Is Efficient PAC Learning Possible with an Oracle That Responds “Yes” or “No”?

Constantinos Daskalakis COSTIS@CSAIL.MIT.EDU
MIT CSAIL & Archimedes AI
Noah Golowich NZG@MIT.EDU
MIT CSAIL

Editors: Shipra Agrawal and Aaron Roth

Abstract

The *empirical risk minimization (ERM)* principle has been highly impactful in machine learning, leading both to near-optimal theoretical guarantees for ERM-based learning algorithms as well as driving many of the recent empirical successes in deep learning. In this paper, we investigate the question of whether the ability to perform ERM, which computes a hypothesis minimizing empirical risk on a given dataset, is necessary for efficient learning: in particular, is there a weaker oracle than ERM which can nevertheless enable learnability? We answer this question affirmatively, showing that in the realizable setting of PAC learning for binary classification, a concept class can be learned using an oracle which only returns a *single bit* indicating whether a given dataset is realizable by some concept in the class. The sample complexity and oracle complexity of our algorithm depend polynomially on the VC dimension of the hypothesis class, thus showing that there is only a polynomial price to pay for use of our weaker oracle. Our results extend to the agnostic learning setting with a slight strengthening of the oracle, as well as to the partial concept, multiclass and real-valued learning settings. In the setting of partial concept classes, prior to our work no oracle-efficient algorithms were known, even with a standard ERM oracle. Thus, our results address a question of (Alon et al., 2021) who asked whether there are algorithmic principles which enable efficient learnability in this setting.
Keywords: PAC learning, ERM oracle, One-inclusion graph, Partial concept class

1. Introduction

Many of the successful techniques in modern machine learning proceed by specifying a large function class $\mathcal{H}$, such as a class of neural networks, and optimizing over $\mathcal{H}$ to find a minimizer of a loss function on a finite dataset. This approach, known as *empirical risk minimization* (ERM), has long been known to lead to near-optimal PAC learning guarantees in fundamental settings such as binary classification and regression (Vapnik and Chervonenkis, 1968, 1974; Blumer et al., 1989; Bartlett and Long, 1998; Alon et al., 1997). Due to the ability of heuristics such as gradient descent to approximately implement ERM for neural network function classes, the ERM principle also lies behind numerous empirical successes in supervised learning (Krizhevsky et al., 2012). Inspired by these successes, various works have also investigated to what extent an *oracle* which can implement ERM for a given function class is useful for learning problems beyond the PAC setting, including online learning (Block et al., 2022; Haghtalab et al., 2022; Assos et al., 2023), bandits (Simchi-Levi and Xu, 2022), and reinforcement learning (Agarwal et al., 2020; Mhammedi et al., 2023).

In this paper, we return to the basics and ask: is ERM necessary? Or can we efficiently perform learning tasks with a *weaker* oracle than an ERM oracle? For the foundational problem of realizable PAC learning, it is known that a *consistency oracle*, which returns a hypothesis $\hat{h}$ in the class $\mathcal{H}$ which is consistent with a given dataset (and fails if one does not exist), is still sufficient for efficient learning. Perhaps the most drastic way to further weaken such a consistency oracle is as follows: suppose that the oracle does not return $\hat{h}$, and only returns a *single bit* indicating whether such a $\hat{h} \in \mathcal{H}$ exists which fits the data. We refer to such an oracle as a *weak consistency oracle*. Is there a PAC learning algorithm which learns efficiently with respect to this oracle? Perhaps surprisingly, we find that the answer is *yes*. Moreover, this positive answer extends to the agnostic PAC setting, if we slightly generalize the weak consistency oracle to return the *value* of the empirical risk minimizer $\hat{h}$ on the dataset, but not $\hat{h}$ itself; we call such an oracle a *weak ERM oracle*.

Motivation. We discuss several possible sources of motivation behind a weakening of an ERM (or consistency) oracle. First, note that a weak consistency oracle, which only returns a single bit indicating whether a consistent hypothesis exists, corresponds to a *decision problem* on $\mathcal{H}$, whereas a standard consistency oracle corresponds to a *search problem*. For many natural classes of problems (e.g., see (Agrawal et al., 2004; Reith and Vollmer, 2003; Khuller and Vazirani, 1991)), the decision variant is known to be computationally cheaper than the search variant.[1] Based off of such a separation, in Proposition J.1, we provide a concrete example of a class for which implementing a weak consistency oracle is possible in polynomial time but implementing a standard consistency oracle is not, under standard computational assumptions. Thus, in such a case, our approach, via a weak consistency oracle, will lead to improved computational guarantees over the standard approach which calls a consistency oracle.

From a more theoretical perspective, the use of weak consistency and weak ERM oracles yields PAC learning bounds that do not rely on uniform convergence, in contrast to some prior analyses of ERM. A notable setting where PAC learning is known to be statistically feasible but uniform convergence fails is that of learning with *partial concept classes* (Alon et al., 2021; Long, 2001), which in turn has numerous applications including to regression (Long, 2001; Bartlett and Long, 1998), learning with fairness constraints such as multicalibration (Hu and Peale, 2023), adversarially robust learning (Attias et al., 2022), and others (see Section 4). Our results provide the first (ERM) oracle-efficient learning algorithm for partial concept classes, which addresses a question asked in (Alon et al., 2021). We emphasize that even with a *standard* ERM oracle, no efficient algorithm was known, whereas our guarantees for partial concept classes hold with a *weak* ERM oracle.

1.1. Overview of results

First, we consider the setting of PAC learning of *partial concept classes* (Alon et al., 2021; Long, 2001), which are classes $\mathcal{H} \subset \{0,1,*\}^{\mathcal{X}}$ for some domain space $\mathcal{X}$. Hypotheses $h \in \mathcal{H}$ should be thought of as undefined at points $x \in \mathcal{X}$ for which $h(x) = *$ (see Section 2 for a formal definition).[2] Our main results for this setting are as follows:

- In the realizable setting of PAC learning, any partial concept class $\mathcal{H}$ of VC dimension at most d_{VC} can be learned by an algorithm that makes polynomially many calls to a *weak consistency oracle* $\mathcal{O}^{\mathsf{con,w}}$, which receives as input a dataset $S = \{(x_i, y_i)\}_{i \in [n]}$ and returns True if there is $h \in \mathcal{H}$ satisfying $h(x_i) = y_i \in \{0,1\}$ for all i, and False otherwise. The sample complexity scales as $\tilde{O}(d_{\mathsf{VC}}^3)$ (see first part of Theorem 3.1).

- In the agnostic setting of PAC learning, the same guarantee holds, except with respect to a *weak ERM oracle* $\mathcal{O}^{\mathsf{erm,w}}$ which receives as input $S = \{(x_i, y_i)\}_{i \in [n]}$ and returns the value $\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{h(x_i) \neq y_i \vee h(x_i) = *\} \in \{0, 1/n, ..., 1\}$ (see second part of Theorem 3.1).

1. More generally, such separations can emerge for non-self-reducible problems in NP.
2. The unfamiliar reader can simply consider the special case of those $\mathcal{H}$ whose hypotheses never take the value $*$, which corresponds to standard binary classification.

We proceed to generalize our upper bounds beyond the binary setting, namely the multiclass and real-valued (regression) settings:

- For $K \in \mathbb{N}$, any multiclass concept class $\mathcal{H} \subset [K]^{\mathcal{X}}$ of Natarajan dimension at most d_{N} can be PAC learned by an algorithm that uses $n = \tilde{O}(d_{\mathsf{N}}^3 \log^4 K)$ samples and makes $K \cdot \mathrm{poly}(n)$ calls to a weak consistency oracle (or to a weak ERM oracle in the agnostic setting); see Theorem 4.1.

