'|_{\mathbf{a}(v)}} = \mathrm{SOA}_{\mathcal{H}|_{\mathbf{a}(v)}} = \mathrm{SOA}_{\mathcal{J}|_{S^\star}},$$

as desired. □

5 Improper private learner for Littlestone classes

In this section we establish a variant of Theorem 1.1 where the learner is only guaranteed to be improper (namely, part 1 of the proof as described in Section 3). In Section 5.1 we introduce the

algorithm `ReduceTree` (defined in Algorithm 1), which, given a dataset $S_n = ((x_1, y_1), \dots, (x_n, y_n))$ outputs a set $\hat{\mathcal{S}}$ of hypotheses of the form $\mathrm{SOA}_{\mathcal{G}}$ for various classes $\mathcal{G} \subset \mathcal{F}$. (To aid our notation in the proofs we also have `ReduceTree` output a generalized tree $\hat{\mathbf{x}}$ and a set $\hat{\mathcal{L}}'$ of leaves of $\hat{\mathbf{x}}$.) In Section 5.2 we state guarantees for a private sparse selection protocol from [GKM20]. In Section 5.3 we will show how to use certain "stability-type" properties of the set $\hat{\mathcal{S}}$ together with the private sparse selection procedure from Section 5.2 to privately output a hypothesis which has low population error, which will establish the desired improper learner. (We will then make it proper in Section 6, thus establishing Theorem 1.1 in its entirety.)

5.1 Building block: `ReduceTree` algorithm

Throughout this section we fix a function class $\mathcal{F}$ and write $d := \mathrm{Ldim}(\mathcal{F})$; we also assume that a given distribution P over $\mathcal{X} \times \{-1, 1\}$ is realizable for $\mathcal{F}$. The algorithm `ReduceTree` takes some $\gamma \in [0, 1], n, k' \in \mathbb{N}$ as parameters (to be specified below). It also takes as input some dataset $S_n \in (\mathcal{X} \times \{-1, 1\})^n$, which is accessed via the empirical distribution $\hat{P}_{S_n}$. Recall the function $\alpha(n, \gamma) \in [0, 1]$ defined after Theorem 2.1. Let $\alpha_\Delta := 6 \cdot \alpha(n, \gamma)$, and define E_{good} to be the event

$$E_{\mathrm{good}} := \left\{ \sup_{f \in \mathcal{F}} \left| \mathrm{err}_P(f) - \mathrm{err}_{\hat{P}_{S_n}}(f) \right| \leqslant \frac{\alpha_\Delta}{6} \right\}. \tag{14}$$

Assuming that the dataset $S_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$ is distributed i.i.d. according to P, by Theorem 2.1, $\Pr_{S_n \sim P^n}[E_{\mathrm{good}}] \geqslant 1 - \gamma$.

The algorithm `ReduceTree` operates as follows. It starts with the class $\mathcal{F}_{\hat{P}_{S_n}, \alpha_0}$ of hypotheses with empirical error at most α_0 (α_0 will be chosen so that it is less than the desired error for the output of the private learner). If it is the case that $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta})$ and $\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}$ is irreducible for some appropriately chosen $\alpha_\Delta > 0$, then by Lemmas 4.2 and 4.3, under the event E_{good}, the classifier $\mathrm{SOA}_{\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - 2\alpha_\Delta/3}}$ is "stable" in the sense that it does not "depend much" on the dataset S_n. (We leave a formalization of this statement to the proof below.) In this case we can output the set consisting of the single hypothesis, $\hat{\mathcal{S}} := \{\mathrm{SOA}_{\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - 2\alpha_\Delta/3}}\}$.

If we did not terminate in the above paragraph, then one of the following two statements must hold: (1) it holds that $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}) < \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0})$, or (2) $\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}$ is not irreducible and $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0})$. If (1) holds, then we may simply recurse, i.e., repeat the above process with $\alpha_1 := \alpha_0 - \alpha_\Delta$ replacing α_0. Otherwise, (2) holds, so (by Lemma 4.5 with $k = 1$) we can choose some $x \in \mathcal{X}$ so that $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}|_{(x,b)}) < \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0 - \alpha_\Delta}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_0})$ for each $b \in \{-1, 1\}$. In this case, we can repeat the above process twice (once for each $b \in \{-1, 1\}$), with $\alpha_1 := \alpha_0 - \alpha_\Delta$ replacing α_0 and with $\mathcal{F}|_{(x,b)}$ replacing $\mathcal{F}$ for each $b \in \{-1, 1\}$. The point $x \in \mathcal{X}$ becomes the label of the root node of a generalized tree, with two children (which are leaves), corresponding to the bits ± 1. Each step $t \geqslant 1$ of the above-described recursion builds upon this generalized tree maintained by the algorithm by adding children to some of its current leaves. For technical reasons, at depth t of this recursion, we will need to replace the requirement of "irreducibility" with that of "$k' \cdot 2^t$-irreducibility", for some k' which does not depend on t. The algorithm is guaranteed to terminate because with each increase in t, the Littlestone dimension of the current class under consideration decreases, and it can only do so at most $\mathrm{Ldim}(\mathcal{F})$ times. Further details may be found in Algorithm 1.

We say that the dataset S_n is *realizable* if there is some $f \in \mathcal{F}$ so that $\mathrm{err}_{\hat{P}_{S_n}}(f) = 0$ (this is the case with probability 1 if $S_n \sim P^n$ and P is realizable). Lemma 5.1 states a basic property of the output set $\hat{\mathcal{L}}'$ of `ReduceTree`.

Algorithm 1: `ReduceTree`

Input: Parameters $n, k' \in \mathbb{N}, \gamma \in [0,1]$, $\alpha_\Delta := 6 \cdot \alpha(n,\gamma)$. Distribution $\hat{P}_{S_n}$ over $\mathcal{X}$. Hypothesis class $\mathcal{F}$, with $d := \mathrm{Ldim}(\mathcal{F})$.

1. Initialize a counter $t = 1$ (t counts the depth of the generalized tree constructed at each step of the algorithm).
2. For $1 \leqslant t \leqslant d+1$, set $\alpha_t := (d+3-t) \cdot \alpha_\Delta$.
3. For $1 \leqslant t \leqslant d$, set $k_t := k' \cdot 2^t$.
4. Initialize $\hat{\mathbf{x}}^{(0)} = \{v_0\}$ to be a tree with a single (unlabeled) leaf v_0. (In general $\hat{\mathbf{x}}^{(t)}$ will be the generalized tree produced by the algorithm after step t is completed.)
5. Initialize $\hat{\mathcal{L}}_1 = \{v_0\}$. (In general $\hat{\mathcal{L}}_t$ will be the set of leaves of the tree before step t is started.)
6. For $t \in \{1, 2, \ldots, d\}$:
   (a) For each leaf $v \in \hat{\mathcal{L}}_t$ and $\alpha \in [0,1]$, set $\hat{\mathcal{G}}(\alpha, v) := \mathcal{F}_{\hat{P}_{S_n},\alpha}|_{\mathbf{a}(v)}$. (Note that since the only way the tree changes from round to round is by adding children to existing nodes, $\mathbf{a}(v)$ will never change for a node v that already exists.)
   (b) Let $\hat{w}_t^\star := \max_{v \in \hat{\mathcal{L}}_t} \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t, v))$ be the maximum Littlestone dimension of any of the classes $\hat{\mathcal{G}}(\alpha_t, v)$.
   Also let $\hat{\mathcal{L}}'_t := \{v \in \hat{\mathcal{L}}_t : \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t, v)) = \hat{w}_t^\star\}$.
   (c) If there is some $v \in \hat{\mathcal{L}}'_t$ so that $\mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) = \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t, v))$ and $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is k_t-irreducible, then break out of the loop and go to step 7.
   (d) Else, for each node $v \in \hat{\mathcal{L}}'_t$:
      i. If $\mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) < \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t, v))$, move on to the next v.
      ii. Else, we must have that $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is not k_t-irreducible. Let k_v be chosen as small as possible so that $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is not k_v-irreducible; then $k_v \leqslant k_t$. By Lemma 4.5, there is some sequence $x_1, \ldots, x_{k_v} \in \mathcal{X}$ and reducing array $b^{(1)}, \ldots, b^{(k_v+1)}$ of depth k_v so that for $1 \leqslant j \leqslant k_v + 1$, it holds that

$$0 \leqslant \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)|_{(x_1, b_1^{(j)}), \ldots, (x_{j \wedge k_v}, b_{j \wedge k_v}^{(j)})}) < \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)). \tag{15}$$

      iii. Give v the label $(x_1, \ldots, x_{k_v})$. Construct $k_v + 1$ children of v (all leaves of the current tree), with edge labels given by $b^{(1)}, \ldots, b^{(k_v+1)}$.
   (e) Let the current tree (with the additions of the previous step) be denoted by $\hat{\mathbf{x}}^{(t)}$, and let $\hat{\mathcal{L}}_{t+1}$ be the list of the leaves of $\hat{\mathbf{x}}^{(t)}$, i.e., the nodes which have not (yet) been assigned labels or children.
7. Let t_{final} be the final value of t the algorithm *completed* the loop of step 6d for before breaking out of the above loop (i.e., if the break at step 6c was taken at step t, then $t_{\mathrm{final}} = t - 1$; if the break was never taken, then $t_{\mathrm{final}} = d$). Let $\hat{w}^\star_{t_{\mathrm{final}}+1}$ and $\hat{\mathcal{L}}'_{t_{\mathrm{final}}+1}$ be defined as in Step 6b.
8. Output the set $\hat{\mathcal{L}}' := \hat{\mathcal{L}}'_{t_{\mathrm{final}}+1}$ of leaves of the tree $\hat{\mathbf{x}}^{(t_{\mathrm{final}})}$, and the tree $\hat{\mathbf{x}} := \hat{\mathbf{x}}^{(t_{\mathrm{final}})}$. Finally, output the set

$$\hat{\mathcal{S}} := \{\mathrm{SOA}_{\hat{\mathcal{G}}(\alpha_{t_{\mathrm{final}}+1} - 2\alpha_\Delta/3, v)} : v \in \hat{\mathcal{L}}' \text{ and } \hat{\mathcal{G}}(\alpha_{t_{\mathrm{final}}+1} - 2\alpha_\Delta/3, v) \text{ is } k'\text{-irreducible \& nonempty}\}. \tag{16}$$

Lemma 5.1. *Suppose the input dataset S_n of* `ReduceTree` *is realizable. The set $\hat{\mathcal{L}}'$ output by* `ReduceTree` *satisfies the following property: letting $t = t_{\text{final}} + 1 \in [d+1]$, there is some leaf $v \in \hat{\mathcal{L}}'$ so that $\text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) = \text{Ldim}(\hat{\mathcal{G}}(\alpha_t, v)) \geqslant 0$ and $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is k_t-irreducible.*

Proof. If for some t, the algorithm breaks at step 6c, then the conclusion of the lemma is immediate: indeed, the condition to break in step 6c gives that for some $v \in \hat{\mathcal{L}}'_t = \hat{\mathcal{L}}'_{t_{\text{final}}+1}$ we have $\text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) = \text{Ldim}(\hat{\mathcal{G}}(\alpha_t, v))$ and $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is k_t-irreducible. In light of (17) below, since v maximizes $\text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v))$ among $v \in \hat{\mathcal{L}}_t$, it follows that $\text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) \geqslant 0$.

Otherwise, the algorithm performs a total of d iterations. We claim that $\hat{w}^\star_{d+1} = 0$. We first show that for all $t \geqslant 1$, $\hat{w}^\star_{t+1} < \hat{w}^\star_t$. To see this fact, note that each leaf v in $\hat{\mathcal{L}}_{t+1}$ belongs to one of the following three categories:

- $v \in \hat{\mathcal{L}}_t \backslash \hat{\mathcal{L}}'_t$. In this case, we have
$$\text{Ldim}(\hat{\mathcal{G}}(\alpha_{t+1}, v)) \leqslant \text{Ldim}(\hat{\mathcal{G}}(\alpha_t, v)) < \hat{w}^\star_t.$$
- $v \in \hat{\mathcal{L}}'_t$ and $\text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) < \text{Ldim}(\hat{\mathcal{G}}(\alpha_t, v))$. Using that $\alpha_{t+1} = \alpha_t - \alpha_\Delta$, we obtain
$$\text{Ldim}(\hat{\mathcal{G}}(\alpha_{t+1}, v)) = \text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) < \text{Ldim}(\hat{\mathcal{G}}(\alpha_t, v)) \leqslant \hat{w}^\star_t.$$
- v is the jth child of some node $u \in \hat{\mathcal{L}}'_t$ for some $1 \leqslant j \leqslant k_u + 1 \leqslant k_t + 1$, as constructed in step 6(d)ii of the algorithm. Let the label of the edge between u and v be denoted by $b^{(j)} \in \{-1, 1\}^{j \wedge k_u}$. Then since $\alpha_{t+1} = \alpha_t - \alpha_\Delta$,
$$\begin{aligned}
\text{Ldim}(\hat{\mathcal{G}}(\alpha_{t+1}, v)) &\leqslant \text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) \\
&= \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n}, \alpha_t - \alpha_\Delta}|_{\mathbf{a}(u) \cup \{(\mathbf{x}^{(t)}(u)_1, b^{(j)}_1), \ldots, (\mathbf{x}^{(t)}(u)_{j \wedge k_u}, b^{(j)}_{j \wedge k_u})\}}) \\
&< \text{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, u)) \\
&\leqslant \hat{w}^\star_t,
\end{aligned}$$
where the strict inequality follows from (15).

Since $\hat{w}^\star_1 \leqslant d$ as $\hat{\mathcal{G}}(\alpha_1, v) \subset \mathcal{F}$, we obtain that $\hat{w}^\star_{d+1} \leqslant 0$. Thus all leaves v in $\hat{\mathcal{L}}_{t+1}$ satisfy $\text{Ldim}(\hat{\mathcal{G}}(\alpha_{t+1}, v)) < \hat{w}^\star_t$, i.e., $\hat{w}^\star_{t+1} < \hat{w}^\star_t$. Thus each $v \in \hat{\mathcal{L}}_{d+1}$ satisfies $\text{Ldim}(\hat{\mathcal{G}}(\alpha_{d+1}, v)) \leqslant 0$, and $\hat{\mathcal{L}}'_{d+1} = \hat{\mathcal{L}}'$ is exactly the set of $v \in \hat{\mathcal{L}}_{d+1}$ for which $\hat{\mathcal{G}}(\alpha_{d+1}, v)$ is nonempty. Hence
$$\mathcal{F}_{\hat{P}_{S_n}, \alpha} = \bigcup_{v \in \hat{\mathcal{L}}} \mathcal{F}_{\hat{P}_{S_n}, \alpha}|_{\mathbf{a}(v)} = \bigcup_{v \in \hat{\mathcal{L}}} \hat{\mathcal{G}}(\alpha, v) \tag{17}$$
for all $\alpha \leqslant \alpha_{d+1}$. Since we assume S_n is realizable, it follows that $\mathcal{F}_{\hat{P}_{S_n}, \alpha}$ is nonempty and thus $\max_{v \in \hat{\mathcal{L}}'}\{\text{Ldim}(\hat{\mathcal{G}}(\alpha, v))\} \geqslant 0$ for $\alpha \geqslant 0$. Since $\alpha_{d+1} - \alpha_\Delta \geqslant 0$, it follows that $\hat{w}^\star_{d+1} = 0$ and also that there is some $v \in \hat{\mathcal{L}}'$ so that $\text{Ldim}(\hat{\mathcal{G}}(\alpha_{d+1} - \alpha_\Delta, v)) \geqslant 0$. Since $\hat{w}^\star_{d+1} = 0$, we have $\text{Ldim}(\hat{\mathcal{G}}(\alpha_{d+1}, v)) \leqslant 0$, so for this $v \in \hat{\mathcal{L}}'$, $\text{Ldim}(\hat{\mathcal{G}}(\alpha_{d+1}, v)) = \text{Ldim}(\hat{\mathcal{G}}(\alpha_{d+1} - \alpha_\Delta, v)) = 0$. The k_{d+1}-irreducibility of $\hat{\mathcal{G}}(\alpha_{d+1} - \alpha_\Delta, v)$ follows from the fact that a class with Littlestone dimension 0 contains a single function, and is thus k-irreducible for all $k \in \mathbb{N}$. □

In order to apply Lemma 4.7 in the proof of Lemma 5.4 below, we will need an upper bound on $\text{ht}(\hat{\mathbf{x}})$ for the tree $\hat{\mathbf{x}}$ output by `ReduceTree`. Lemma 5.2 provides this upper bound; roughly speaking, the growth is exponential in t (recall $k_t = k' \cdot 2^t$ from step 3 of Algorithm 1) because the tree may grow in height by k_t with each increase of t by 1 (due to step 6(d)ii of the algorithm), and in order to satisfy the preconditions of Lemma 4.7 we need to ensure that k_{t+1} is an upper bound on $\text{ht}(\hat{\mathbf{x}}^{(t)})$ for each t.

Lemma 5.2. *For all t the tree $\hat{\mathbf{x}}^{(t)}$ of Algorithm 1 satisfies $\mathrm{ht}(\hat{\mathbf{x}}^{(t)}) \leqslant k_{t+1} - k'$. In particular, the tree $\hat{\mathbf{x}}$ satisfies $\mathrm{ht}(\hat{\mathbf{x}}) \leqslant k_{t_{\mathrm{final}}+1} - k'$.*

Proof. We prove that $\mathrm{ht}(\hat{\mathbf{x}}^{(t)}) \leqslant k_{t+1} - k' = k' \cdot 2^{t+1} - k'$ by induction. For the base case, note that $\mathrm{ht}(\hat{\mathbf{x}}^{(0)}) = 0 < 2k' - k' = k' \cdot 2^1 - k'$. Since at step 6(d)ii of the algorithm, in the tth iteration, each leaf is labeled with a tuple of length at most k_t, it follows that

$$\mathrm{ht}(\hat{\mathbf{x}}^{(t)}) \leqslant \mathrm{ht}(\hat{\mathbf{x}}^{(t-1)}) + k_t \leqslant k_t - k' + k_t = k_{t+1} - k',$$

for all $t \geqslant 1$. The lemma statement follows since $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(t_{\mathrm{final}})}$. □

For each $\alpha \in [0,1]$ and $t \in [d+1]$, define the set:

$$\mathcal{M}_{\alpha,t} := \left\{ S \in (\mathcal{X} \times \{-1,1\})^{k_t - k'} : \begin{array}{c} \mathcal{F}_{P,\alpha-\alpha_\Delta/3}|_S \text{ is } k_t\text{-irreducible and nonempty,} \\ \text{and } \mathrm{Ldim}(\mathcal{F}_{P,\alpha-\alpha_\Delta/3}|_S) = \mathrm{Ldim}(\mathcal{F}_{P,\alpha+\alpha_\Delta/3}|_S) \end{array} \right\} \tag{18}$$

Lemma 5.3. *Suppose that the event E_{good} occurs. Then for $t = t_{\mathrm{final}} + 1$, the set $\mathcal{M}_{\alpha_t - \alpha_\Delta/2,t}$ is nonempty.*

Proof. Let v be a node as guaranteed by Lemma 5.1, i.e., so that $\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)$ is k_t-irreducible, and so that $\mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v)) = \mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t, v)) \geqslant 0$. Since the event E_{good} holds,

$$\hat{\mathcal{G}}(\alpha_t - \alpha_\Delta, v) = \mathcal{F}_{\hat{P}_{S_n},\alpha_t-\alpha_\Delta}|_{\mathbf{a}(v)} \subset \mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v)} = \hat{\mathcal{G}}(\alpha_t, v).$$

It follows from Lemma 4.2 that $\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v)}$ is k_t-irreducible and that $\mathrm{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(v)}) = \mathrm{Ldim}(\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v)})$. Since the height of the tree $\hat{\mathbf{x}}^{(t-1)} = \hat{\mathbf{x}}^{(t_{\mathrm{final}})}$ is at most $k_t - k'$ (Lemma 5.2), it follows that the number of tuples in $\mathbf{a}(v)$ is at most $k_t - k'$; thus, after duplicating some of the tuples in $\mathbf{a}(v)$ if necessary, we get that $\mathbf{a}(v) \in \mathcal{M}_{\alpha-\alpha_\Delta/2,t}$. □

For any $\alpha \in [0,1], t \in [d+1]$ for which $\mathcal{M}_{\alpha,t}$ is nonempty, define:

$$S^\star_{\alpha,t} \in \operatorname*{arg\,max}_{S \in \mathcal{M}_{\alpha,t}} \{\mathrm{Ldim}(\mathcal{F}_{P,\alpha}|_S)\}, \qquad \ell^\star_{\alpha,t} := \max_{S \in \mathcal{M}_{\alpha,t}} \{\mathrm{Ldim}(\mathcal{F}_{P,\alpha}|_S)\} \geqslant 0. \tag{19}$$

Also set

$$\sigma^\star_{\alpha,t} := \mathrm{SOA}_{\mathcal{F}_{P,\alpha}|_{S^\star_{\alpha,t}}}. \tag{20}$$

We emphasize here that $\mathcal{M}_{\alpha,t}$ and $S^\star_{\alpha,t}$ are both independent of the output of the algorithm `ReduceTree` (and in particular, they do not depend on the particular input dataset S_n).

Lemma 5.4. *Under the event E_{good}, the following holds: for $t = t_{\mathrm{final}} + 1 \in [d+1]$ and some leaf $\hat{v} \in \hat{\mathcal{L}}'$, we have $\sigma^\star_{\alpha_t-\alpha_\Delta/2,t} = \mathrm{SOA}_{\hat{\mathcal{G}}(\alpha_t-2\alpha_\Delta/3,\hat{v})}$. (In particular, for this t, $\sigma^\star_{\alpha_t-\alpha_\Delta/2,t}$ is well-defined, i.e., $\mathcal{M}_{\alpha_t-\alpha_\Delta/2,t}$ is nonempty.)*

Moreover, $\hat{\mathcal{G}}(\alpha_t - 2\alpha_\Delta/3, \hat{v})$ is k'-irreducible and nonempty, and $\mathrm{Ldim}(\hat{\mathcal{G}}(\alpha_t - 2\alpha_\Delta/3, \hat{v})) = \ell^\star_{\alpha_t-\alpha_\Delta/2,t} \geqslant 0$.

Proof. By Lemma 5.1, for $t = t_{\mathrm{final}}+1 \in [d+1]$, there is some leaf $v' \in \hat{\mathcal{L}}'$ so that $\mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t-\alpha_\Delta}|_{\mathbf{a}(v')}) = \mathrm{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v')}) \geqslant 0$ and $\mathcal{F}_{\hat{P}_{S_n},\alpha_t-\alpha_\Delta}|_{\mathbf{a}(v')}$ is k_t-irreducible. Under the event E_{good}, for each node v of the tree $\hat{\mathbf{x}}$ output by the algorithm, we have that

$$\begin{aligned} \mathcal{F}_{\hat{P}_{S_n},\alpha_t-\alpha_\Delta}|_{\mathbf{a}(v)} \subset \mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{\hat{P}_{S_n},\alpha_t-4\alpha_\Delta/6}|_{\mathbf{a}(v)} \\ \subset \mathcal{F}_{P,\alpha_t-3\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{\hat{P}_{S_n},\alpha_t-2\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(v)} \subset \mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v)}. \end{aligned} \tag{21}$$

Now we apply Lemma 4.7 with $\mathcal{J} = \mathcal{J}' = \mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}$, $\mathcal{H} = \mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}$, $\mathcal{G} = \mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}$, $k = k_t, k' = k'$, $\mathbf{x}$ equal to the tree $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(t_{\text{final}})}$ output by `ReduceTree`, and $S^\star = S^\star_{\alpha_t-\alpha_\Delta/2,t}$. Since $t = t_{\text{final}} + 1$, Lemma 5.3 guarantees that $S^\star$ is well-defined (in particular, that $\mathcal{M}_{\alpha_t-\alpha_\Delta/2,t}$ is nonempty). We check that the preconditions of Lemma 4.7 hold: Notice that (8) holds by the definitions (18) and (19), and that $\mathcal{H}|_{S^\star} = \mathcal{F}_{P,\alpha_t-\alpha_\Delta/2-\alpha_\Delta/3}|_{S^\star}$ is k_t-irreducible, also by (18) and (19). By definition of $\ell^\star_{\alpha,t}$ in (19), we have

$$\ell^\star_{\alpha_t-\alpha_\Delta/2,t} = \text{Ldim}(\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{S^\star}) = \text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{S^\star}).$$

Lemma 5.2 establishes that $\text{ht}(\hat{\mathbf{x}}) \leqslant k_t - k'$. Also, from the guarantee on v' in Lemma 5.1, (21), and Lemma 4.2, we have $\text{Ldim}(\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v')}) = \text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(v')})$ and $\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(v')}$ is k_t-irreducible. Thus $\mathbf{a}(v') \in \mathcal{M}_{\alpha_t-\alpha_\Delta/2,t}$, so by definition of $\ell^\star_{\alpha,t}$,

$$\ell^\star_{\alpha_t-\alpha_\Delta/2,t} \geqslant \text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}|_{\mathbf{a}(v')}).$$

Moreover, for any other leaf u of the tree $\hat{\mathbf{x}}$, we have, by definition of $\hat{\mathcal{L}}' = \hat{\mathcal{L}}'_{t_{\text{final}}+1}$,

$$\text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(u)}) \leqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(u)}) \leqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v')}) = \text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}|_{\mathbf{a}(v')}) \leqslant \ell^\star_{\alpha_t-\alpha_\Delta/2,t}.$$

(In more detail, the first inequality above holds due to (21), the second inequality is due to the fact that $v' \in \hat{\mathcal{L}}' = \hat{\mathcal{L}}'_{t_{\text{final}}+1}$ (see step 6b of `ReduceTree`), and the equality holds due to (21) and $\text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t-\alpha_\Delta}|_{\mathbf{a}(v')}) = \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v')})$). Then the hypotheses of Lemma 4.7 hold and letting $\mathcal{J}' = \mathcal{J} = \mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}$, it follows that for some leaf $\hat{v}$ of $\hat{\mathbf{x}}$, we have

$$\sigma^\star_{\alpha_t-\alpha_\Delta/2,t} = \text{SOA}_{\mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}|_{S^\star}} = \text{SOA}_{\mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}|_{\mathbf{a}(\hat{v})}}$$

as well as $\text{Ldim}(\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}) = \text{Ldim}(\mathcal{F}_{P,\alpha_t-\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}) = \ell^\star_{\alpha_t-\alpha_\Delta/2,t}$, and that $\mathcal{F}_{P,\alpha_t-5\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}$ is k'-irreducible. From (21), it follows that $\text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t-4\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}) = \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t-2\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}) = \ell^\star_{\alpha_t-\alpha_\Delta/2,t} \geqslant 0$, and that $\mathcal{F}_{\hat{P}_{S_n},\alpha_t-4\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})} = \hat{\mathcal{G}}(\alpha_t - 2\alpha_\Delta/3, \hat{v})$ is k'-irreducible. Then by (21) and Lemma 4.3, we have

$$\sigma^\star_{\alpha_t-\alpha_\Delta/2,t} = \text{SOA}_{\mathcal{F}_{P,\alpha_t-\alpha_\Delta/2}|_{\mathbf{a}(\hat{v})}} = \text{SOA}_{\mathcal{F}_{\hat{P}_{S_n},\alpha_t-2\alpha_\Delta/3}|_{\mathbf{a}(\hat{v})}} = \text{SOA}_{\hat{\mathcal{G}}(\alpha_t-2\alpha_\Delta/3,\hat{v})}.$$

Finally we check that $\hat{v} \in \hat{\mathcal{L}}' = \hat{\mathcal{L}}'_t = \hat{\mathcal{L}}'_{t_{\text{final}}+1}$, i.e., all leaves u of the tree $\hat{\mathbf{x}}$ satisfy $\text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(u)}) \leqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(\hat{v})})$. This is a consequence of the fact that for all such u,

$$\text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(\hat{v})}) \geqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t-2\alpha_\Delta/6}|_{\mathbf{a}(\hat{v})}) = \ell^\star_{\alpha_t-\alpha_\Delta/2,t} \geqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(v')}) \geqslant \text{Ldim}(\mathcal{F}_{\hat{P}_{S_n},\alpha_t}|_{\mathbf{a}(u)}),$$

since $v' \in \hat{\mathcal{L}}'$. □

Lemma 5.5. *The set $\hat{\mathcal{S}}$ output by `ReduceTree` has size $|\hat{\mathcal{S}}| \leqslant \prod_{t=1}^d (k_t + 1)$.*

Proof. It suffices to show that for $t \in [d]$, the tree $\mathbf{x}^{(t)}$ has at most $\prod_{t'=1}^t (k_{t'} + 1)$ leaves. In turn, this is a simple consequence of the fact that $\mathbf{x}^{(0)}$ has a single leaf, and the tree $\mathbf{x}^{(t)}$ is formed by adding at most $k_t + 1$ leaves to some of the leaves of the tree $\mathbf{x}^{(t-1)}$. □

5.2 Building block: sparse selection protocol

We use the following primitive for solving the *sparse selection* problem from [GKM20]:

Definition 5.1 (Sparse selection)**.** For $m, \ell \in \mathbb{N}$, in (m, ℓ)-*sparse selection* problem, there is some (possibly infinite) universe $\mathcal{U}$, and m users. Each user $i \in [m]$ is given some set $\mathcal{S}_i \subset \mathcal{U}$ of size $|\mathcal{S}_i| \leqslant \ell$. An algorithm solves the (m, ℓ)-sparse selection problem with with additive error η if it outputs some universe element $\hat{u} \in \mathcal{U}$ such that

$$|\{i : \hat{u} \in \mathcal{S}_i\}| \geqslant \max_{u \in \mathcal{U}} |\{i : u \in \mathcal{S}_i\}| - \eta. \tag{22}$$

Proposition 5.6 shows that the sparse selection problem can be solved privately with error independent of the size of the universe $\mathcal{U}$. It can be thought of as an analogue of the private stable histogram of [BNS16, Proposition 2.20] for the problem of private selection.