- Any real-valued class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ whose fat-shattering dimension at scales $\gamma \in (0,1)$ is at most $d_{\mathsf{fat},\gamma}$ can be agnostically PAC learned by an algorithm that uses n samples as long as $n \gtrsim d_{\mathsf{fat},\gamma}^3$ for an appropriate chocie of γ, using $\mathrm{poly}(n)$ calls to a weak ERM oracle. Moreover, a similar guarantee holds for the realizable setting; see Theorem 4.2.

In the setting of partial concept classes as well as the *agnostic* setting of regression, VC dimension and fat-shattering dimension, respectively, are known to characterize learnability. Thus, our results above show that there is at most a polynomial price to pay in terms of sample complexity if we require efficiency with respect to a weak ERM oracle. In contrast, in the multiclass and realizable regression settings, the optimal sample complexity is characterized by the *Daniely-Schwartz dimension* (Daniely and Shalev-Shwartz, 2014; Brukhim et al., 2022) and the *one-inclusion graph dimension* (Attias et al., 2023), respectively. These quantities can be arbitrarily smaller than our corresponding sample complexities above, namely $d_{\mathsf{N}} \log K$ and $d_{\mathsf{fat},\gamma}$, respectively.

Can our bounds for the multiclass and realizable regression settings be improved to get near-optimal sample complexity while retaining oracle efficiency? Our final results show a negative answer to this question, even when the algorithm is given a standard ERM oracle:

- Multiclass concept classes of Daniely-Schwartz dimension 1 are not PAC learnable with any finite number of ERM oracle queries; see Theorem I.2.

- In the realizable setting of regression, real-valued classes with one-inclusion graph dimension 1 are not PAC learnable with any finite number of ERM oracle queries; see Theorem I.3.

Techniques. Our results rest on a new technique to efficiently implement a randomized variant of the *one-inclusion graph algorithm*, formalized in Theorem 3.2 (see also Definition 2.7). In particular, we show first that we can obtain a weak learner by constructing a random orientation of the one-inclusion graph with bounded out-degree, as follows. For each edge we wish to orient, we take a random walk starting from each of its endpoints and inspect the distribution of hitting times of the complement of the one-inclusion graph. The vertex whose random walk reaches the complement of the one-inclusion graph sooner should have the edge directed away from it. We then use standard boosting techniques to improve the weak learner to a strong learner. A detailed proof overview may be found in Section 3.1.

Open questions. Taken together, our results represent a comprehensive treatment of the weak oracle efficiency of PAC learning in the standard settings of partial, multiclass, and real-valued learning. One intruiging question that remains is closing the gap between our $\tilde{O}(d_{\mathsf{VC}}^3)$ sample complexity in the binary setting and the fact that only $O(d_{\mathsf{VC}})$ samples are required when one is allowed access to a standard ERM oracle. In particular, is there a (polynomial-sized) cost in sample complexity to pay for using a weak oracle? Analogous questions can be asked in the multiclass and real-valued settings. Along different lines, it would be interesting to investigate the use of weaker notions of ERM oracles in more complex learning situations such as contextual bandits, online learning, and reinforcement learning.

1.2. Broader perspectives

Broadly speaking, our work connects to an extensive body of literature, both in empirical and theoretical communities, on query-efficient learning. Spurred by the increasing prevelance of proprietary models and the availability of APIs to query inputs to these models at a small cost, recent empirical research has studied to what extent such API calls, which typically each reveal a small amount of information, can be used to reconstruct information such as the model's training data or an approximation to the model itself. For instance, (Tramèr et al., 2016) showed that several types of models, including decision trees, SVMs, and neural networks can be reconstructed to high fidelity using a relatively small number of queries to evaluate the model at chosen inputs. Similar results have been shown for various specific domains, including sentence classification (Krishna et al., 2020), machine translation (Wallace et al., 2020), and sentence embedding encoders (Dziedzic et al., 2023).[3] These papers on "model stealing" roughly parallel an old line of work in learning theory on learning from queries (Angluin, 1988), in which one can make several types of queries to the ground-truth hypothesis $h^\star$, such as a *membership query* where one specifies x and receives $h^\star(x)$. (Angluin, 1988) and many follow-up works study the question of how many such queries are sufficient for learning $h^\star$.

The high-level implications of the works mentioned above parallel our own in that one often arrives at the conclusion that a surprisingly large amount of information be gleaned from a relatively small number of queries, each of which returns a relatively small number of bits. At a technical level, the above papers differ from our own in that the queries performed by the learning algorithm depend explicitly on the ground-truth hypothesis $h^\star$, whereas the oracle queries we consider are queries to the *hypothesis class* $\mathcal{H}$ without any mention of $h^\star$. Nevertheless, with the advent of methods such as in-context learning (Brown et al., 2020), which allows a fully trained large language model to simulate learning algorithms such as gradient descent, the distinction between these two settings is somewhat blurred. In particular, one could imagine a setting where a proprietary large language model itself serves as an ERM oracle for simpler classes, and thus our results demonstrating that queries which return only a few bits of information still permit learning could be informative. We leave it to future work to elucidate the question of whether the success of such weak oracles ultimately amounts to a feature (allowing learning without giving too much away) or a bug (giving enough away to allow reconstruction attacks).

2. Preliminaries

Consider a domain $\mathcal{X}$, a label set $\mathcal{Y}$, and a *concept class* $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$. Elements $h \in \mathcal{H}$ are known as *concepts* (or *hypotheses*). In this paper, we consider the following different label sets $\mathcal{Y}$:

- If $\mathcal{Y} = \{0,1,*\}$, then we say that $\mathcal{H}$ is a *partial (binary) concept class* (Alon et al., 2021). A hypothesis which outputs a label of $*$ on some $x \in \mathcal{X}$ should be interpreted as being undefined at x. In the special case that no hypothesis ever outputs $*$, a partial binary concept class is known as a *total* binary concept class. We define the binary loss function $\ell^{\mathsf{bin}}(y,y') := \mathbb{1}\{y \neq y' \vee y = * \vee y' = *\}$, for $y,y' \in \{0,1,*\}$. In words, we suffer a loss for true label y when predicting y' if we predict the wrong label *or* either y,y' is $*$.

- If $\mathcal{Y} = [K]$, then $\mathcal{H}$ is said to be a *multiclass concept class*. We define the *multiclass loss function* $\ell^{\mathsf{mc}}(y,y') := \mathbb{1}\{y \neq y'\}$, for $y,y' \in [K]$.

3. See also (Dosovitskiy and Brox, 2015; Morris et al., 2023) and references within for work on the related problem of inverting trained models.

- If $\mathcal{Y}=[0,1]$, then $\mathcal{H}$ is said to be a *real-valued concept class*. We define the *absolute loss function* $\ell^{\mathsf{abs}}(y,y') := |y-y'|$, for $y,y' \in [0,1]$.

Throughout the paper, all concept classes $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ will be understood as being either partial, multiclass, or real-valued concept classes.