Proposition 5.6 ([GKM20], Lemma 36)**.** *For $\varepsilon \in (0, 1]$, $\delta \in (0, 1)$, $\beta \in (0, 1)$, there is an (ε, δ)-differentially private algorithm that given an input dataset to the (m, ℓ)-sparse selection problem, outputs a universe element $\hat{u}$ such that with probability at least $1 - \beta$, the error of $\hat{u}$ is*

$$O\left(\frac{1}{\varepsilon} \log\left(\frac{m\ell}{\varepsilon\delta\beta}\right)\right).$$

5.3 Overall algorithm

In this section we combine the components of Sections 5.1 and 5.2 to prove the following theorem, which gives an improper learner for hypothesis classes with sample complexity polynomial in the Littlestone dimension.

Theorem 5.7. *Let $\mathcal{F}$ be a concept class of domain $\mathcal{X}$ with $d_{\mathrm{V}} := \mathrm{VCdim}(\mathcal{F}), d_{\mathrm{L}} := \mathrm{Ldim}(\mathcal{F})$. For any $\varepsilon, \delta, \eta \in (0, 1)$, for some*

$$n = O\left(\frac{d_{\mathrm{L}}^5 d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}}}{\varepsilon\delta\eta\beta}\right)}{\varepsilon\eta^2}\right)$$

the algorithm `PolyPriLearn` *(Algorithm 2) takes as input n i.i.d. samples from any realizable distribution P, is (ε, δ)-differentially private, and produces a hypothesis $\hat{f}$ so that $\mathrm{err}_P(\hat{f}) \leqslant \eta$ with probability at least $1 - \beta$.*

Moreover, under the same $(1 - \beta)$-probability event, $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ for some $\mathcal{G} \subset \mathcal{F}$ for which $\mathcal{G}$ is $\left\lceil \frac{64 C_0 d_{\mathrm{L}}}{\eta^2} \right\rceil$-irreducible.

Remark 5.1. The assertion that $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ for some $\mathcal{G}$ which is $\left\lceil \frac{64 C_0 d_{\mathrm{L}}}{\eta^2} \right\rceil$-irreducible is for use in Section 6 when we use `PolyPriLearn` as a component of a proper private learning algorithm.

`PolyPriLearn` (Algorithm 2) operates as follows. For sufficiently large positive integers m, n_0, `PolyPriLearn` runs `ReduceTree` on m independent samples of size n_0 from the distribution P. Each run of `ReduceTree` outputs some set $\hat{\mathcal{S}}$ of classifiers in $\{-1, 1\}^{\mathcal{X}}$. `PolyPriLearn` then uses the sparse selection protocol of Proposition 5.6 to choose some classifier that lies in many of the sets $\hat{\mathcal{S}}$.

Algorithm 2: `PolyPriLearn`

Input: Parameters $\varepsilon, \delta, \eta, \beta \in (0,1)$, i.i.d. samples $(x,y) \in \mathcal{X} \times \{-1,1\}$ from a realizable distribution P.

1. Set $m \leftarrow \frac{C(d_{\mathrm{L}}^3 \log(1/(\varepsilon\delta\beta\eta)))}{\varepsilon}$, $n_0 \leftarrow \frac{Cd_{\mathrm{L}}^2 d_{\mathrm{V}} \log\left(\frac{d_{\mathrm{L}} m}{\eta\beta}\right)}{\eta^2}$, $n \leftarrow n_0 m$, $\alpha_\Delta \leftarrow 6 \cdot \alpha(n_0, \beta/(2m))$, where $C > 0$ is a sufficiently large constant.
   Also set $k' \leftarrow \max\{\lceil n_0 \cdot (d_{\mathrm{L}}+3)\alpha_\Delta \rceil, \left\lceil \frac{64 C_0 d_{\mathrm{L}}}{\eta^2} \right\rceil\}$, where C_0 is the constant of Theorem 2.1.
2. For $1 \leqslant j \leqslant m$:
   - Run the algorithm `ReduceTree` with $n = n_0, \gamma = \beta/(2m)$, and the parameters α_Δ, k' set in step 1 (i.e., with a fresh i.i.d. sample from P). Let its output set $\hat{\mathcal{S}}$ (defined in (16)) be denoted by $\hat{\mathcal{S}}^{(j)}$.
3. Run the sparse selection protocol of Proposition 5.6 on the sets $\hat{\mathcal{S}}^{(1)}, \ldots, \hat{\mathcal{S}}^{(m)}$, and output the function $\hat{f} : \mathcal{X} \to \{-1,1\}$ that it outputs.

Proof of Theorem 5.7. In the proof we will often refer to the parameters $n_0, m, \alpha_\Delta, k'$, which are set in step 1 of `PolyPriLearn` (Algorithm 2). Notice that by our choice of

$$n_0 = \frac{Cd_{\mathrm{L}}^2 d_{\mathrm{V}} \log\left(\frac{d_{\mathrm{L}} m}{\eta\beta}\right)}{\eta^2},$$

as long as C is sufficiently large, we have that $\alpha_\Delta := 6 \cdot \alpha(n_0, \beta/(2m))$ satisfies $(d_{\mathrm{L}}+3) \cdot \alpha_\Delta < \eta$. Recall the definition of $\alpha_t := (d_{\mathrm{L}} + 3 - t) \cdot \alpha_\Delta$ for $1 \leqslant t \leqslant d_{\mathrm{L}}+1$ from `ReduceTree`.

For $1 \leqslant j \leqslant m$, let $T^{(j)} := \{(x_1^{(j)}, y_1^{(j)}), \ldots, (x_{n_0}^{(j)}, y_{n_0}^{(j)})\}$ be the dataset of size n_0 drawn (i.i.d. from P) in the jth iteration of Step 2 of `PolyPriLearn`. Let $\hat{P}^{(j)} := \frac{1}{n_0}\sum_{i=1}^{n_0} \delta_{(x_i^{(j)}, y_i^{(j)})}$ be the empirical measure over $T^{(j)}$.

We say that a class $\mathcal{G} \subset \mathcal{F}$ is a *finite restriction subclass (of $\mathcal{F}$)* if we can write $\mathcal{G} = \mathcal{F}|_{(x_1,y_1),\ldots,(x_M,y_M)}$ for some $(x_1,y_1), \ldots, (x_M, y_M) \in \mathcal{X} \times \{-1,1\}$. Note that the set of all finite restriction subclasses of $\mathcal{F}$ is countable by our assumption that $\mathcal{X}$ is countable. It follows that the set of all finite unions of finite restriction subclasses of $\mathcal{F}$ is also countable. Now define

$$\tilde{\mathcal{F}} = \mathcal{F} \cup \{\mathrm{SOA}_{\mathcal{G}} : \substack{\mathcal{G} \subset \mathcal{F},\ \mathcal{G} \text{ is nonempty, } (d_{\mathrm{L}}+1)\text{-irreducible,} \\ \text{and a finite union of finite restriction subclasses of } \mathcal{F}}\}.$$

Notice that the set $\hat{\mathcal{S}}$ output by `ReduceTree` consists entirely of functions in $\tilde{\mathcal{F}}$. (This follows since the set $\hat{\mathcal{S}}$ consists of hypotheses of the form $\mathrm{SOA}_{\hat{\mathcal{G}}(\alpha,v)}$, where $\hat{\mathcal{G}}(\alpha, v)$ is k'-irreducible: we then use that $d_{\mathrm{L}}+1 \leqslant k'$ and that for any $\alpha \in [0,1]$, and any dataset S_n, $\mathcal{F}_{\hat{P}_{S_n},\alpha}$ is the union of at most 2^n finite restriction subclasses of $\mathcal{F}$.) Moreover, $\tilde{\mathcal{F}}$ is countable, and Lemma 4.4 gives that $\mathrm{VCdim}(\tilde{\mathcal{F}}) \leqslant \mathrm{Ldim}(\tilde{\mathcal{F}}) \leqslant d_{\mathrm{L}}$. Then Theorem 2.1 gives that

$$\Pr\left[\forall j \in [m] : \sup_{\tilde{f} \in \tilde{\mathcal{F}}} \left|\mathrm{err}_P(\tilde{f}) - \mathrm{err}_{\hat{P}^{(j)}}(\tilde{f})\right| \leqslant \frac{\alpha_\Delta}{6}\right] \geqslant 1 - (\beta/(2m)) \cdot m = 1 - \beta/2. \tag{23}$$

Let E_0 be the event inside the probability above, namely that for all $j \in [m]$, $\sup_{\tilde{f} \in \tilde{\mathcal{F}}} \left|\mathrm{err}_P(\tilde{f}) - \mathrm{err}_{\hat{P}^{(j)}}(\tilde{f})\right| \leqslant \frac{\alpha_\Delta}{6}$. Since $\tilde{\mathcal{F}} \supseteq \mathcal{F}$, E_0 contains the event that E_{good} simultaneously holds for each dataset $T^{(1)}, \ldots, T^{(m)}$ (recall that E_{good} was defined for any dataset $T^{(j)}$ in (14)).

The bulk of the proof of Theorem 5.7 is to show the following two claims:

Claim 5.8. *Suppose* $m > \frac{Cd_{\mathrm{L}}^3 \log\left(\frac{1}{\varepsilon\delta\beta\eta}\right)}{\varepsilon}$ *for a sufficiently large constant* $C > 0$. *There is an event* E_1 *that occurs with probability at least* $1-\beta/2$ *(over the randomness of the dataset and the algorithm), so that under* $E_1 \cap E_0$, `PolyPriLearn` *outputs a hypothesis* $\mathrm{SOA}_{\mathcal{G}}$, *for some* $\mathcal{G} \subset \mathcal{F}$ *so that* $\mathcal{G}$ *is* k'*-irreducible. Moreover, this hypothesis belongs to* $\hat{\mathcal{S}}^{(j)}$ *for some* $j \in [m]$.

Claim 5.9. *Suppose* $k' \geqslant \lceil n_0 \cdot (d_{\mathrm{L}}+2) \cdot \alpha_\Delta \rceil$. *Under the event* $E_1 \cap E_0$, *the output of* `PolyPriLearn` *has empirical error at most* $(d_{\mathrm{L}}+2)\cdot\alpha_\Delta$ *on at least one of the* m *datasets* $T^{(j)}$ *drawn in Step 2 of* `PolyPriLearn`.

Assuming Claims 5.8 and 5.9, we complete the proof of Theorem 5.7. Notice that the assumptions of Claims 5.8 and 5.9 hold by our choices of m, k' in Step 1 of `PolyPriLearn`. Denote the output of `PolyPriLearn` by $\hat{f} : \mathcal{X} \to \{-1, 1\}$. By Claim 5.9, we have that $\mathrm{err}_{\hat{P}^{(j)}}(\hat{f}) \leqslant (d_{\mathrm{L}}+2)\cdot\alpha_\Delta$ for some $j \in [m]$. By Claim 5.8 and the definition of the sets $\hat{\mathcal{S}}^{(j)}$ in (16), we have that under the event $E_1 \cap E_0$, $\hat{f} \in \tilde{\mathcal{F}}$; moreover, $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ for some $\mathcal{G} \subset \mathcal{F}$ which is $\left\lceil \frac{64C_0 d_{\mathrm{L}}}{\eta^2} \right\rceil$-irreducible, by the choice of k' in step 1 of Algorithm 2. By the definition of E_0, it follows that under the event $E_0 \cap E_1$, since $\hat{f} \in \tilde{\mathcal{F}}$, we have $\mathrm{err}_P(\hat{f}) \leqslant (d_{\mathrm{L}}+2)\cdot\alpha_\Delta + \alpha_\Delta/6 \leqslant (d_{\mathrm{L}}+3)\cdot\alpha_\Delta \leqslant \eta$. By (23) and a union bound, $\Pr[E_0 \cap E_1] \geqslant 1-\beta$, so $\Pr[\mathrm{err}_P(\hat{f}) \leqslant \eta] \geqslant 1-\beta$, as desired.

That `PolyPriLearn` is (ε, δ)-differentially private follows as an immediate consequence of Proposition 5.6 and the fact that each data point lies in exactly one $T^{(j)}$. Summarizing, the sample complexity of `PolyPriLearn` is

$$n_0 \cdot m \leqslant O\left(\frac{d_{\mathrm{L}}^5 d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}}}{\varepsilon\delta\eta\beta}\right)}{\varepsilon\eta^2}\right)$$

Finally we prove Claims 5.8 and 5.9.

Proof of Claim 5.8. Notice that for each $j \in [m]$, each element of $\hat{\mathcal{S}}^{(j)}$ is of the form $\mathrm{SOA}_{\mathcal{G}}$ for some $\mathcal{G} \subset \mathcal{F}$ which is k'-irreducible, and thus $(d_{\mathrm{L}}+1)$-irreducible (as $k' \geqslant d_{\mathrm{L}}+1$). It therefore suffices to show that under the event $E_0 \cap E_1$ (for an appropriate choice of E_1), `PolyPriLearn` outputs some element of some $\hat{\mathcal{S}}^{(j)}$, $j \in [m]$.

For $\alpha \in [0,1], t \in [d_{\mathrm{L}}+1]$, recall the definition $\mathcal{M}_{\alpha,t}$ in (18), and for those α, t for which $\mathcal{M}_{\alpha,t}$ is nonempty, the definition of $\sigma^\star_{\alpha,t}$ in (20). By the definition of $\hat{\mathcal{S}}^{(j)}$ (see (16)) and Lemma 5.4, under the event E_0 each $\hat{\mathcal{S}}^{(j)}$ contains at least one of $\sigma^\star_{\alpha_t - \alpha_\Delta/2, t}$ for some $t \in [d_{\mathrm{L}}+1]$ (which is well-defined). By the pigeonhole principle, it follows that some $\sigma^\star_{\alpha_t-\alpha_\Delta/2,t}$ lies in at least $\lceil m/(d_{\mathrm{L}}+1) \rceil$ sets $\hat{\mathcal{S}}^{(j)}$.

By Lemma 5.5, we have that

$$|\hat{\mathcal{S}}^{(j)}| \leqslant \prod_{t=1}^{d_{\mathrm{L}}} (k_t+1) = \prod_{t=1}^{d_{\mathrm{L}}} k' \cdot 2^t = (k')^{d_{\mathrm{L}}} \cdot 2^{(d_{\mathrm{L}}+1)d_{\mathrm{L}}/2} \leqslant 2^{d_{\mathrm{L}}^2 + d_{\mathrm{L}} \log k'}.$$

Now choose $\nu > 0$ so that the $(m, 2^{d_{\mathrm{L}}^2 + d_{\mathrm{L}} \log k'})$-sparse selection protocol of Proposition 5.6 (with universe $\mathcal{U} = \tilde{\mathcal{F}}$), has error at most ν on some event E_1 with probability at least $1-\beta/2$. By Proposition 5.6, we may choose $\nu = \frac{C}{\varepsilon} \log\left(\frac{m 2^{d_{\mathrm{L}}^2 + d_{\mathrm{L}} \log k'}}{\varepsilon\delta\beta}\right)$ for a sufficiently large constant C.

Summarizing, under the event $E_0 \cap E_1$, as long as $\nu < \lceil m/(d_{\mathrm{L}}+1) \rceil$, the hypothesis $\hat{f}$ output by the sparse selection protocol belongs to some set $\hat{\mathcal{S}}^{(j)}$. Since

$$k' \leqslant 200C_0 \max\left\{n_0(d_{\mathrm{L}}+3)\alpha_\Delta, \frac{d_{\mathrm{L}}}{\eta^2}\right\} \leqslant 200C_0 \max\left\{\frac{d_{\mathrm{L}}}{\eta^2}, n_0\eta\right\} \leqslant \frac{200C_0 d_{\mathrm{L}}^2 d_{\mathrm{V}} \log\left(\frac{d_{\mathrm{L}} m}{\eta\beta}\right)}{\eta^2}$$

to ensure $\nu < \lceil m/(d_{\mathrm{L}}+1)\rceil$ it suffices to have

$$m > \frac{C'(d_{\mathrm{L}}+1)}{\varepsilon}\left(\log(m) + d_{\mathrm{L}}^2 + \log\left(\frac{1}{\varepsilon\delta\beta}\right) + d_{\mathrm{L}}\left(\log(1/\eta) + \log\log\left(\frac{m}{\beta}\right)\right)\right),$$

for which it in turn suffices that

$$m \geqslant \frac{C'' d_{\mathrm{L}}^3 \log\left(\frac{1}{\varepsilon\delta\beta\eta}\right)}{\varepsilon}$$

for sufficiently large constants C', C''. □

Proof of Claim 5.9. By Claim 5.8, it suffices to show that under the event $E_1 \cap E_0$, each element of $\hat{\mathcal{S}}^{(j)}$ has empirical error at most $(d_{\mathrm{L}}+2)\cdot\alpha_\Delta$ on the dataset $T^{(j)}$. By definition, each element of $\hat{\mathcal{S}}^{(j)}$ is of the form $\mathrm{SOA}_{\mathcal{F}_{\hat{P}^{(j)},\alpha_t-2\alpha_\Delta/3}|_{\mathbf{a}(v)}}$ for some node v of the tree $\hat{\mathbf{x}}$ output by `ReduceTree` for which $\mathcal{F}_{\hat{P}^{(j)},\alpha_t-2\alpha_\Delta/3|_{\mathbf{a}(v)}}$ is nonempty and k'-irreducible (see (16)). Fix any such element, and write $\hat{\mathcal{H}} := \mathcal{F}_{\hat{P}^{(j)},\alpha_t-2\alpha_\Delta/3}|_{\mathbf{a}(v)}$. By definition we have that each $f \in \hat{\mathcal{H}} = \mathcal{F}_{\hat{P}^{(j)},\alpha_t-2\alpha_\Delta/3}|_{\mathbf{a}(v)} \subset \mathcal{F}_{\hat{P}^{(j)},\alpha_t-2\alpha_\Delta/3}$ satisfies

$$\mathrm{err}_{\hat{P}^{(j)}}(f) \leqslant \alpha_t - 2\alpha_\Delta/3 \leqslant \alpha_1 - 2\alpha_\Delta/3 = (d_{\mathrm{L}}+2)\cdot\alpha_\Delta - 2\alpha_\Delta/3. \tag{24}$$

Let $\ell = \lceil n_0\alpha_\Delta\cdot(d_{\mathrm{L}}+2)\rceil$. Suppose for the purpose of contradiction that

$$\mathrm{err}_{\hat{P}^{(j)}}(\mathrm{SOA}_{\hat{\mathcal{H}}}) \geqslant \alpha_\Delta\cdot(d_{\mathrm{L}}+2).$$

Let $i_1,\ldots,i_\ell \in [n_0]$ be indices on which $\mathrm{SOA}_{\hat{\mathcal{H}}}$ is incorrect; i.e., for $t\in[\ell]$, we have $\mathrm{SOA}_{\hat{\mathcal{H}}}(x_{i_t}^{(j)}) = -y_{i_t}^{(j)}$, i.e., $\mathrm{Ldim}(\hat{\mathcal{H}}|_{(x_{i_t}^{(j)},-y_{i_t}^{(j)})}) = \mathrm{Ldim}(\hat{\mathcal{H}})$. Since $\hat{\mathcal{H}}$ is k'-irreducible and $k' \geqslant \ell$, it follows that

$$\mathrm{Ldim}(\hat{\mathcal{H}}|_{(x_{i_1}^{(j)},-y_{i_1}^{(j)}),\ldots,(x_{i_\ell}^{(j)},-y_{i_\ell}^{(j)})}) = \mathrm{Ldim}(\hat{\mathcal{H}}),$$

and in particular since $\hat{\mathcal{H}}$ is nonempty there is some $f\in\hat{\mathcal{H}}$ so that for $t\in[\ell]$, $f(x_{i_t}^{(j)}) = -y_{i_t}^{(j)}$, i.e., $\mathrm{err}_{\hat{P}^{(j)}}(f) \geqslant \ell/n_0 > \alpha_\Delta\cdot(d_{\mathrm{L}}+2) - 2\alpha_\Delta/3$. This is a contradiction to (24). □

□

6 Proper private learner for Littlestone classes

In this section we show how to use the improper private learner of Theorem 5.7 to obtain a proper one, thus proving Theorem 6.4 (the formal version of Theorem 1.1). For simplicity we assume in this section that $\mathcal{X},\mathcal{F}$ are finite. The case in which they are allowed to be infinite is treated in Appendix A. Let $\Delta(\mathcal{X}),\Delta(\mathcal{F})$ be the spaces of probability distributions over $\mathcal{X},\mathcal{F}$, respectively.

Lemma 6.1. *Let C_0 be the constant of Theorem 2.1. Fix any $\alpha\in(0,1)$ and $\mathcal{G}\subset\mathcal{F}$ which is $\left\lceil\frac{C_0 d}{\alpha^2}\right\rceil$-irreducible and suppose $\mathrm{VCdim}(\mathcal{F})\leqslant d$ for some $d\in\mathbb{N}$. Then it holds that*

$$\sup_{P\in\Delta(\mathcal{X})}\inf_{D\in\Delta(\mathcal{F})}\mathbb{E}_{x\sim P,h\sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x)\neq h(x)]\right] \leqslant \alpha.$$

Proof. It suffices to show that for any $P \in \Delta(\mathcal{X})$, there is some $g \in \mathcal{F}$ so that $\mathbb{E}_{x\sim P}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq g(x)]\right] \leqslant \alpha$. By Theorem 2.1, for $n = \left\lceil \frac{C_0 d}{\alpha^2} \right\rceil$, then with probability at least $1/2$ over a sample $x_1, \ldots, x_n \sim P$, we have

$$\sup_{f\in\mathcal{F}} \left| \mathbb{E}_{x\sim P}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq f(x)]\right] - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x_i) \neq f(x_i)] \right| \leqslant \alpha. \tag{25}$$

Fix any sample $x_1, \ldots, x_n$ for which (25) holds. Since $\mathcal{G}$ is n-irreducible, there exists some $g \in \mathcal{G}$ so that $g(x_i) = \mathrm{SOA}_{\mathcal{G}}(x_i)$ for each $i \in [n]$. Then by (25), we have that $\mathbb{E}_{x\sim P}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq g(x)]\right] \leqslant \alpha$. □

Lemma 6.2. *Let C_0 be the constant of Theorem 2.1. Fix any $\alpha \in (0,1)$ and $\mathcal{G} \subset \mathcal{F}$ which is $\left\lceil \frac{C_0 d}{\alpha^2} \right\rceil$-irreducible and suppose $\mathrm{VCdim}(\mathcal{F}) \leqslant d$ and $\mathrm{VCdim}^\star(\mathcal{F}) \leqslant d^\star$ for some $d, d^\star \in \mathbb{N}$. Then there is a set $\mathcal{H} \subset \mathcal{F}$, depending only on the function $\mathrm{SOA}_{\mathcal{G}} : \mathcal{X} \to \{-1,1\}$, and of size $|\mathcal{H}| \leqslant \left\lceil \frac{C_0 d^\star}{\alpha^2} \right\rceil$, so that for any distribution $P \in \Delta(\mathcal{X})$, it holds that*

$$\min_{h\in\mathcal{H}} \mathbb{E}_{x\sim P}\left[\mathbb{1}[h(x) \neq \mathrm{SOA}_{\mathcal{G}}(x)]\right] \leqslant 2\alpha. \tag{26}$$

Proof. By Lemma 6.1 and von Neumann's minimax theorem, it holds that

$$\begin{aligned} &\inf_{D\in\Delta(\mathcal{F})} \sup_{P\in\Delta(\mathcal{X})} \mathbb{E}_{x\sim P, h\sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] \\ &= \sup_{P\in\Delta(\mathcal{X})} \inf_{D\in\Delta(\mathcal{F})} \mathbb{E}_{x\sim P, h\sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] \leqslant \alpha. \end{aligned} \tag{27}$$

Fix some $D \in \Delta(\mathcal{F})$ obtaining the infimum in (27); this is possible because $\Delta(\mathcal{F})$ is compact. Note that D depends only on $\mathrm{SOA}_{\mathcal{G}} \in \{0,1\}^{\mathcal{X}}$ (i.e., it can be written as a function of $\mathrm{SOA}_{\mathcal{G}}$). By Theorem 2.1, for $m = \left\lceil \frac{C_0 d^\star}{\alpha^2} \right\rceil$, with probability at least $1/2$ over an i.i.d. sample $h_1, \ldots, h_m \sim D$, we have that[10]

$$\sup_{x\in\mathcal{X}} \left| \mathbb{E}_{h\sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h_j(x)] \right| \tag{28}$$

$$= \sup_{x\in\mathcal{X}} \left| \mathbb{E}_{h\sim D}\left[\frac{h(x)}{2}\right] - \frac{1}{m}\sum_{j=1}^{m} \frac{h_j(x)}{2} \right| \leqslant \alpha \tag{29}$$

$$= \sup_{x\in\mathcal{X}} \left| \mathbb{E}_{h\sim D}\left[\mathbb{1}[h(x) \neq 1]\right] - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[h_j(x) \neq 1] \right| \leqslant \alpha.$$

(To see why the equalities above hold, note that for any $h \in \mathcal{F}$, if $\mathrm{SOA}_{\mathcal{G}}(x) = 1$, then $\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] = \frac{1-h(x)}{2}$, and if $\mathrm{SOA}_{\mathcal{G}}(x) = -1$, then $\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] = \frac{1+h(x)}{2}$.) Fix any $h_1, \ldots, h_m$ so that (28) holds, and set $\mathcal{H} := \{h_1, \ldots, h_m\}$. Write $h \sim_U \mathcal{H}$ to mean that h is drawn uniformly from $\mathcal{H}$. Then by (27) and (28), we have, for any $P \in \Delta(\mathcal{X})$,

$$\begin{aligned} &\mathbb{E}_{x\sim P, h\sim_U \mathcal{H}}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] \\ &\leqslant^{(28)} \mathbb{E}_{x\sim P, h\sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] + \alpha \\ &\leqslant^{(27)} 2\alpha. \end{aligned}$$

(26) is an immediate consequence of the above display. □

[10] As remarked by [MY16], this application of Theorem 2.1 to the dual class can be viewed as a sort of combinatorial and approximate version of Carathéodory's theorem.

6.1 Private proper learning protocol

Before introducing our private proper learning algorithm, we need the following basic lemma which establishes that the use of the exponential mechanism can output a good hypothesis privately from a class of small size:

Lemma 6.3 (Generic Private Learner, [KLN+08]). *Let $\mathcal{H} \subset \{-1,1\}^{\mathcal{X}}$ be a finite set of hypotheses. For*

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

there exists an $(\varepsilon, 0)$-differentially private algorithm `GenericLearner` $: (\mathcal{X} \times \{-1,1\})^n \to \mathcal{H}$ *so that the following holds. For any distribution P over $\mathcal{X} \times \{-1,1\}$ so that there exists $h^\star \in \mathcal{H}$ with*

$$\mathrm{err}_P(h^\star) \leqslant \alpha,$$

on input $S_n := \{(x_1,y_1),\ldots,(x_n,y_n)\} \sim P^n$, `GenericLearner` *outputs, with probability at least $1-\beta$, a hypothesis $\hat{h} \in \mathcal{H}$ so that*

$$\mathrm{err}_P(\hat{h}) \leqslant 2\alpha.$$

The precise formulation of Lemma 6.3 is proved in [BLM20b, Lemma 16].

Our algorithm, `PolyPriPropLearn`, for privately and properly learning a hypothesis class, is presented in Algorithm 3. Given sufficiently many samples from a realizable distribution P, `PolyPriPropLearn` first runs `PolyPriLearn` to come up with a hypothesis of the form $\mathrm{SOA}_{\mathcal{G}} \in \{-1,1\}^{\mathcal{X}}$ with low population loss on the distribution P. It then uses the guarantee of Lemma 6.2 to come up with a small subclass $\mathcal{H} \subset \mathcal{F}$ which is guaranteed to contain a hypothesis that performs nearly as well as $\mathrm{SOA}_{\mathcal{G}}$ on the distribution P. It then privately chooses such a hypothesis $\hat{h} \in \mathcal{H}$ using the exponential mechanism (Lemma 6.3).

Algorithm 3: `PolyPriPropLearn`

Input: Parameters $\varepsilon, \delta, \eta, \beta \in (0,1)$, i.i.d. samples $(x,y) \in \mathcal{X} \times \{-1,1\}$ from a realizable distribution P.