2.1. PAC learning

Given a distribution $P \in \Delta(\mathcal{X} \times \mathcal{Y})$, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$, and a hypothesis $h : \mathcal{X} \to \mathcal{Y}$, we define $\mathrm{er}_{P,\ell}(h) := \mathbb{E}_{(x,y)\sim P}[\ell(h(x),y)]$. Given a sequence $S \in (\mathcal{X} \times \mathcal{Y})^n$, which we refer to as a *sample*, a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$, and $h : \mathcal{X} \to \mathcal{Y}$, we define $\widehat{\mathrm{er}}_{S,\ell}(h) := \frac{1}{n}\sum_{(x,y)\in S} \mathbb{1}\{h(x) \neq y\}$. When working with partial, multiclass, or real-valued concept classes, we will typically take ℓ to be the corresponding respective loss function among $\ell^{\mathsf{bin}},\ell^{\mathsf{mc}},\ell^{\mathsf{abs}}$. Thus, unless otherwise stated, in such situations we will write $\mathrm{er}_P(\cdot)$ in place of $\mathrm{er}_{P,\ell}(\cdot)$ and $\widehat{\mathrm{er}}_S(\cdot)$ in place of $\widehat{\mathrm{er}}_{S,\ell}(\cdot)$, where $\ell \in \{\ell^{\mathsf{bin}},\ell^{\mathsf{mc}},\ell^{\mathsf{abs}}\}$ is understood to be the appropriate choice. We denote samples using curly braces, but emphasize that samples should be interpreted as sequences of n examples (in particular, examples can be repeated).

Given a concept class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, a *sample* $S = \{(x_i,y_i)\}_{i\in[n]} \subset (\mathcal{X} \times \mathcal{Y})^n$ is said to be *$\mathcal{H}$-realizable* if there is $h \in \mathcal{H}$ so that $h(x_i) = y_i \neq *$ for each $i \in [n]$. Moreover, a distribution $P \in \Delta(\mathcal{X} \times \mathcal{Y})$ is defined to be *$\mathcal{H}$-realizable* if the following holds: in the case that $\mathcal{H}$ is a partial concept class, for any $n \in \mathbb{N}$, then a sample $S \sim P^n$ is $\mathcal{H}$-realizable with probability 1; in the case that $\mathcal{H}$ is a multiclass or real-valued concept class, then $\inf_{h\in\mathcal{H}} \mathrm{er}_P(h) = 0$. We remark that these two conditions coincide if P has finite or countable support (see (Alon et al., 2021, Lemma 33)).

Realizable oracle-efficient PAC learning. In the problem of *realizable PAC learning* (or simply *PAC learning*), for a concept class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ and a $\mathcal{H}$-realizable distribution P, an algorithm Alg receives a sample $S \sim P^n$ and outputs a hypothesis $H : \mathcal{X} \to \mathcal{Y}$. While often it is assumed that Alg has full knowledge of the class $\mathcal{H}$, we are concerned with the setting in which Alg's only access to $\mathcal{H}$ comes in the form of an oracle $\mathcal{O} : (\mathcal{X} \times \mathcal{Y})^\star \times \{0,1\}^\star \to \{0,1\}^\star$, which takes as input a sequence of examples $(x,y) \in \mathcal{X} \times \mathcal{Y}$ as well as a string of bits, and outputs a string of bits. While much prior work in the literature has focused on oracles, such as a (strong) ERM oracle (Definition 2.6), which can return *elements of* $\mathcal{H}$, the oracles we consider are weaker in the sense that their only outputs are strings of bits, which will be quite short.[4] For an input $(S,z) \in (\mathcal{X} \times \mathcal{Y})^\star \times \{0,1\}^\star$ to an oracle $\mathcal{O}$, we let the *size* of (S,z) be $|S|+|z|$, namely the sum of the number of examples in S and the number of bits in z. We say that an algorithm Alg has *cumulative query cost* q if the sum of the sizes of the inputs for all oracle calls that Alg makes to $\mathcal{O}$ is at most q. Note that the number of oracle calls made by Alg is bounded above by q.

Definition 2.1 (Oracle-efficient PAC learning)**.** Let domain and label spaces $\mathcal{X},\mathcal{Y}$ be given. Given $n \in \mathbb{N}$, let Alg be an algorithm which takes as input a dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$, $x \in \mathcal{X}$, and a string of uniformly random bits $R \in \{0,1\}^\star$, has cumulative query cost q to an oracle $\mathcal{O} : (\mathcal{X} \times \mathcal{Y})^\star \times \{0,1\}^\star \to \{0,1\}^\star$, and outputs some value $\mathsf{Alg}_R(S,x) \in \mathcal{Y}$, which is a deterministic function of R,S,x, and the results of the oracle calls.

Let $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class and $\mathcal{O}$ be an oracle as above. Given $\epsilon,\delta \in (0,1)$, we say that the class $\mathcal{H}$ is $(\mathcal{O};\epsilon,\delta)$-*PAC learnable* by Alg with *sample complexity* n and *oracle complexity* q if the following holds. Letting $H_{S,R}(x) := \mathsf{Alg}_R(S,x)$, we have $\Pr_{S\sim P^n,R}(\mathrm{er}_P(H_{S,R}) \leq \epsilon) \geq 1-\delta$.

4. Of course, elements of $\mathcal{H}$ can be represented with $\log|\mathcal{H}|$ bits, but our oracles will always return strings of length $\mathrm{poly}(\log d, \log 1/\epsilon, \log\log 1/\delta)$, which can be infinitely smaller than $\log|\mathcal{H}|$. Here d denotes a dimension quantity (e.g., VC dimension) and ϵ,δ are accuracy parameters.

We emphasize that in the above definition Alg has no knowledge of $\mathcal{H}$ (apart from its calls to $\mathcal{O}$); of course, the oracle $\mathcal{O}$ will depend on $\mathcal{H}$. Often we will slightly abuse terminology by stating that Alg "outputs" the hypothesis $H_{S,R}$.

Agnostic oracle-efficient PAC learning. The setting of *agnostic PAC learning* is similar to the setting of realizable PAC learning, except that the distribution $P\in\Delta(\mathcal{X}\times\mathcal{Y})$ is no longer required to be realizable. As such, we measure the performance of the output hypothesis of an algorithm by comparing to the best hypothesis in the class $\mathcal{H}$. In the case that $\mathcal{H}$ is a multiclass or real-valued class, the error of the best-performing concept in $\mathcal{H}$ on P is defined as $\mathrm{er}_P(\mathcal{H}):=\inf_{h\in\mathcal{H}}\mathrm{er}_P(h)$. In the case that $\mathcal{H}$ is a concept class, we instead define $\mathrm{er}_P(\mathcal{H}) := \lim_{n\to\infty}\mathbb{E}_{S\sim P^n}[\min_{h\in\mathcal{H}}\widehat{\mathrm{er}}_S(h)]$. (Alon et al., 2021, Lemma 39) shows that the limit exists, and that when $\mathcal{H}$ is a total class, the two notions of $\mathrm{er}_P(\mathcal{H})$ coincide.[5]

Definition 2.2 (Oracle-efficient agnostic PAC learning)**.** Using the setup and terminology of Definition 2.1, the class $\mathcal{H}$ is said to be $(\mathcal{O};\epsilon,\delta)$-agnostically PAC learnable by Alg with sample complexity n and oracle complexity q if, for $H_{S,R}(x):=\mathsf{Alg}_R(S,x)$, we have

$$\mathrm{Pr}_{S\sim P^n,R}(\mathrm{er}_P(H_{S,R})\leq \mathrm{er}_P(\mathcal{H})+\epsilon)\geq 1-\delta.$$

2.2. Oracles

In this section, we formally introduce the oracles that our algorithms will use. We begin with a *weak consistency oracle*, which will be used by our realizable PAC learning algorithms for partial and multiclass concept classes.