1. Run the algorithm `PolyPriLearn` with parameters $\varepsilon, \delta, \eta/4, \beta/2$. Let $\hat{f} \in \{-1,1\}^{\mathcal{X}}$ be its output.
   With probability at least $1-\eta/2$, it is then guaranteed that $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ for some k'-irreducible $\mathcal{G} \subset \mathcal{F}$. Choose any such $\mathcal{G}$.
2. Choose a set $\mathcal{H}$ as in Lemma 6.2 from the function $\mathrm{SOA}_{\mathcal{G}}$.
3. On a fresh sample of size $O\left(\frac{\log|\mathcal{H}|+\log(1/\beta)}{\eta\varepsilon}\right)$, run the `GenericLearner` of Lemma 6.3 with the set $\mathcal{H}$, and return its output $\hat{h}$.

Theorem 6.4 (Private proper PAC learning). *Let $\mathcal{F}$ be a concept class of domain $\mathcal{X}$ with $d_{\mathrm{V}} := \mathrm{VCdim}(\mathcal{F}), d_{\mathrm{L}} := \mathrm{Ldim}(\mathcal{F})$. For any $\varepsilon, \delta, \eta, \beta \in (0,1)$, for some*

$$n = O\left(\frac{d_{\mathrm{L}}^5 d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}}}{\varepsilon\delta\eta\beta}\right)}{\varepsilon\eta^2}\right), \tag{30}$$

there is an (ε,δ)-differentially private algorithm $A : (\mathcal{X} \times \{-1,1\})^n \to \{-1,1\}^{\mathcal{X}}$, which, given n i.i.d. samples from any realizable distribution P, products a hypothesis $\hat{f}$ so that $\mathrm{err}_P(\hat{f}) \leqslant \eta$ with probability at least $1-\beta$.

Proof. We let the algorithm A be `PolyPriPropLearn` (Algorithm 3). To establish accuracy, note that the output $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ of `PolyPriLearn` computed in Step 1 of Algorithm 3 satisfies $\mathrm{err}_P(\hat{f}) \leqslant \eta/4$ with probability at least $1 - \beta/2$ over the algorithm and the samples, by Theorem 5.7. Then using $\alpha = \eta/8$ in Lemma 6.2, we get that for the set $\mathcal{H}$ produced in Step 2 of Algorithm 3, there is some $h^\star \in \mathcal{H}$ so that $\mathrm{err}_P(h^\star) \leqslant \eta/2$. Then by Lemma 6.3, the output $\hat{h}$ of `PolyPriPropLearn` satisfies $\mathrm{err}_P(\hat{h}) \leqslant \eta$ with probability at least $1 - \beta$.

The (ε, δ)-differential privacy of the output $\hat{h}$ of `PolyPriPropLearn` follows from the (ε, δ)-differential privacy of the output $\hat{f} = \mathrm{SOA}_{\mathcal{G}}$ of `PolyPriLearn` with respect to its input samples, the post-processing property of differential privacy, and the $(\varepsilon, 0)$-differential privacy of `GenericLearner` with respect to its input samples. (Note that `PolyPriLearn` and `GenericLearner` are run on different samples.)

Finally, to see that the claimed upper bound on sample complexity holds, it suffices to upper bound the number of samples used by `GenericLearner` by the quantity in (30). This follows since by Lemma 6.2, we have

$$\log|\mathcal{H}| \leqslant \log\left(O\left(\frac{\mathrm{VCdim}^\star(\mathcal{F})}{\eta^2}\right)\right) \leqslant O\left(\mathrm{VCdim}(\mathcal{F}) + \log 1/\eta\right).$$

(Here we use that for any hypothesis class $\mathcal{F}$, $\mathrm{VCdim}^\star(\mathcal{F}) \leqslant 2^{\mathrm{VCdim}(\mathcal{F})+1}$ [Ass83].) □

As a corollary of Theorem 6.4 and [BNS15, Theorem 4.16] (or [ABMS20, Theorem 2.4], which is a more general result) we get a sample complexity bound for agnostic private proper PAC learning:

Corollary 6.5 (Agnostic private proper PAC learning)**.** *Let $\mathcal{F}$ be a concept class of domain $\mathcal{X}$ with $d_{\mathrm{V}} := \mathrm{VCdim}(\mathcal{F}), d_{\mathrm{L}} := \mathrm{Ldim}(\mathcal{F})$. For any $\varepsilon, \delta, \eta, \beta \in (0,1)$, for some*

$$n = O\left(\frac{d_{\mathrm{L}}^5 d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}}}{\varepsilon\delta\eta\beta}\right)}{\varepsilon\eta^2}\right),$$

there is an (ε, δ)-differentially private algorithm $A : (\mathcal{X} \times \{-1,1\})^n \to \mathcal{F}$, which, given n i.i.d. samples from any distribution P, produces a hypothesis $\hat{f} \in \mathcal{F}$ so that $\mathrm{err}_P(\hat{f}) \leqslant \eta + \inf_{f\in\mathcal{F}} \mathrm{err}_P(f)$ with probability at least $1 - \beta$.

6.2 Application to private data sanitization

In this section we show how to prove Corollaries 1.2 and 1.3 using a result of [BLM20a] that shows how to convert a private proper agnostic PAC learner into a sanitizer for a binary hypothesis class. We say that an algorithm A is an (α, β)-*accurate* proper agnostic PAC learner for a class $\mathcal{F}$ with *sample complexity* n if for any distribution P over $\mathcal{X} \times \{-1,1\}$, when given as input n i.i.d. samples from P, the algorithm A produces as output a function $\hat{f} \in \mathcal{F}$ so that with probability at least $1 - \beta$ over the sample and the randomness in A, we have $\mathrm{err}_P(\hat{f}) \leqslant \alpha + \inf_{f\in\mathcal{F}} \mathrm{err}_P(f)$.

Theorem 6.6 (Slight strengthening of [BLM20a], Propositions 1 & 2)**.** *Suppose $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ is a class of VC dimension d_{V} and dual Littlestone dimension $d_{\mathrm{L}}^\star$. Moreover suppose that for any $\alpha', \beta', \varepsilon', \delta' \in (0,1)$, there is some $n_0(\alpha', \beta', \varepsilon', \delta') \in \mathbb{N}$ so that $\mathcal{F}$ has a proper PAC learner with sample complexity $n_0(\alpha', \beta', \varepsilon', \delta')$ that is (ε', δ')-differentially private and (α', β')-accurate. Then there is a (sufficiently large) constant $C > 0$ so that for any $\alpha, \beta, \varepsilon, \delta \in (0,1)$, as long as $n \in \mathbb{N}$ is chosen to satisfy*

$$n \geqslant \frac{C}{\varepsilon} \cdot \left(\left(n_0(\alpha/8, \tau_0\beta/2, 1, \delta) + \frac{\log\left(\frac{d_{\mathrm{L}}^\star}{\beta\alpha}\right)}{\alpha}\right) \cdot \left(\frac{d_{\mathrm{L}}^\star \log(d_{\mathrm{L}}^\star/\alpha)\log(1/\delta)}{\alpha^2}\right)^{1/2}\right), \tag{31}$$

where $\tau_0 = \frac{\alpha^2}{Cd^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha)\log(1/\delta)}$, $\mathcal{F}$ *has a* $\left(n, \alpha, \beta, 1, \delta \cdot \frac{\sqrt{Cd^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha)}}{\alpha}\right)$*-sanitizer.*

We explain how to derive Theorem 6.6 using the proof technique in [BLM20a, Propositions 1 & 2] in Section C.

As an immediate corollary of Theorem 6.6 and Corollary 6.5 we obtain the following:

Corollary 6.7 (Private sanitization; formal version of Corollary 1.2)**.** *Let* $\mathcal{F}$ *be a hypothesis class with VC dimension* d_{V}*, Littlestone dimension* d_{L}*, and dual Littlestone dimension* $d^\star_{\mathrm{L}}$*. For any* $\alpha, \beta, \varepsilon, \delta \in (0,1)$*, for any* $n \in \mathbb{N}$ *satisfying*

$$n \geqslant C \cdot \frac{d^5_{\mathrm{L}} d_{\mathrm{V}} \sqrt{d^\star_{\mathrm{L}}} \log^2\left(\frac{d_{\mathrm{L}} d^\star_{\mathrm{L}}}{\delta\alpha\beta}\right) \log\left(\frac{d^\star_{\mathrm{L}}}{\alpha\delta}\right)}{\alpha^3 \varepsilon}, \tag{32}$$

$\mathcal{F}$ *has a* $(n, \alpha, \beta, \varepsilon, \delta)$*-sanitizer.*

We remark that the dependence of (32) on $d^\star_{\mathrm{L}}$, namely $\tilde{O}(\sqrt{d^\star_{\mathrm{L}}})$, is tight up to polylogarithmic factors in the sense that for all $d^\star_{\mathrm{L}}$, there is a class $\mathcal{F}$ with $\max\{\mathrm{VCdim}(\mathcal{F}), \mathrm{Ldim}(\mathcal{F})\} \leqslant O(\log d^\star_{\mathrm{L}})$ and $\mathrm{Ldim}^\star(\mathcal{F}) = d^\star_{\mathrm{L}}$, yet the sample complexity of sanitization for $\mathcal{F}$ is $\tilde{\Omega}(\sqrt{d^\star_{\mathrm{L}}})$, by Theorem 6.8 below.

Proof of Corollary 6.7. By Corollary 6.5, the following holds, for a sufficiently large constant $C > 0$: for any $\alpha', \beta', \varepsilon', \delta' \in (0,1)$, for any $n_0 \geqslant \frac{Cd^5_{\mathrm{L}} d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}}}{\varepsilon'\delta'\alpha'\beta'}\right)}{\varepsilon'(\alpha')^2}$, $\mathcal{F}$ has a proper agnostic PAC learner with sample complexity n_0 that is (ε', δ')-differentially private and (α', β')-accurate. We first show that for any $\alpha, \beta, \delta \in (0,1)$, $\mathcal{F}$ has a $(n, \alpha, \beta, 1, \delta)$-sanitizer for an appropriate value of n. To do this, we apply Theorem 6.6. To ensure that the number of samples n is at least the quantity in (31), it suffices to have at least

$$\begin{aligned} &C \cdot \left(\frac{d^5_{\mathrm{L}} d_{\mathrm{V}} \log^2\left(\frac{d_{\mathrm{L}} d^\star_{\mathrm{L}}}{\delta\alpha\beta}\right)}{\alpha^2} + \frac{\log\left(\frac{d^\star_{\mathrm{L}}}{\beta\alpha}\right)}{\alpha}\right) \cdot \frac{\sqrt{d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha) \log(d^\star_{\mathrm{L}}/(\alpha\delta))}}{\alpha} \\ \leqslant &C \cdot \frac{d^5_{\mathrm{L}} d_{\mathrm{V}} \sqrt{d^\star_{\mathrm{L}}} \log^2\left(\frac{d_{\mathrm{L}} d^\star_{\mathrm{L}}}{\delta\alpha\beta}\right) \log\left(\frac{d^\star_{\mathrm{L}}}{\alpha\delta}\right)}{\alpha^3} \end{aligned}$$

samples, where C is a sufficiently large constant.

The existence of a $(n, \alpha, \beta, \varepsilon, \delta)$-sanitizer for $\mathcal{F}$ for any $\alpha, \beta, \varepsilon, \delta \in (0,1)$ and n satisfying (32) now follows from Theorem 2.1 and a standard privacy amplification by subsampling argument [BNSV15, Lemma 4.12]:[11] in particular, by increasing the number of samples n by a factor of $O(1/\varepsilon)$ and sampling an $O(\varepsilon)$ fraction of the samples, we can convert a $(O(1), \delta)$-differentially private algorithm into an (ε, δ)-differentially private algorithm. The accuracy loss due to this subsampling can be bounded by a small constant times α, by Theorem 2.1 and the fact that the number of samples n in (32) must be at least $\Omega\left(\frac{\mathrm{VCdim}(\mathcal{F}) + \log 1/\beta}{\alpha^2}\right)$. □

Finally, we may prove Corollary 1.3:

Proof of Corollary 1.3. The fact that finite Littlestone dimension of a class $\mathcal{F}$ implies sanitizability follows from the fact that for all binary hypothesis classes $\mathcal{F}$, $\mathrm{VCdim}(\mathcal{F}) \leqslant \mathrm{Ldim}(\mathcal{F})$, $\mathrm{Ldim}^\star(\mathcal{F}) \leqslant 2^{2^{\mathrm{Ldim}(\mathcal{F})+2}}$ [BLM20a, Lemma 4], and Corollary 6.7. For the opposite direction,

[11] Similar arguments have been used in, e.g., [BST14, Lemma 2.2], [BBKN14], [BLM20b].

we use the fact that for any $\mathcal{F}$, the threshold dimension of $\mathcal{F}$[12], denoted $\mathrm{Tdim}(\mathcal{F})$, satisfies $\mathrm{Tdim}(\mathcal{F}) \geqslant \lfloor \log \mathrm{Ldim}(\mathcal{F}) \rfloor$ [ALMM19, Theorem 3]. Thus any $(n, \alpha, \beta, \varepsilon, \delta)$-sanitizer for a class $\mathcal{F}$ yields a $(n, \alpha, \beta, \varepsilon, \delta)$-sanitizer for the class of thresholds on a linearly ordered domain of size T for any $T \leqslant \lfloor \log \mathrm{Ldim}(\mathcal{F}) \rfloor$. But [BNSV15, Theorem 1] yields that any $(n, 1/10, 1/10, 1/10, 1/(50n^2))$-sanitizer for the class of thresholds on a linearly ordered comain of size T must satisfy $n \geqslant \Omega(\log^\star T)$. Thus, if $\mathrm{Ldim}(\mathcal{F})$ is infinite, then there is no $(n, 1/10, 1/10, 1/10, 1/(50n^2))$-sanitizer for the class $\mathcal{F}$, and so $\mathcal{F}$ is not sanitizable. □

Lower bounds We end this section by discussing how the sample complexity bound of Corollary 6.7 compares to existing lower bounds for sanitization. First, we remark that it follows from fingerprinting-based lower bounds [BUV14] that in general the sample complexity of a sanitizer for a class $\mathcal{F}$ must grow at least polynomially in the the dual Littlestone dimension of $\mathcal{F}$:

Theorem 6.8 ([BUV14], Theorem 5.8)**.** *For any constant $\ell \in \mathbb{N}$, the following holds for all $d, t \in \mathbb{N}$ so that $\ell + 2 \leqslant t \leqslant d/2$. For $\mathcal{X} = \{-1, 1\}^d$, there is a class $\mathcal{F} \subset \{-1, 1\}^{\mathcal{X}}$ so that:*