Definition 2.3 (Weak consistency oracle)**.** Given a concept class $\mathcal{H}\subset\mathcal{Y}^{\mathcal{X}}$, a *(weak) consistency oracle* $\mathcal{O}^{\mathsf{con,w}}$ for $\mathcal{H}$ is defined as follows: it takes as input a sample $S\in(\mathcal{X}\times\mathcal{Y})^n$, and $\mathcal{O}^{\mathsf{con,w}}(S)$ outputs True if S is $\mathcal{H}$-realizable and False otherwise.

For real-valued learning, a weak consistency oracle is not sufficient for learning, due to the fact that labels in $\mathcal{Y}=[0,1]$ can take infinitely many values. Therefore, for realizable PAC learning in the real-valued setting, we make use of a *range consistency oracle*, which is a natural generalization of a weak consistency oracle when one allows some margin of error in label space:

Definition 2.4 (Range consistency oracle)**.** Given a real-valued concept class $\mathcal{H}\subset[0,1]^{\mathcal{X}}$, a *range consistency oracle* $\mathcal{O}^{\mathsf{range}}$ for $\mathcal{H}$ is defined as follows: it takes as input a sample $S=\{(x_i,\ell_i,u_i)\}_{i\in[n]}\in(\mathcal{X}\times[0,1]^2)^n$, and outputs True if there is some $h\in\mathcal{H}$ so that $\ell_i\leq h(x_i)\leq u_i$ for all $i\in[n]$, and False otherwise.

The consistency oracles defined above are not sufficient for oracle-efficient agnostic PAC learning: the challenge is that even approximating the *value* of the empirical risk minimizer $\min_{h\in\mathcal{H}}\widehat{\mathrm{er}}_S(h)\in[0,1]$ on a sample $S\in(\mathcal{X}\times\{0,1\})^n$ can require many weak consistency queries to $\mathcal{H}$. Rather surprisingly, it turns out that an oracle which returns only the value of the empirical risk minimizer on a sample, as defined formally below, is sufficient for efficient agnostic PAC learnability.

Definition 2.5 (Weak ERM oracle)**.** Consider a concept class $\mathcal{H}\subset\mathcal{Y}^{\mathcal{X}}$, and a real-valued loss function $\ell:\mathcal{Y}\times\mathcal{Y}\to[0,1]$. A *weak ERM oracle* $\mathcal{O}^{\mathsf{erm,w}}$ is a mapping which takes as input a dataset $S\in(\mathcal{X}\times\mathcal{Y})^n$ and outputs the value $\min_{h\in\mathcal{H}}\widehat{\mathrm{er}}_{S,\ell}(h)\in[0,1]$.[6]

5. See Footnote 12 of (Alon et al., 2021) for discussion on why the particular choice of $\mathrm{er}_P(\mathcal{H})$ is made.
6. When $\ell\in\{\ell^{\mathsf{bin}},\ell^{\mathsf{mc}}\}$ is binary-valued, $\min_{h\in\mathcal{H}}\widehat{\mathrm{er}}_{S,\ell}(h)\in\{0,1/n,...,1\}$ can be represented with $O(\log n)$ bits. In the real-valued setting, while this is no longer the case, one can assume that $\mathcal{O}^{\mathsf{erm,w}}$ returns only the $\log(1/\epsilon)$ most significant bits of the empirical risk, at the cost of an $O(\epsilon)$ error that propagates through the PAC bounds. For simplicity, we ignore such considerations relating to arithmetic precision.

Finally, for reference, we introduce the standard notion of ERM oracle, which returns a hypothesis that minimizes the empirical risk on a sample; to contrast with a weak ERM oracle, we call such an oracle a strong ERM oracle.

Definition 2.6 (Strong ERM oracle)**.** Consider a concept class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ and a real-valued loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0,1]$. A *strong ERM oracle* $\mathcal{O}^{\mathsf{erm,s}}$ is a mapping which takes as input a dataset $S \in (\mathcal{X} \times \mathcal{Y})^n$ and outputs some concept in $\mathrm{argmin}_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{S,\ell}(h) \in \mathcal{H}$.

2.3. The one-inclusion graph

In this section, we introduce the one-inclusion graph (Haussler et al., 1988), which plays a fundamental role in many PAC learning results. For a (partial, multiclass, or real-valued) concept class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ and $X = (x_1, \ldots, x_m) \in \mathcal{X}^m$, we define $\mathcal{H}|_X := \{ y \in (\mathcal{Y} \backslash \{*\})^m \ : \ \exists h \in \mathcal{H} \text{ s.t. } h(x_i) = y_i \ \forall i \in [m] \}$. Note that, in the special case of partial concept classes, $\mathcal{H}|_X \subset \{0,1\}^m$ and in particular does *not* include the $*$ symbol. For a partial binary concept class $\mathcal{H}$, its *VC dimension*, denoted $d_{\mathsf{VC}}(\mathcal{H})$, is the largest positive integer d so that there is some $X = (x_1, \ldots, x_d) \in \mathcal{X}^d$ so that $\mathcal{H}|_X = \{0,1\}^d$. It is known that the VC dimension tightly characterizes statistical learnability of (partial) concept classes (Alon et al., 2021).

For $v \in \{0,1\}^n$ and $i \in [n]$, we write $v^{\oplus i}$ to denote v with coordinate i flipped, i.e., $v_j^{\oplus i} = v_j$ for all $j \neq i$ and $v_i^{\oplus i} = 1 - v_i$. For $n \in \mathbb{N}$, let $G_n = (V_n, E_n)$ be the n-dimensional hypercube graph, so that $V_n = \{0,1\}^n$ and $E_n = \{ ((v_{-i}, 0), (v_{-i}, 1)) \ : \ i \in [n], v_{-i} \in \{0,1\}^{n-1} \}$.

Definition 2.7 (One-inclusion graph)**.** Consider a set $\mathcal{W} \subset \{0,1\}^n$. The *one-inclusion graph* $G(\mathcal{W}) = (V, E)$ induced by $\mathcal{W}$ is defined as the following graph. The vertex set V is equal to $\mathcal{W}$. The edge set E is the subgraph of G_n induced by $\mathcal{W}$, namely:

$$E := \{ (v, v^{\oplus i}) \ : \ i \in [n], \, v, v^{\oplus i} \in \mathcal{W} \}.$$

For any $i \in [n]$ and $h \in \{0,1\}^n$, we will occassionally write $e_{i,h}$ to refer to the edge $(h, h^{\oplus i})$. For a partial concept class $\mathcal{H} \subset \{0,1,*\}^{\mathcal{X}}$ and $X \in \mathcal{X}^n$, we refer to the the one-inclusion graph induced by $\mathcal{H}|_X \subset \{0,1\}^n$ as the one-inclusion graph of $\mathcal{H}$ induced by X.

For a set $\mathcal{W} \subset \{0,1\}^n$, $\mathcal{W}^{\mathrm{c}}$ denotes its complement in $\{0,1\}^n$.

Orientations. Given a graph $G = (V, E)$, a *random orientation* of G is a mapping $\sigma : E \to \Delta(V)$, where, for all $e \in E$, $\mathrm{supp}(\sigma(e)) \subseteq e$ (i.e., $\sigma(e)$ is supported on the 2 vertices of e). σ is called an *orientation* if $\sigma(e)$ is supported on a single vertex v, in which case we will write $v = \sigma(e)$. Given a function $F : V \to [0,1]$ and $\lambda \in [0,1]$, we consider a random orientation $\sigma_{F,\lambda}$ *induced by* F, defined as follows: for an edge $e = (v, v')$, we set

$$\sigma_{F,\lambda}(e)(v) = \frac{1 + \lambda \cdot (F(v') - F(v))}{2}, \qquad \sigma_F(e)(v') = \frac{1 + \lambda \cdot (F(v) - F(v'))}{2}. \tag{1}$$

Given a random orientation σ and a vertex $v \in V$, we define the *out-degree* of σ at v to be $\mathsf{outdeg}(v; \sigma) := \sum_{e \ni v} (1 - \sigma(e)(v))$, and the out-degree of σ is $\mathsf{outdeg}(\sigma) := \max_{v \in V} \mathsf{outdeg}(v; \sigma)$.

3. Learning partial concept classes with a weak oracle

In this section, we give an algorithm for realizable PAC learning with low oracle complexity for a weak consistency oracle, and an algorithm for agnostic PAC learning with low oracle complexity for a weak ERM oracle.

Theorem 3.1 (Oracle-efficient partial concept class learning). *For any $\epsilon,\delta \in (0,1)$ and $d_{\mathsf{VC}} \in \mathbb{N}$, the following statements hold:*

1. *There is an algorithm $\mathsf{Alg}^{\mathsf{R}}$ so that for any class $\mathcal{H} \subset \{0,1,*\}^{\mathcal{X}}$ satisfying $d_{\mathsf{VC}}(\mathcal{H}) \le d_{\mathsf{VC}}$ and any weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{con,w}};\epsilon,\delta)$-PAC learnable by $\mathsf{Alg}^{\mathsf{R}}$ with sample complexity $n = \tilde{O}\left(\frac{d_{\mathsf{VC}}^3 \log(1/\delta)}{\epsilon}\right)$ and oracle complexity $\mathrm{poly}(n)$.*
2. *There is an algorithm $\mathsf{Alg}^{\mathsf{A}}$ so that for any class $\mathcal{H} \subset \{0,1,*\}^{\mathcal{X}}$ satisfying $d_{\mathsf{VC}}(\mathcal{H}) \le d_{\mathsf{VC}}$ and any weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{erm,w}};\epsilon,\delta)$-PAC learnable by $\mathsf{Alg}^{\mathsf{A}}$ with sample complexity $n = \tilde{O}\left(\frac{d_{\mathsf{VC}}^3 \log(1/\delta)}{\epsilon^2}\right)$ and oracle complexity $\mathrm{poly}(n)$.*

In the theorem statement above, $\tilde{O}(\cdot)$ hides factors which are polynomial in $\log(d_{\mathsf{VC}}), \log(1/\epsilon), \log\log(1/\delta)$. The proof of the realizable case of Theorem 3.1 proceeds by first constructing an algorithm (`WeakRealizable`; Algorithm 1) which is a *weak learner* for any class of VC dimension at most d_{VC} in the realizable setting and makes polynomially many oracle calls to $\mathcal{O}^{\mathsf{con,w}}$. We then use a standard boosting algorithm (namely, `Adaboost`; Algorithm 5) to boost the performance of the weak learner so as to obtain a learner which has error at most ϵ with high probability. To analyze the generalization error of this approach, we use a technique involving sample compression schemes (David et al., 2016; Schapire and Freund, 2012). Finally, to handle the agnostic case of Theorem 3.1, we reduce to the realizable setting by showing that a weak ERM oracle can be used in an efficient manner to determine, given any sample $S \in (\mathcal{X} \times \{0,1\})^n$, a subsample of maximum size which is $\mathcal{H}$-realizable (Lemma C.1). In the remainder of the section, we introduce our weak learner; the remaining ingredients are (mostly) standard and are presented in Appendices E and F.

3.1. An oracle-efficient weak learner

Our goal is to construct an oracle-efficient weak learner, namely one that improves upon random guessing in expectation over its dataset by a small margin $\eta > 0$:

Definition 3.1 (Weak learner). For $m \in \mathbb{N}$ and $\eta \in (0,1)$, a randomized learning algorithm $\mathscr{A} : (\mathcal{X} \times \{0,1\})^m \times \mathcal{X} \to \{0,1\}$ is defined to be a *m-sample weak learner with margin η* for the concept class $\mathcal{H}$ if the following holds. For any $\mathcal{H}$-realizable distribution $P \in \Delta(\mathcal{X} \times \{0,1\})$, $\mathscr{A}$ takes as input an i.i.d. sample $S \sim P^m$ and $x \in \mathcal{X}$ and outputs a (random) bit $\mathscr{A}(S,x)$, so that

$$\mathbb{E}_{S\sim P^m}\mathbb{E}_{(x,y)\sim P}\mathbb{E}_{\mathscr{A}}\left[\ell^{\mathsf{bin}}(\mathscr{A}(S,x),y)\right] \le \frac{1}{2} - \eta. \tag{2}$$

We construct an oracle-efficient weak learner using polynomially many calls to a weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ by simulating a random walk on the one-inclusion graph of $\mathcal{H}|_X$ for an appropriate choice of $X \in \mathcal{X}^m$. This procedure is formalized in the `WeakRealizable` algorithm (Algorithm 1), whose main guarantee is shown below:

Theorem 3.2 (Weak learning guarantee). *There are constants C_1, C_2 so that the following holds. Consider a partial concept class $\mathcal{H}$ of VC dimension d, $\delta \in (0,1)$, and suppose $m \ge C_1 d \log d$. For an $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^{m-1}$ and $x \in \mathcal{X}$, let $\mathscr{A}(S,x) \in \{0,1\}$ be the output of `WeakRealizable`$(S, x, 1 - \frac{1}{C_1 m \log m}, 1, C_1 m^2 \log^3 m, \mathcal{O}^{\mathsf{con,w}})$ (Algorithm 1), which is a random variable. Then for any $\mathcal{H}$-realizable sample $S = \{(x_i,y_i)\}_{i\in[m]} \in (\mathcal{X} \times \{0,1\})^m$, it holds that*

$$\frac{1}{m}\sum_{i=1}^m \mathbb{E}\left[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i},x_i),y_i)\right] \le \frac{1}{2} - \frac{1}{C_2 m \log m} \tag{3}$$

where the expectation is taken over the randomness in the runs of $\mathscr{A}(S_{-i},x_i)$. Moreover, `WeakRealizable` *makes at most $\tilde{O}(m^3)$ calls to $\mathcal{O}^{\mathsf{con,w}}$, each with a dataset of size $m-1$.*

Algorithm 1 Weak oracle-efficient OIG learner

Require: A partial concept class $\mathcal{H}$, an $\mathcal{H}$-realizable sample $S=\{(x_i,y_i)\}_{i\in[m-1]}\in(\mathcal{X}\times\{0,1\})^{m-1}$, query point $x\in\mathcal{X}$, consistency oracle $\mathcal{O}^{\mathsf{con,w}}$, parameters $\lambda,\gamma\in(0,1)$, $U\in\mathbb{N}$.

1: **function** $\textsc{WeakRealizable}(S,x,\gamma,\lambda,U,\mathcal{O}^{\mathsf{con,w}})$

2: Set $X\leftarrow(x_1,\ldots,x_{m-1},x)\in\mathcal{X}^m$.