- $\mathrm{Ldim}(\mathcal{F}) = \Theta(t \log(d/t))$*;*
- $\mathrm{Ldim}^\star(\mathcal{F}) = \Theta(d)$*,*

and so that for all $\varepsilon \in (0, 1)$ and $\alpha \geqslant \tilde{\Omega}\left(\frac{d^{-\ell/3+1/4}}{\sqrt{\varepsilon}}\right)$, any $(n, \alpha, 1/100, \varepsilon, 1/(10n))$-sanitizer for $\mathcal{F}$ must have

$$n \geqslant \tilde{\Omega}\left(\frac{t\sqrt{d}}{\varepsilon\alpha^2}\right).$$

For any fixed ℓ, the $\tilde{\Omega}(\cdot)$ in Theorem 6.8 hides factors which are inverse polynomial in $\log t, \log d, \log \frac{1}{\varepsilon}, \log \frac{1}{\alpha}$. We also remark that the VC and dual VC dimensions are within constant factors of the Littlestone and dual Littlestone dimensions of the class $\mathcal{F}$ of Theorem 6.8 (this will be clear from the proof below). Since [BUV14] does not explicitly compute the Littlestone and dual Littlestone dimensions of the class $\mathcal{F}$, we give a short proof that the entirety of the claim in Theorem 6.8 holds, using [BUV14, Theorem 5.8]:

Proof of Theorem 6.8 using [BUV14]. We take the class $\mathcal{F}$ to be the class of t-wise conjunctions on $\mathcal{X} = \{-1, 1\}^d$, i.e., the class of all ANDs of t literals on $\{-1, 1\}^d$ (for concreteness, view 1 as "False" and -1 as "True"). From [Lit87, Lemma 6] we have that $\mathrm{Ldim}(\mathcal{F}) \geqslant \mathrm{VCdim}(\mathcal{F}) \geqslant \Omega(t \log(d/t))$; also $\mathrm{Ldim}(\mathcal{F}) \leqslant O(t \log(d/t))$ since the size of $\mathcal{F}$ is bounded above by $2^{O(t \log(d/t))}$. For the dual quantity, it is clear that $\mathrm{Ldim}^\star(\mathcal{F}) \leqslant d$ since the size of the dual class is 2^d. Moreover, $\mathrm{Ldim}^\star(\mathcal{F}) \geqslant \mathrm{VCdim}^\star(\mathcal{F}) \geqslant d/2$ since the class of $d/2$ functions $x \mapsto x_1 \wedge \cdots \wedge x_{t-1} \wedge x_j$, for $d/2 \leqslant j \leqslant d$, is shattered by the dual class $\mathcal{X}$. Finally, [BUV14, Theorem 5.8] gives us the fact that there is no $(\varepsilon, 1/(10n))$-differentially private algorithm which takes as input a dataset S of size n and outputs some function $\mathrm{Est} : \mathcal{F} \to [0, 1]$ satisfying $|\mathrm{Est}(f) - \mathrm{err}_S(f)| \leqslant \alpha$ for all $f \in \mathcal{F}$ with probability at least $2/3$. □

By choosing $\ell = 1$, and arbitrary positive integers t, d tending to ∞ and satisfying $t \leqslant d/2$, Theorem 6.8 rules out a sample complexity bound for sanitization that depends polynomially on only the Littlestone dimension of $\mathcal{F}$ (such as one in Theorem 6.4 for proper private learning). Because of the requirement that $t \leqslant d/2$ in Theorem 6.8, it does not rule out a sample complexity bound

[12]The *threshold dimension* of $\mathcal{F} \subset \{-1, 1\}^{\mathcal{X}}$ is the largest positive integer T so that there are $x_1, \ldots, x_T \in \mathcal{X}$ and $f_1, \ldots, f_T \in \mathcal{F}$ so that for $1 \leqslant i, j \leqslant t$, $f_i(x_j) = \begin{cases} 1 : i \geqslant j \\ 0 : i < j \end{cases}$.

that depends polynomially on only the dual Littlestone dimension (and only sub-polynomially on the Littlestone dimension). This latter possibility is ruled out by discrepancy-based lower bounds:

Theorem 6.9 ([NTZ12]). *For any binary hypothesis class $\mathcal{F}$, any $\alpha < 1/50$ and any $\varepsilon \in (0,1)$, any $(n, \alpha, 1/100, \varepsilon, 0.1)$-sanitizer for $\mathcal{F}$ must have $n \geqslant \Omega\left(\frac{\mathrm{VCdim}(\mathcal{F})}{\varepsilon\alpha}\right)$.*

For a proof of the precise statement of Theorem 6.9, see Theorem 5.8 and Proposition 5.11 of [Vad17]. Note that for any positive integer d, there is a class $\mathcal{F}$ for which $\mathrm{Ldim}(\mathcal{F}) = \mathrm{VCdim}(\mathcal{F}) = d$ and $\mathrm{Ldim}^\star(\mathcal{F}) = \Theta(\log d)$ (for instance, we may take the class of all functions on d distinct points). Thus Theorem 6.9 rules out the existence of a sanitizer with sample complexity polynomial in only dual Littlestone dimension.

Summarizing, from Theorems 6.8 and 6.9, we obtain that Corollary 6.7 is "best possible up to a polynomial" in the sense that polynomial dependence on both d_{L} *and* $d_{\mathrm{L}}^\star$ is necessary in a worst-case sense. Moreover, when $d_{\mathrm{L}}, d_{\mathrm{L}}^\star$ are of the same order, then any sample complexity upper bound must be superlinear $\max\{d_{\mathrm{L}}, d_{\mathrm{L}}^\star\}$ (Theorem 6.8). Finally, in light of Theorem 6.8, the square-root dependence on $d_{\mathrm{L}}^\star$ (up to polylogarithmic factors) in Corollary 6.7 is best possible up to polylogarithmic factors.

7 Conclusions

In this paper we showed that it is possible to privately and properly learn binary hypothesis classes of Littlestone dimension d with sample complexity polynomial in d. As a corollary we showed that such classes have sanitizers with sample complexity polynomial in d and the dual Littlestone dimension $d^\star$. A central open question remaining (see, e.g., [BNS19, Section 1.6]) is to determine a characterization of the sample complexity of (proper and improper) PAC learning with approximate differential privacy, up to (ideally) a constant factor, much like the VC dimension provides such a characterization for (non-private) PAC learning [Vap98], the Littlestone dimension provides such a characterization for online learning [Lit87, BPS09], and the probabilistic representation dimension [BNS19] and the one-way public coin communication complexity [FX14] both provide such a characterization for improper PAC learning with pure differential privacy. As noted by [ABMS20], current lower bounds even allow for the possibility that the sample complexity of (proper or improper) PAC learning with approximate differential privacy is linear in $\mathrm{VCdim}(\mathcal{F}) + \log^\star(\mathrm{Ldim}(\mathcal{F}))$. Below we list some intermediate questions which may be useful in attacking this question and the related question of characterizing the sample complexity of sanitization. (Throughout by "private" we mean (ε, δ)-differentially private with δ negligible in the number of users n.)

1. **Sample complexity linear in Littlestone dimension.** The most immediate open question is to reduce the exponent of d from the current value of 6 in Theorem 1.1. In particular, one could hope for sample complexity that scales linearly with the Littlestone dimension d (see the discussion following Theorem 1.1).
2. **Polynomial characterization of private learnability.** One could also attempt to show bounds with sublinear dependence on the Littlestone dimension, as long as there is at least linear dependence on the VC dimension. Rather optimistically, we ask: is the sample complexity of (properly or improperly) PAC learning a class $\mathcal{F}$ with (ε, δ)-differential privacy at most $n = \mathrm{poly}\left(\mathrm{VCdim}(\mathcal{F}), \log^\star(\mathrm{Ldim}(\mathcal{F}))\right)$? (Here we omit dependence on $1/\alpha, 1/\varepsilon, \log 1/\delta$, for which the dependence should be polynomial as well.) In light of the lower bound of $\Omega(\mathrm{VCdim}(\mathcal{F}) + \log^\star(\mathrm{Ldim}(\mathcal{F})))$ by Alon et al. [ALMM19] on the sample complexity, this would give a characterization for the sample complexity of private PAC learning up to a polynomial factor.

3. **Proper vs. improper learning.** Is there a family of hypothesis classes for which the sample complexity of proper private learning is asymptotically larger than the sample complexity of improper private learning? The answer to this question is "yes" for the case of pure privacy (e.g., exhibited by the class of point functions [BBKN14]), but it remains open for approximate privacy to the best of our knowledge.
4. **Direct proof of Corollary 6.7.** The current proof of Corollary 6.7 is quite long: it consists of first proving the existence of an improper private learner (Theorem 5.7), then showing how to make it proper (Corollary 6.5), and finally applying Theorem 6.6 of Bousquet et al. [BLM20a], which itself has two fairly involved parts, the first of which shows that $\mathcal{F}$ is "Sequentially-Foolable" [BLM20a, Theorem 2], and the second of which shows that $\mathcal{F}$ is sanitizable [BLM20a, Theorem 1]. It would be interesting to find a more direct proof of Corollary 6.7, namely one that does not "go through" a proper learner.
5. **Improved bounds for sanitization.** Finally, it would be interesting to improve quantitatively upon the upper bound for sanitization of Corollary 6.7. In particular, analogously to item 2, it is natural to ask: is the sample complexity of sanitizating a class $\mathcal{F}$ (with approximate privacy) at most $n = \text{poly}\left(\text{VCdim}(\mathcal{F}), \text{VCdim}^\star(\mathcal{F}), \log^\star(\text{Ldim}(\mathcal{F}))\right)$? By [BUV14, Corollary 3.6], Theorem 6.9, and [BNSV15, Theorems 3.2 & 4.6], the sample complexity of sanitization is at least $\tilde{\Omega}\left(\text{VCdim}(\mathcal{F}) + \sqrt{\text{VCdim}^\star(\mathcal{F})} + \log^\star(\text{Ldim}(\mathcal{F}))\right)$,[13] so this would provide a characterization for the sample complexity of sanitization up to a polynomial factor. Since our approach of using the results of [BLM20a] seems to necessarily incur at least a *square-root* dependence on the dual Littlestone dimension $\text{Ldim}^\star(\mathcal{F})$, any positive answer to this question would likely involve a positive answer to the question in item 4.

A Private proper learner for infinite $\mathcal{F}$ and $\mathcal{X}$

In this section we extend the arguments from Section 6 to cover the case where $\mathcal{X}, \mathcal{F}$ are allowed to be countably infinite. The techniques closely follow those in [BLM20a].

A.1 Preliminaries

Product topology Let $\mathcal{V}$ be an arbitrary set, and let $\{-1,1\}$ have the discrete topology. The *product topology* on the space $\{-1,1\}^{\mathcal{V}}$ of functions $f : \mathcal{V} \to \{-1,1\}$ is defined to be the coarsest topology so that the functions $\pi_v : \{-1,1\}^{\mathcal{V}} \to \{-1,1\}$, defined by $\pi_v(f) := f(v)$ are all continuous. It is known that this topology is Hausdorff. The following fact is an immediate consequence of Tychanoff's theorem:

Theorem A.1 (Tychanoff's theorem; e.g., [Mun00], Chapter 5, Theorem 1.1)**.** *The space* $\{-1,1\}^{\mathcal{V}}$ *is compact (under the product topology).*

Compactness Let $\mathcal{W}$ be a compact Hausdorff topological space. Let $C(\mathcal{W})$ denote the space of real-valued continuous functions on $\mathcal{W}$. Let $\mathscr{B}(\mathcal{W})$ denote the space of Borel measures on $\mathcal{W}$, and let $\Delta(\mathcal{W})$ denote the space of Borel *probability* measures on $\mathcal{W}$ (a measure μ on $\mathcal{W}$ is a probability

[13] [BUV14, Corollary 3.6] gives a lower bound of $\tilde{\Omega}(\sqrt{d})$ on the sample complexity of private release of 1-way marginals on $\{-1,1\}^d$; the $\tilde{\Omega}(\text{VCdim}^\star(\mathcal{F}))$ lower bound on the sample complexity of sanitization in any class $\mathcal{F}$ follows since a class of 1-way marginals on a copy of $\{-1,1\}^{\text{VCdim}^\star(\mathcal{F})}$ may be embedded in any class $\mathcal{F}$. Similarly, [BNSV15] gives a $\Omega(\log^\star|\mathcal{X}|)$ lower bound on the sample complexity of release of threshold functions on a domain $\mathcal{X}$; the $\Omega(\log^\star|\mathcal{X}|)$ lower bound on the sample complexity of sanitization in any class $\mathcal{F}$ follows since $\log \text{Ldim}(\mathcal{F})$ thresholds may be embedded in $\mathcal{F}$.

measure if for all measurable subsets $A \subset \mathcal{W}$, $\mu(A) \in [0,1]$, and $\mu(\mathcal{W}) = 1$). The weak* topology on $\mathscr{B}(\mathcal{W})$ (also known as the *vague* topology) is defined to be the coarsest topology so that all of the mappings $\mu \mapsto \int_{f \in \mathcal{W}} \omega(f) d\mu(f)$, where $\omega \in C(\mathcal{W})$, are continuous. The following lemma is a consequence of the Banach-Alaoglu theorem (see, e.g., [RS81, Theorem IV.21]) and the Riesz-Markov theorem which states that the dual space of the Banach space $C(\mathcal{W})$ is the space $\mathscr{B}(\mathcal{W})$ of Borel measures on $\mathcal{W}$ (see, e.g., [RS81, Theorem IV.14], and also [BLM20a, Claim 2]):

Lemma A.2. *The space $\Delta(\mathcal{W})$ is compact in the weak* topology.*

Spaces of distributions Next, recall that Σ is a σ-algebra on the data space $\mathcal{X}$. We consider the product topology on the space $\{-1,1\}^{\mathcal{X}}$; by Tychonoff's theorem (Theorem A.1), $\{-1,1\}^{\mathcal{X}}$ is compact (and Hausdorff). Let $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$ have the subspace topology, so that $\mathcal{F}$ is also compact. By Lemma A.2, $\Delta(\mathcal{F})$ is compact in the weak* topology.