3: Set $y^0\leftarrow(y_1,\ldots,y_{m-1},0)\in\{0,1\}^m$ and $y^1\leftarrow(y_1,\ldots,y_{m-1},1)\in\{0,1\}^m$.

4: **if** $\mathcal{O}^{\mathsf{con,w}}(\{(X_j,y_j^b)\}_{j\in[m]})=\mathsf{False}$ for some $b\in\{0,1\}$ **then**

5: **return** $1-b$.

6: For $b\in\{0,1\}$, set $\hat{F}(y^b)\leftarrow\texttt{EstimatePotential}(X,y^b,K,\gamma,\mathcal{O}^{\mathsf{con,w}})$.

7: **return** a sample from $\mathrm{Ber}(\hat{\sigma})$, where $\hat{\sigma}:=\frac{1+\lambda\cdot(\hat{F}(y^0)-\hat{F}(y^1))}{2}$.

Require: $U,\gamma,\mathcal{O}^{\mathsf{con,w}}$ as above, and $X\in\mathcal{X}^m,y\in\{0,1\}^m$.

8: **function** $\textsc{EstimatePotential}(X,y,U,\gamma,\mathcal{O}^{\mathsf{con,w}})$ $\triangleright$ *y represents a vertex of the OIG induced by $\mathcal{H}|_X$, and U is the number of trials*

9: **for** $1\leq u\leq U$ **do**

10: Set $Y^{(0)}\leftarrow y$ and $T_u\leftarrow\frac{\log(32e/(1-\gamma))}{\log(1/\gamma)}$.

11: **for** $0\leq t\leq\frac{\log(32e/(1-\gamma))}{\log(1/\gamma)}$ **do**

12: **if** $\mathcal{O}^{\mathsf{con,w}}(\{(X_j,(Y^{(t)})_j)\}_{j\in[m]})=\mathsf{False}$ **then**

13: Set $T_u\leftarrow t$, and **break** (out of the inner **for** loop).

14: **else**

15: Choose $i\sim\mathrm{Unif}([m])$, and set $Y^{(t+1)}\leftarrow(Y^{(t)})^{\oplus i}$.

16: **return** the quantity $\frac{1}{U}\sum_{u=1}^{U}\gamma^{T_u}$.

The guarantee (3), in which an arbitrary realizable dataset S is fixed and the algorithm's performance is measured on all leave-one-out configurations of S, is known as a *transductive learning* guarantee. A standard exchangeability argument (see Lemma D.4) shows that (3) implies an in-expectation error guarantee under any realizable distribution $P\in\Delta(\mathcal{X}\times\{0,1\})$, and thus Theorem 3.2 implies that `WeakRealizable` is an m-sample weak learner with margin $\eta=\Theta(1/(m\log m))$ for $\mathcal{H}$. In the remainder of the section we focus on the proof of Theorem 3.2.

Analyzing `WeakRealizable`. Given a dataset $S=\{(x_i,y_i)\}_{i\in[m-1]}$ together with a "query point" $x\in\mathcal{X}$, `WeakRealizable` considers the two vertices $y^0=(y_1,\ldots,y_{m-1},0),y^1=(y_1,\ldots,y_{m-1},1)$ of the one-inclusion graph $G(\mathcal{H}|_X)$ induced by $\mathcal{H}$ on the sequence $X=(x_1,\ldots,x_{m-1},x)$. (If y^b is not a vertex of $G(\mathcal{H}|_X)$ for some $b\in\{0,1\}$, then, by realizability, the correct prediction on x must be $1-b$ – see Line 5 of Algorithm 1.) `WeakRealizable` then calls `EstimatePotential` on each of the vertices y^0,y^1, which returns estimates $\hat{F}(y^0),\hat{F}(y^1)$ of a certain potential function on vertices of G. These potentials are used to randomly return an output bit in Line 7.

The proof that `WeakRealizable` satisfies (3) proceeds by considering the following perspective: the value $\hat{\sigma} = \frac{1+\lambda(\hat{F}(y^0)-\hat{F}(y^1))}{2}$ computed in Line 7 can be viewed as a decision to randomly orient the edge (y^0, y^1) of the one-inclusion graph $G(\mathcal{H}|_X)$ by putting mass $\hat{\sigma}$ on y^1 and mass $1-\hat{\sigma}$ on y^0 (see Section 2.3). To minimize loss, we hope that this orientation puts as much mass as possible on whichever of y^0, y^1 corresponds to the ground-truth hypothesis, i.e., we want the edge (y^0, y^1) to not contribute much to the out-degree of the ground-truth.

Translated into this language of orientations, the transductive error guarantee (3) of Theorem 3.2 is therefore equivalent to the following statement: fix $m \in \mathbb{N}$, consider any $\mathcal{H}$-realizable dataset $S = \{(x_i, y_i)\}_{i\in[m]}$, and let $X = (x_1, \ldots, x_m)$. Then the random orientations of the m edges adjacent to $y \in G(\mathcal{H}|_X)$ induced by running `WeakRealizable` with inputs (S_{-i}, x_i), for each $i \in [m]$, lead the *out-degree* of y to be bounded above by $m \cdot (1/2 - \Omega(1/(m \log m)))$. As a sanity check, it is trivial to achieve out-degree $m/2$ by orienting each edge to each of its vertices with probability $1/2$; thus, the quantity of interest is the decrease of $-m \cdot \Omega(1/(m \log m))$ in the out-degree.

To explain how we achieve such an out-degree bound, consider the m-dimensional hypercube graph $G_m = (V_m, E_m)$. Note that the one-inclusion graph $G(\mathcal{H}|_X)$ is the subgraph of G_m induced by $\mathcal{H}|_X$. Given $v \in V_m$, we consider the (lazy) random walk on G_m started at v. In particular, it is the sequence $Z_v^{(0)}, Z_v^{(1)}, Z_v^{(2)}, \ldots \in V_m$ of random variables with $Z_v^{(0)} = v$, and with $Z_v^{(t)}$ defined as follows, for $t \geq 0$: given $Z_v^{(t)} \in V_m$, the value of $Z_v^{(t+1)}$ is defined by selecting uniformly at random an edge e of G_m containing $Z_v^{(t)}$, and then letting $Z_v^{(t+1)}$ to be a uniformly random vertex of e. Given a subset $\mathcal{S} \subset V_m$ and a vertex $v \in V_m$, the *hitting time* for $\mathcal{S}$ starting at v is the random variable

$$\tau_{\mathcal{S},v} := \min\left\{ t \geq 0 \; : \; Z_v^{(t)} \in \mathcal{S} \right\}. \tag{4}$$

Moreover, the generating function $M_{\mathcal{S},v}(\gamma)$, for $\gamma \in (0,1)$, is defined for $v \in V_m$ by $M_{\mathcal{S},v}(\gamma) := \mathbb{E}[\gamma^{\tau_{\mathcal{S},v}}]$.[7] The definition of the random walk yields the following recursive formula for $M_{\mathcal{S},v}(\gamma)$ (see Lemma D.1 for a formal statement): for all $v \in \mathcal{S}^{\mathrm{c}}$,

$$M_{\mathcal{S},v}(\gamma) = \frac{\gamma}{(2-\gamma)m} \sum_{i\in[m]} M_{\mathcal{S},v^{\oplus i}}(\gamma). \tag{5}$$