Following [BLM20a], let $\mathbb{R}^{\mathcal{X}}_{\text{fin}}$ to be the space of real-valued functions $p : \mathcal{X} \to \mathbb{R}$ so that there are only finitely many $x \in \mathcal{X}$ so that $p(x) \neq 0$. Give $\mathbb{R}^{\mathcal{X}}_{\text{fin}}$ the topology induced by the ℓ_1 norm; more formally, a basis of open sets is given by the balls $\mathcal{B}_{q,a}$, for $q \in \mathbb{R}^{\mathcal{X}}_{\text{fin}}$, $a > 0$, where:

$$\mathcal{B}_{q,a} := \left\{ p \in \mathbb{R}^{\mathcal{X}}_{\text{fin}} : \sum_{x \in \mathcal{X}} |p(x) - q(x)| < a \right\}.$$

Let $\Delta_{\text{fin}}(\mathcal{X})$ be the subspace of $\mathbb{R}^{\mathcal{X}}_{\text{fin}}$ consisting of functions p so that for all $x \in \mathcal{X}$, $p(x) \geqslant 0$ and $\sum_{x \in \mathcal{X}} p(x) = 1$. We will often identify $\Delta_{\text{fin}}(\mathcal{X})$ with the space of probability measures on $\mathcal{X}$ with *finite support*. In particular, for some $p : \mathcal{X} \to \mathbb{R}$, the corresponding measure P is the one defined by, for $A \in \Sigma$,

$$P(A) = \sum_{x \in \mathcal{X}} p(x) \cdot \delta_x(A) = \sum_{x \in \mathcal{X}} p(x) \cdot \mathbb{1}[x \in A].$$

Semi-continuity, Sion's minimax theorem Let $\mathcal{W}$ be a topological space. A function $f : \mathcal{W} \to \mathbb{R}$ is *upper semi-continuous (u.s.c)* if for every $r \in \mathbb{R}$, the set $\{w : f(w) \geqslant r\}$ is closed. Similarly, f is *lower semi-continuous (l.s.c)* if for every $r \in \mathbb{R}$, the set $\{w : f(w) \leqslant r\}$ is closed. We will use the following fact:

Lemma A.3 ([BLM20a], Claim 3)**.** *Let $\mathcal{W}$ be a compact hausdorff space, and let $\mathcal{K} \subset \mathcal{W}$ be a closed subset. Consider the mapping $T_{\mathcal{K}} : \Delta(\mathcal{W}) \to [0,1]$, defined by $T_{\mathcal{K}}(\mu) := \mu(\mathcal{K})$. Then $T_{\mathcal{K}}$ is u.s.c. with respect to the weak* topology on $\Delta(\mathcal{W})$.*

Sion's minimax theorem, stated below, is a generalization of the von Neumann minimax theorem.

Theorem A.4 ([Sio58])**.** *Let $\mathcal{W}$ be a compact and convex subset of a topological vector space and $\mathcal{U}$ be a convex subset of a topological vector space. Suppose $F : \mathcal{W} \times \mathcal{U} \to \mathbb{R}$ is a real-valued function so that:*

- *For all $u \in \mathcal{U}$, the function $w \mapsto F(w,u)$ is l.s.c. and convex on $\mathcal{W}$.*
- *For all $w \in \mathcal{W}$, the function $u \mapsto F(w,u)$ is u.s.c. and concave on $\mathcal{U}$.*

Then

$$\inf_{w \in \mathcal{W}} \sup_{u \in \mathcal{U}} F(w,u) = \sup_{u \in \mathcal{U}} \inf_{w \in \mathcal{W}} F(w,u).$$

A.2 Modifications to the finite case

In this section we detail the modifications that it is necessary to make to the proofs in Section 6 to establish Theorem 6.4 (and thus Corollary 6.5) for the case that $\mathcal{X}, \mathcal{F}$ are countably infinite.

We begin with Lemma 6.1; notice that nowhere in the proof of Lemma 6.1 do we use that $\mathcal{X}, \mathcal{F}$ are finite; i.e., it holds if $\mathcal{X}, \mathcal{F}$ are allowed to be infinite. Corollary A.5 is then an immediate corollary of Lemma 6.1 (with infinite $\mathcal{X}, \mathcal{F}$), since $\Delta_{\text{fin}}(\mathcal{X}) \subset \Delta(\mathcal{X})$.

Corollary A.5. *There is a constant $C > 0$ so that the following holds. Fix any $\alpha \in (0,1)$ and $\mathcal{G} \subset \mathcal{F}$ which is $\left\lceil \frac{C(d+\log 1/\alpha)}{\alpha^2} \right\rceil$-irreducible and suppose $\mathrm{VCdim}(\mathcal{F}) \leqslant d$ for some $d \in \mathbb{N}$. Then it holds that*

$$\sup_{P \in \Delta_{\text{fin}}(\mathcal{X})} \inf_{D \in \Delta(\mathcal{F})} \mathbb{E}_{x \sim P, h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right] \leqslant \alpha. \tag{33}$$

Lemma A.6 is a generalization of Lemma 6.2 to the case that $\mathcal{X}, \mathcal{F}$ are infinite; the main technical portion of the proof departing from that of Lemma 6.2 is the verification that the preconditions of Sion's minimax theorem hold.

Lemma A.6. *There is a constant $C > 0$ so that the following holds. Fix any $\alpha \in (0,1)$ and $\mathcal{G} \subset \mathcal{F}$ which is $\left\lceil \frac{C(d+\log 1/\alpha)}{\alpha^2} \right\rceil$-irreducible and suppose $\mathrm{VCdim}(\mathcal{F}) \leqslant d$ and $\mathrm{VCdim}^\star(\mathcal{F}) \leqslant d^\star$ for some $d, d^\star \in \mathbb{N}$. Then there is a set $\mathcal{H} \subset \mathcal{F}$, depending only on the function $\mathrm{SOA}_{\mathcal{G}} : \mathcal{X} \to \{-1, 1\}$, and of size $|\mathcal{H}| \leqslant \left\lceil \frac{C(d^\star+\log 1/\alpha)}{\alpha^2} \right\rceil$, so that for any distribution $P \in \Delta(\mathcal{X})$, it holds that*

$$\min_{h \in \mathcal{H}} \mathbb{E}_{X \sim P} \left[\mathbb{1}[h(X) \neq \mathrm{SOA}_{\mathcal{G}}(X)] \right] \leqslant 3\alpha. \tag{34}$$

Proof. We will first use Sion's minimax theorem (Theorem A.4) to argue that

$$\inf_{D \in \Delta(\mathcal{F})} \sup_{P \in \Delta_{\text{fin}}(\mathcal{X})} \mathbb{E}_{x \sim P, h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right] \tag{35}$$

$$= \sup_{P \in \Delta_{\text{fin}}(\mathcal{X})} \inf_{D \in \Delta(\mathcal{F})} \mathbb{E}_{x \sim P, h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right] \leqslant \alpha. \tag{36}$$

(Notice that the inequality in (36) is from Corollary A.5; below we argue that the equality in the above display.) In particular, we will have $\mathcal{W} = \Delta(\mathcal{F}), \mathcal{U} = \Delta_{\text{fin}}(\mathcal{X})$, and for $D \in \mathcal{W}, P \in \mathcal{U}$,

$$F(D, P) := \mathbb{E}_{x \sim P, h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right].$$

Notice that $\mathcal{W}, \mathcal{U}$ are subsets of the topological vector spaces $\mathscr{B}(\mathcal{F}), \mathbb{R}^{\mathcal{X}}_{\text{fin}}$, respectively. Moreover, it is immediate that both $\mathcal{W}, \mathcal{U}$ are convex, and by Theorem A.1 and Lemma A.2 we have that $\mathcal{W}$ is compact. To check the u.s.c. and l.s.c. preconditions of Theorem A.4, we argue as follows:

- Fix any $D \in \mathcal{W}$. Notice that the function $P \mapsto F(D, P)$ may be written as

  $$F(D, P) = \sum_{x \in \mathcal{X}} P(x) \cdot \mathbb{E}_{h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right].$$

  It is evident that $P \mapsto F(D, P)$ is a linear function, hence convex. Moreover, it is continuous (hence l.s.c.) since for each $x \in \mathcal{X}$, $\left| \mathbb{E}_{h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right] \right| \leqslant 1$ and since the topology on $\mathcal{W}$ is induced by the ℓ_1 norm on $\mathbb{R}^{\mathcal{X}}_{\text{fin}}$.
- Fix any $P \in \mathcal{U}$. It is evident that $D \mapsto F(D, P)$ is a linear function, hence concave. Note that for any $x \in \mathcal{X}$, by definition of the product topology, the map from $\mathscr{B}(\mathcal{F})$ to $\mathbb{R}$ that sends $h \mapsto \mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]$ is continuous. Thus $\{h \in \mathcal{F} : \mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)\}$ is a closed subset of $\mathcal{F}$. By Lemma A.3, the mapping $D \mapsto \mathbb{E}_{h \sim D} \left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)] \right]$ is u.s.c. with respect to the weak* topology on $\Delta(\mathcal{F})$. That $D \mapsto F(D, P)$ is u.s.c. follows since a finite sum of u.s.c. functions is u.s.c.

We have verified that all of the conditions of Theorem A.4 hold, and thus we may conclude that the equality (35) holds.

Fix any $P \in \Delta(\mathcal{X})$. Since $\mathrm{VCdim}(\mathcal{F})$ is finite, by Theorem 2.1, there is some $P' \in \Delta_{\mathrm{fin}}(\mathcal{X})$ so that

$$\sup_{h \in \mathcal{F}} |\mathbb{E}_{x \sim P}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] - \mathbb{E}_{x \sim P'}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right]| \leqslant \alpha. \tag{37}$$

Fix an arbitrary $\alpha' > 0$. Fix some $D \in \Delta(\mathcal{F})$ obtaining a value of at most $\alpha + \alpha'$ in (35). Note that D depends only on $\mathrm{SOA}_{\mathcal{G}} \in \{0,1\}^{\mathcal{X}}$, i.e., it can be written as a function of $\mathrm{SOA}_{\mathcal{G}}$. By Theorem 2.1, for a sufficiently large $C > 0$, for $m = \left\lceil \frac{C(d^\star + \log 1/\alpha)}{\alpha^2} \right\rceil$, then with probability at least $1/2$ over an i.i.d. sample $h_1, \ldots, h_m \sim D$, we have that

$$\begin{aligned} &\sup_{x \in \mathcal{X}} \left| \mathbb{E}_{h \sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h_j(x)] \right| \\ =&\sup_{x \in \mathcal{X}} \left| \mathbb{E}_{h \sim D}\left[\mathbb{1}[h(x) \neq 1]\right] - \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[h_j(x) \neq 1] \right| \leqslant \alpha. \end{aligned} \tag{38}$$

Fix any $h_1, \ldots, h_m$ so that (28) holds, and set $\mathcal{H} := \{h_1, \ldots, h_m\}$. Write $h \sim_U \mathcal{H}$ to mean that h is drawn uniformly from $\mathcal{H}$. Then by (35), (37), and (38), we have, for the given $P \in \Delta(\mathcal{X})$,

$$\begin{aligned} &\mathbb{E}_{x \sim P, h \sim_U \mathcal{H}}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] \\ \leqslant^{(37)} &\mathbb{E}_{x \sim P', h \sim_U \mathcal{H}}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] + \alpha \\ \leqslant^{(38)} &\mathbb{E}_{x \sim P, h \sim D}\left[\mathbb{1}[\mathrm{SOA}_{\mathcal{G}}(x) \neq h(x)]\right] + 2\alpha \\ \leqslant^{(35)\&(36)} &3\alpha + \alpha'. \end{aligned}$$

Since $\alpha' > 0$ is arbitrary, (34) is an immediate consequence of the above. □

No modifications to the algorithm `PolyPriPropLearn` (Algorithm 3) are necessary to deal with the case of infinite $\mathcal{X}, \mathcal{F}$, except the reference to Lemma 6.2 on Step 2 should instead be to Lemma A.6. That Theorem 6.4 holds for the case that $\mathcal{X}, \mathcal{F}$ are countably infinite follows without any modifications to its proof.

B On the Littlestone dimension of SOA classes

Given a class $\mathcal{F}$, the algorithm `PolyPriLearn` (Algorithm 2) will output with high probability a hypothesis of the form $\mathrm{SOA}_{\mathcal{G}}$ for some $\mathcal{G} \subset \mathcal{F}$. To address the question of whether this hypothesis has small population error, we used Lemma 4.4 to upper bound the Littlestone dimension (and thus VC dimension) of the class of hypotheses $\mathrm{SOA}_{\mathcal{G}}$ for which $\mathcal{G} \subset \mathcal{F}$ is $(d+1)$-irreducible, where d is the Littlestone dimension of $\mathcal{F}$. In this section, we show that one cannot drop the requirement that $\mathcal{G}$ be $(d+1)$-irreducible; in particular, the VC dimension of

$$\tilde{\mathcal{F}} := \{\mathrm{SOA}_{\mathcal{G}} : \mathcal{G} \subset \mathcal{F}\} \tag{39}$$

can be infinite even if $\mathrm{VCdim}(\mathcal{F})$ is finite.

Let $\mathcal{F}^{\text{neg-pt}}$ be the class of point functions and negated point functions on an infinite set $\mathcal{X}$. In particular, for $x \in \mathcal{X}$, write δ_x to be the point function for x, defined by $\delta_x(y) = 1$ if $y = x$ and else $\delta_x(y) = -1$. Then:

$$\mathcal{F}^{\text{neg-pt}} := \{\delta_x : x \in \mathcal{X}\} \cup \{-\delta_x : x \in \mathcal{X}\}.$$

It is straightforward to check that $\mathrm{Ldim}(\mathcal{F}^{\text{neg-pt}}) = \mathrm{VCdim}(\mathcal{F}^{\text{neg-pt}}) = 3$. However, the Littlestone dimension of the class $\tilde{\mathcal{F}}^{\text{neg-pt}}$ defined in (39) is infinite, as shown in the following proposition:

Proposition B.1. *It holds that* $\mathrm{VCdim}(\tilde{\mathcal{F}}^{\text{neg-pt}}) = \mathrm{Ldim}(\tilde{\mathcal{F}}^{\text{neg-pt}}) = \infty$.

Proof. We show that for any $d \in \mathbb{N}$, $d \geqslant 2$, and distinct points $x_1, \ldots, x_d \in \mathcal{X}$, there is some $h \in \tilde{\mathcal{F}}^{\text{neg-pt}}$ so that $h(x_1) = \cdots = h(x_d) = 1$ and $h(x) = -1$ for all $x \notin \{x_1, \ldots, x_d\}$.

To do so, fix $d \geqslant 2$ and the points $x_1, \ldots, x_d$. Define

$$\mathcal{G} := \{f \in \mathcal{F}^{\text{neg-pt}} : \exists j \in [d] \text{ s.t. } f(x_j) = 1\}.$$

We first show that for all $x \notin \{x_1, \ldots, x_d\}$, it holds that $\mathrm{SOA}_{\mathcal{G}}(x) = -1$. This in turn follows from the following two facts:

- $\mathrm{Ldim}(\mathcal{G}|_{(x,1)}) = 1$. To see this, note first that any $f \in \mathcal{G}|_{(x,1)}$ must be of the form $f(y) = -\delta_z(y)$ for $z, y \in \mathcal{X}$. The class of such f has Littlestone dimension at most 1. The Littlestone dimension is exactly 1 since $-\delta_{x_1}, -\delta_{x_2} \in \mathcal{G}|_{(x,1)}$.
- $\mathrm{Ldim}(\mathcal{G}|_{(x,-1)}) = 2$. To see that the Littlestone dimension is at most 2, note that $\mathrm{Ldim}(\mathcal{F}^{\text{neg-pt}}|_{(x,-1)}) = 2$ and $\mathcal{G} \subset \mathcal{F}^{\text{neg-pt}}$. To see that the Littlestone dimension is at least 2, consider the tree $\mathbf{x}$ of depth 2 defined by
$$\mathbf{x}_1 = x_1, \qquad \mathbf{x}_2(-1) = \mathbf{x}_2(1) = x_2.$$
This tree is shattered by $\mathcal{G}|_{(x,-1)}$ since:
$$\begin{aligned} \delta_{x_3}(x_1) = \delta_{x_3}(x_1) &= -1 \\ \delta_{x_1}(x_1) = 1, \delta_{x_2}(x_2) &= -1 \\ \delta_{x_2}(x_1) = -1, \delta_{x_2}(x_2) &= 1 \\ -\delta_x(x_1) = -\delta_x(x_2) &= 1, \end{aligned}$$
and $\delta_{x_1}, \delta_{x_2}, \delta_{x_3}, -\delta_x \in \mathcal{G}|_{(x,-1)}$.

We next show that for all $x \in \{x_1, \ldots, x_d\}$, it holds that $\mathrm{SOA}_{\mathcal{G}}(x) = 1$. This in turn follows from the following facts. We note that by symmetry we may assume without loss of generality that $x = x_1$:

- $\mathrm{Ldim}(\mathcal{G}|_{(x_1,1)}) = 2$. Since $\mathcal{G} \subset \mathcal{F}^{\text{neg-pt}}$, we have that $\mathcal{G}|_{(x_1,1)} \subset \mathcal{F}^{\text{neg-pt}}|_{(x_1,1)}$. On the other hand, any $f \in \mathcal{F}^{\text{neg-pt}}$ with $f(x_1) = 1$ necessarily lies in $\mathcal{G}$, so $\mathcal{F}^{\text{neg-pt}}|_{(x_1,1)} \subset \mathcal{G}|_{(x_1,1)}$. Thus $\mathcal{F}^{\text{neg-pt}}|_{(x_1,1)} = \mathcal{G}|_{(x_1,1)}$, so we have that $\mathrm{Ldim}(\mathcal{G}|_{(x_1,1)}) = \mathrm{Ldim}(\mathcal{F}^{\text{neg-pt}}|_{(x_1,1)}) = 2$.
- $\mathrm{Ldim}(\mathcal{G}|_{(x_1,-1)}) = 2$. The Littlestone dimension is at most 2 since $\mathrm{Ldim}(\mathcal{F}^{\text{neg-pt}}|_{(x_1,-1)}) = 2$. A similar argument as the one used above to establish that $\mathrm{Ldim}(\mathcal{G}|_{(x,-1)}) = 2$ may be used to show that here $\mathrm{Ldim}(\mathcal{G}|_{(x_1,-1)}) = 2$; however, we note that this argument isn't necessary to show that $\mathrm{SOA}_{\mathcal{G}}(x_1) = 1$.

We have shown that $\mathrm{SOA}_{\mathcal{G}}(x) = 1$ if and only if $x \in \{x_1, \ldots, x_d\}$, which completes the proof of the proposition. □

C Proof of Theorem 6.6

In this section we sketch how the quantitative bound in Theorem 6.6 (restated below for convenience) may be derived from the argument in [BLM20a].

Theorem 6.6 (Slight strengthening of [BLM20a], Propositions 1 & 2)**.** *Suppose* $\mathcal{F} \subset \{-1, 1\}^{\mathcal{X}}$ *is a class of VC dimension* d_{V} *and dual Littlestone dimension* $d_{\mathrm{L}}^{\star}$*. Moreover suppose that for any*

$\alpha', \beta', \varepsilon', \delta' \in (0,1)$, there is some $n_0(\alpha', \beta', \varepsilon', \delta') \in \mathbb{N}$ so that $\mathcal{F}$ has a proper PAC learner with sample complexity $n_0(\alpha', \beta', \varepsilon', \delta')$ that is (ε', δ')-differentially private and (α', β')-accurate. Then there is a (sufficiently large) constant $C > 0$ so that for any $\alpha, \beta, \varepsilon, \delta \in (0,1)$, as long as $n \in \mathbb{N}$ is chosen to satisfy

$$n \geqslant \frac{C}{\varepsilon} \cdot \left(\left(n_0(\alpha/8, \tau_0\beta/2, 1, \delta) + \frac{\log\left(\frac{d^\star_{\mathrm{L}}}{\beta\alpha}\right)}{\alpha} \right) \cdot \left(\frac{d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha) \log(1/\delta)}{\alpha^2} \right)^{1/2} \right), \tag{31}$$

where $\tau_0 = \frac{\alpha^2}{C d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha)\log(1/\delta)}$, $\mathcal{F}$ has a $\left(n, \alpha, \beta, 1, \delta \cdot \frac{\sqrt{C d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha)}}{\alpha}\right)$-sanitizer.

Proof of Theorem 6.6 using [BLM20a]. We assume familiarity with the notation and terminology of [BLM20a]. We first remark that by Proposition 2 of [BLM20a], it suffices to show the existence of a DP-fooling algorithm (for an arbitrary distribution p_{real} over $\mathcal{X}$) with sample complexity given by (31). In turn, the proof of existence of a DP-fooling algorithm is a slight modification of the proof of Proposition 1 of [BLM20a] with the parameter κ therein taken to be equal to 1; the main difference is the use of the advanced composition lemma for differential privacy as opposed to the basic composition lemma used in [BLM20a].

For any $\alpha, \beta, \delta \in (0,1)$, let the number of samples in the input dataset S to the DP-fooling algorithm be the quantity on the right-hand side of (31). As in [BLM20a], we let G be a generator that fools $\mathcal{F}$ with round complexity $T(\alpha') \leqslant O\left(\frac{d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha')}{(\alpha')^2}\right)$, for any $\alpha' \in (0,1)$. (Such a G is guaranteed by [BLM20a, Theorem 2].) Let D be the discriminator used in [BLM20a, Figure 2]. We use exactly the same fooling algorithm as in [BLM20a], except with the number of rounds set to $T_0 = T(\alpha/4)$, and set $\tau_0 = 1/\sqrt{T_0 \log(1/\delta)}$ (in [BLM20a] the settings were instead $T_0 = \min\{|S|^\kappa, T(\alpha/4)\}$ and $\tau_0 = 1/T_0$). Thus there is some constant C so that $T_0 \leqslant C \cdot \left(\frac{d^\star_{\mathrm{L}} \log(d^\star_{\mathrm{L}}/\alpha)}{\alpha^2}\right)$. To analyze privacy and utility, we use [BLM20a, Lemma 6], which uses as a black-box a proper PAC learner with sample complexity $n_0(\alpha', \beta', \varepsilon', \delta')$: in particular, in our usage of Lemma 6, the privacy parameters (ε', δ') of this PAC learner, which are referred to as $(\alpha(\tau|S|), \beta(\tau|S|))$ in [BLM20a], are $(\varepsilon', \delta') := (1, \delta)$. Moreover, the parameter α' (referred to as ε in [BLM20a, Lemma 6]) is set to $\alpha' := \alpha/8$, the parameter β' (referred to as δ in [BLM20a]) is set to $\beta' := \beta\tau_0/2$.[14] The number of samples in (31) satisfies Eq. (12) of [BLM20a], thus allowing us to apply Lemma 6 therein.

To analyze privacy of this algorithm, note that [BLM20a, Lemma 6] gives that the discriminator D is $(6\tau_0\varepsilon' + \tau_0, 4e^{6\tau_0\varepsilon'}\tau_0\delta')$-differentially private, where ε', δ' are the privacy parameters used in the proper PAC learner for $\mathcal{F}$ which has sample complexity $n_0(\alpha', \beta', \varepsilon', \delta')$. By the advanced composition lemma (e.g., [DR14, Theorem 3.20]), the overall algorithm is

$$\left(\sqrt{2T_0 \ln(1/\delta')} \cdot (6\tau_0\varepsilon' + \tau_0) + 3T_0(6\tau_0\varepsilon' + \tau_0)^2,\ T_0 \cdot 4e^{6\tau_0\varepsilon'}\tau_0\delta' + \delta'\right)$$

differentially private. Using our choice of τ_0, as well as our settings $\varepsilon' = 1, \delta' = \delta$ (recall that δ is the target approximate privacy parameter), the overall algorithm is $(O(1), O(\sqrt{T_0})\cdot\delta)$-differentially private.

To analyze utility of the overall algorithm, the argument is essentially identical as in [BLM20a]: the choice of $T_0 = T(\alpha/4)$ and the fact that the generator G has round complexity $T(\cdot)$ implies that if the guarantee of [BLM20a, Lemma 6] holds for all T_0 iterations (which is the case with probability at least $1 - T_0 \cdot (\tau_0^2\beta/2) \geqslant 1 - \beta/2$), then the distribution p_{syn} output by the generator G at its

[14] The sample complexity bound in [BLM20a, Lemma 6] includes an additional factor of τ_0, explaining the dependence of $\tau_0^2\beta/2$ in (31).

termination satisfies $\sup_{f\in\mathcal{F}}\left|\mathbb{E}_{x\sim p_{syn}}[f(x)]-\mathbb{E}_{x\sim p_S}[f(x)]\right|\leqslant\alpha/2$. Moreover, we observe that since the number of samples in (31) is at least $\Omega\left(\frac{d_V+\log(1/\beta)}{\alpha^2}\right)$, with probability at least $1-\beta/2$, it will also hold that $\sup_{f\in\mathcal{F}}|\mathbb{E}_{x\sim p_S}[f(x)]-\mathbb{E}_{x\sim p_{real}}[f(x)]|\leqslant\alpha/2$. Thus, with probability at least $1-\beta$, we have $\sup_{f\in\mathcal{F}}\left|\mathbb{E}_{x\sim p_{syn}}[f(x)]-\mathbb{E}_{x\sim p_{real}}[f(x)]\right|$, which establishes the desired DP-foolability property. □

References

[ABMS20] Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In *COLT*, pages 119–152, 2020.

[AJL+19] Jacob D. Abernethy, Young Hun Jung, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via the differential privacy lens. In *NeurIPS*, pages 8892–8902, 2019.

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *STOC*, page 852860, 2019.

[AS17] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *ICML*, page 3240, 2017.

[Ass83] Patrick Assouad. Densité et dimension. *Annales de l'Institut Fourier*, 33(3):233–282, 1983.

[BBKN14] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94:401–437, 2014.

[BBNS19] Jaroslaw Blasiok, Mark Bun, Aleksandar Nikolov, and Thomas Steinke. Towards instance-optimal private query release. In *SODA*, page 24802497, 2019.

[BDKT12] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In *STOC*, page 12691284, 2012.

[BDRS18] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated CDP. In *STOC*, page 7486, 2018.

[Ben15] Shai Ben-David. 2 notes on classes with Vapnik–Chervonenkis dimension 1. *arXiv:1507.05307*, 2015.

[Bha17] Siddharth Bhaskar. Thicket density. *arXiv:1702.03956*, 2017.

[BLM20a] Olivier Bousquet, Roi Livni, and Shay Moran. Synthetic data generators: Sequential and private. In *NeurIPS*, 2020. https://arxiv.org/abs/1902.03468, v3.

[BLM20b] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *FOCS*, 2020.

[BLR08] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[BM03] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463482, 2003.

[BMNS19] Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *COLT*, pages 269–282, 2019.

[BNS14] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory of Computing*, 12, 07 2014.

[BNS15] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *SODA*, pages 461–477, 2015.

[BNS16] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *ITCS*, page 369380, 2016.

[BNS19] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. *JMLR*, 20(146):1–33, 2019.

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

[BST14] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014.

[Bun20] Mark Bun. A computational separation between private learning and online learning. *arXiv:2007.05665*, 2020.

[BUV14] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, page 110, 2014.

[CN20] Aloni Cohen and Kobi Nissim. Towards formalizing the GDPRs notion of singling out. *PNAS*, 117(15), 2020.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[DNR+09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *STOC*, page 381390, 2009.

[DR14] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers Inc., 2014.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

[Dud99] Richard M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, 1999.

[Dwo06] Cynthia Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[ENU20] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. The power of factorization mechanisms in local and central differential privacy. In *STOC*, page 425438, 2020.

[FX14] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *COLT*, pages 1–20, 2014.

[GHM19] Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. In *NeurIPS*, pages 8702–8712, 2019.

[GKM20] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In *NeurIPS*, 2020.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *NIPS*, pages 2339–2347, 2012.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

[KLM+20] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *COLT*, pages 2263–2285, 2020.

[KLN+08] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Rashkodnikova, and Adam Smith. What can we learn privately? In *FOCS*, pages 531–540, 2008.

[KMST20] Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. In *NeurIPS*, 2020.

[KSS20] Haim Kaplan, Micha Sharir, and Uri Stemmer. How to Find a Point in the Convex Hull Privately. In *SoCG*, pages 52:1–52:15, 2020.

[Lit87] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In *FOCS*, pages 68–77, 1987.

[Mun00] J.R. Munkres. *Topology*. Featured Titles for Topology. Prentice Hall, Inc., 2000.

[MY16] Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *J. ACM*, 63(3), June 2016.

[NBW+18] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R. O'Brien, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. *Harvard Journal of Law & Technology*, 31:687–780, 2016 2018.

[Nik15] A. Nikolov. An improved private mechanism for small databases. In *ICALP*, pages 1010–1021, 2015.

[NRW19] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. In *FOCS*, pages 72–93, 2019.

[NTZ12] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. *STOC*, pages 351–360, 2012.

[Par14] Article 29 Data Protection Working Party. Opinion 05/2014 on anonymisation techniques, 2014.

[RK19] Aaron Roth and Michael Kearns. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019.

[RR10] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC*, page 765774, 2010.

[RS81] M. Reed and B. Simon. *I: Functional Analysis*. Methods of Modern Mathematical Physics. Elsevier Science, 1981.

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

[Sha12] S. Shalev-Shwartz. *Online Learning and Online Convex Optimization*. Foundations and Trends in Machine Learning, 2012.

[Sio58] Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958.

[Vad17] Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

[Vap98] Vladimir Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.