Given $\mathcal{H}$ and $X \in \mathcal{X}^m$ as above, we now choose $\mathcal{S} := (\mathcal{H}|_X)^{\mathrm{c}}$, $\gamma := 1 - \Theta(1/(m \log m))$, and define $F(v) = M_{\mathcal{S},v}(\gamma)$. We may consider the orientation $\sigma_{F,1}$ induced by F (see (1)). It is a simple consequence of (5) (see Lemma D.2 that

$$\mathsf{outdeg}(\sigma_{F,1}) \leq \frac{m}{2} - (1-\gamma)m \cdot \min_{v\in V_m} F(v). \tag{6}$$

Finally, we can show (in Lemma D.3) that as long as $m \geq \Omega(d \log d)$ (where d is an upper bound on $d_{\mathsf{VC}}(\mathcal{H})$), we have $\min_{v\in V_m} F(v) \geq \Omega(1)$. This statement is a consequence of the Sauer-Shelah lemma, which bounds $|\mathcal{S}^{\mathrm{c}}| = |\mathcal{H}|_X| \leq (em)^d$. In particular, since $\mathcal{S}^{\mathrm{c}}$ is relatively "small", the hitting time $\tau_{\mathcal{S},v}$ cannot get too large for any vertex v, meaning that $\gamma^{\tau_{\mathcal{S},v}}$ cannot become too small. To summarize, we thus obtain from (6) that $\mathsf{outdeg}(\sigma_{F,1}) \leq m \cdot \left(\frac{1}{2} - \Omega(1-\gamma)\right) = m \cdot \left(\frac{1}{2} - \Omega\left(\frac{1}{m \log m}\right)\right)$.

Since the generating function $F(v)$ is not known exactly, `WeakRealizable` cannot compute the orientation $\sigma_{F,1}$ exactly. Instead, it computes estimates $\hat{F}(y^0), \hat{F}(y^1)$ of $F(y^0), F(y^1)$ respectively

7. For $v \in \mathcal{S}$, we have $\tau_{\mathcal{S},v} = 0$ and hence $M_{\mathcal{S},v}(\gamma) = 1$.

(using `EstimatePotential`), via random rollouts. Crucially, doing so is possible using only a weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$: we only need to be able to check, at each step, whether the random walk has hit $\mathcal{S}=(\mathcal{H}|_X)^{\mathsf{c}}$, which is exactly what is accomplished by $\mathcal{O}^{\mathsf{con,w}}$. By standard concentration arguments, we can show that the induced orientation $\sigma_{\hat{F},1}$ is sufficiently close to $\sigma_{F,1}$ to enjoy the same outdegree bounds, thus establishing Theorem 3.2. The full details of the proof of Theorem 3.2 may be found in Appendix D.

4. Extensions to multiclass and real-valued classes

We next extend the guarantee of Theorem 3.1 to the settings of *multiclass classification* and *regression*. The proofs for both of these settings proceed via a reduction to the case of partial concept classes.

4.1. Multiclass concept classes

Our upper bounds for the multiclass setting are phrased in terms of *Natarajan dimension*: for a multiclass concept class $\mathcal{H}\subset[K]^{\mathcal{X}}$, its *Natarajan dimension*, denoted $d_{\mathsf{N}}(\mathcal{H})$, is the smallest $d\in\mathbb{N}$ so that there is some $X=(x_1,...,x_d)\in\mathcal{X}^d$ together with vectors $a,b\in[K]^d$ with $a_i\neq b_i$ for all $i\in[d]$ so that $\mathcal{H}|_X\supseteq\{a_1,b_1\}\times\cdots\times\{a_d,b_d\}$. It is known that the algorithm the algorithm which returns an empirical risk minimizer of $\mathcal{H}$ on an i.i.d. sample, which requires access to a *strong* ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$, enjoys sample complexity for PAC learning of $\tilde{O}(d_{\mathsf{N}}(\mathcal{H})\log(K)/\epsilon)$ in the realizable setting and of $\tilde{O}(d_{\mathsf{N}}(\mathcal{H})\log(K)/\epsilon^2)$ in the agnostic setting (Daniely et al., 2011). Theorem 4.1 shows that we can extend this result to the setting where we only have a *weak* ERM oracle, as long as the oracle complexity is allowed to grow linearly with K.

Theorem 4.1 (Oracle-efficient multiclass learning)**.** *For any $\epsilon,\delta\in(0,1)$ and $d_{\mathsf{N}}\in\mathbb{N}$, the following statements hold:*

1. *There is an algorithm $\mathsf{Alg}^{\mathsf{R}}$ so that for any class $\mathcal{H}\subset[K]^{\mathcal{X}}$ satisfying $d_{\mathsf{N}}(\mathcal{H})\leq d_{\mathsf{N}}$ and any weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{con,w}};\epsilon,\delta)$-PAC learnable by $\mathsf{Alg}^{\mathsf{R}}$ with sample complexity $n=\tilde{O}\left(\frac{d_{\mathsf{N}}^3\log^4(K/\delta)}{\epsilon}\right)$ and oracle complexity $K\cdot\mathrm{poly}(n)$.*

2. *There is an algorithm $\mathsf{Alg}^{\mathsf{A}}$ so that for any class $\mathcal{H}\subset[K]^{\mathcal{X}}$ satisfying $d_{\mathsf{N}}(\mathcal{H})\leq d_{\mathsf{N}}$ and any weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{erm,w}};\epsilon,\delta)$-agnostically PAC learnable with sample complexity $n=\tilde{O}\left(\frac{d_{\mathsf{N}}^3\log^4(K/\delta)}{\epsilon^2}\right)$ and oracle complexity $K\cdot\mathrm{poly}(n)$.*

The $\tilde{O}(\cdot)$ above hides factors that are polynomial in $\log(1/\epsilon),\log(d_{\mathsf{N}}),\log(K/\delta)$. It is straightforward to show that oracle complexity growing linearly in K is necessary if one only uses a weak ERM or consistency oracle, by considering the case where $\mathcal{H}$ is a class that consists of a single unknown hypothesis on a large domain $\mathcal{X}$, and where the covariates are uniformly distributed on $\mathcal{X}$.

It is known that for any class $\mathcal{H}\subset[K]^{\mathcal{X}}$, the sample complexity of PAC learning $\mathcal{H}$ is always within a polynomial factor of the DS dimension of $\mathcal{H}$, denoted $d_{\mathsf{DS}}(\mathcal{H})$ (Brukhim et al., 2022), and is in particular bounded below by $\Omega(d_{\mathsf{DS}}(\mathcal{H}))$ (see Appendix I.1 for a definition of the DS dimension). Moreover, we always have $d_{\mathsf{N}}(\mathcal{H})\leq d_{\mathsf{DS}}(\mathcal{H})\leq O(d_{\mathsf{N}}(\mathcal{H})\cdot\log K)$. Thus, the sample complexity obtained by the oracle-efficient algorithms $\mathsf{Alg}^{\mathsf{R}},\mathsf{Alg}^{\mathsf{A}}$ of Theorem 4.1 comes within a $\mathrm{polylog}K$ factor of the optimal sample complexity. While the $\log K$ factor is unlikely to be large in many applications, it is nevertheless of theoretical interest to wonder if there is an oracle-efficient algorithm with sample complexity $\mathrm{poly}(d_{\mathsf{DS}}(\mathcal{H}))$, even if one allows a *strong* ERM oracle. We show in Theorem I.2 (Appendix I) that no such algorithm exists, even if we restrict $d_{\mathsf{DS}}(\mathcal{H})=1$.

4.2. Real-valued concept classes

Our bounds for the regression setting are phrased in terms of *fat-shattering dimension*: for a real-valued concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ and $\gamma \in (0,1)$, its *fat-shattering dimension at scale* γ, denoted $d_{\mathsf{fat},\gamma}(\mathcal{H})$, is the largest positive integer d so that there exist $x_1,...,x_d \in \mathcal{X}$ and $s_1,...,s_d \in [0,1]$ so that, for all $b \in \{0,1\}^d$, there is some $h \in \mathcal{H}$ so that $h(x_i) \geq s_i + \gamma$ if $b_i = 1$ and $h(x_i) \leq s_i - \gamma$ if $b_i = 0$. It is known that finiteness of the fat-shattering dimension at all scales γ is a sufficient condition for learnability in both the realizable and agnostic settings, and that a sample complexity scaling nearly linearly with the fat-shattering dimension at an appropriate scale can be obtained by outputting an empirical risk minimizer of $\mathcal{H}$ on an i.i.d. sample (which requires access to a strong ERM oracle) (Long, 2001; Bartlett and Long, 1998; Alon et al., 1997). Theorem 4.2 shows that we can extend this result to the setting where we only have a weak ERM oracle, with a polynomial cost in the sample complexity.

Theorem 4.2 (Oracle-efficient regression). *For any $\delta \in (0,1)$, $n \in \mathbb{N}$, and function $\gamma \mapsto d_{\mathsf{fat},\gamma} \in \mathbb{N}$ (for $\gamma \in (0,1)$), the following statements hold:*

1. *There is an algorithm $\mathsf{Alg}^{\mathsf{R}}$ so that for any class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ satisfying $d_{\mathsf{fat},\gamma}(\mathcal{H}) \leq d_{\mathsf{fat},\gamma}$ for all γ and any weak range oracle $\mathcal{O}^{\mathsf{range}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{range}};\epsilon,\delta)$-PAC learnable with sample complexity n and oracle complexity $\mathrm{poly}(n)$, for $\epsilon = \inf_{\gamma \in [0,1]} \left\{ O(\gamma) + \tilde{O}\left(\frac{d_{\mathsf{fat},\gamma}^3 \cdot \log(1/\delta)}{n} \right) \right\}$.*

2. *There is an algorithm $\mathsf{Alg}^{\mathsf{A}}$ so that for any class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ satisfying $d_{\mathsf{fat},\gamma}(\mathcal{H}) \leq d_{\mathsf{fat},\gamma}$ for all γ and any weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$, the class $\mathcal{H}$ is $(\mathcal{O}^{\mathsf{erm,w}};\epsilon,\delta)$-agnostically PAC learnable with sample complexity n and oracle complexity $\mathrm{poly}(n)$, for*
$\epsilon = \inf_{\gamma \in [0,1]} \left\{ O(\gamma) + \tilde{O}\left(\sqrt{\frac{d_{\mathsf{fat},\gamma}^3 \cdot \log(1/\delta)}{n}} \right) \right\}$.

The $\tilde{O}(\cdot)$ above hides factors that are polynomial in $\log(n), \log(d_{\mathsf{fat},\gamma}), \log\log(1/\delta)$. In the agnostic setting the fat-shattering dimension is known to characterize PAC learnability, and thus Theorem 4.2 shows that the price to pay for oracle-efficiency with respect to $\mathcal{O}^{\mathsf{erm,w}}$ is only a polynomial (assuming reasonable growth of $d_{\mathsf{fat},\gamma}$). In contrast, in the realizable setting, the sample complexity is characterized by a different quantity known as *the one-inclusion graph (OIG) dimension* (Attias et al., 2023), which can be smaller than the fat-shattering dimension by an arbitrarily large factor. We show in Theorem I.3 (Appendix I.2) that, even with a strong ERM oracle, it is impossible to obtain an oracle-efficient algorithm even for classes whose OIG dimension is a constant.

Acknowledgements

CD is supported by NSF Awards CCF-1901292, DMS-2022448, and DMS2134108, a Simons Investigator Award, and the Simons Collaboration on the Theory of Algorithmic Fairness. NG is supported by a Fannie & John Hertz Foundation Fellowship and an NSF Graduate Fellowship.

References

Ishaq Aden-Ali, Yeshwanth Cherapanamjeri, Abhishek Shetty, and Nikita Zhivotovskiy. Optimal pac bounds without uniform convergence, 2023.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20095–20107. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/e894d787e2fd6c133af47140aa156f00-Paper.pdf`.

Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of Mathematics*, 160:781–793, 2004. URL `https://api.semanticscholar.org/CorpusID:5314184`.

Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *J. ACM*, 44(4):615–631, jul 1997. ISSN 0004-5411. doi: 10.1145/263867.263927. URL `https://doi.org/10.1145/263867.263927`.

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of pac learnability of partial concept classes. *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 658–671, 2021. URL `https://api.semanticscholar.org/CorpusID:236087943`.

Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, apr 1988. ISSN 0885-6125. doi: 10.1023/A:1022821128753. URL `https://doi.org/10.1023/A:1022821128753`.

Angelos Assos, Idan Attias, Yuval Dagan, Constantinos Daskalakis, and Maxwell K. Fishelson. Online learning and solving infinite games with an erm oracle. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 274–324. PMLR, 12–15 Jul 2023. URL `https://proceedings.mlr.press/v195/assos23a.html`.

Idan Attias, Steve Hanneke, and Yishay Mansour. A characterization of semi-supervised adversarially robust PAC learnability. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=B7Q2mbIFa6Q`.

Idan Attias, Steve Hanneke, Alkis Kalavasis, Amin Karbasi, and Grigoris Velegkas. Optimal learners for realizable regression: Pac learning and online learning, 2023.

Peter L. Bartlett and Philip M. Long. Prediction, learning, uniform convergence, and scale-sensitive dimensions. *Journal of Computer and System Sciences*, 56(2):174–190, 1998. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1557. URL `https://www.sciencedirect.com/science/article/pii/S0022000097915579`.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1716–1786. PMLR, 02–05 Jul 2022. URL `https://proceedings.mlr.press/v178/block22a.html`.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, oct 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL https://doi.org/10.1145/76359.76371.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

N. Brukhim, D. Carmon, I. Dinur, S. Moran, and A. Yehudayoff. A characterization of multiclass learnability. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 943–955, Los Alamitos, CA, USA, nov 2022. IEEE Computer Society. doi: 10.1109/FOCS54457.2022.00093. URL https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00093.

Nataly Brukhim, Elad Hazan, Shay Moran, Indraneel Mukherjee, and Robert E Schapire. Multiclass boosting and the cost of weak learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3057–3067. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/17f5e6db87929fb55cebeb7fd58c1d41-Paper.pdf.

Nataly Brukhim, Amit Daniely, Yishay Mansour, and Shay Moran. Multiclass boosting: Simple and intuitive weak learning criteria. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=nQ84YY9Iut.

Nataly Brukhim, Steve Hanneke, and Shay Moran. Improper multiclass boosting. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5433–5452. PMLR, 12–15 Jul 2023b. URL https://proceedings.mlr.press/v195/brukhim23a.html.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 287–316, Barcelona, Spain, 13–15 Jun 2014. PMLR. URL https://proceedings.mlr.press/v35/daniely14b.html.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In Sham M. Kakade and Ulrike von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 207–232, Budapest, Hungary, 09–11 Jun 2011. PMLR. URL https://proceedings.mlr.press/v19/daniely11a.html.

Ofir David, Shay Moran, and Amir Yehudayoff. Supervised learning through the lens of compression. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

URL https://proceedings.neurips.cc/paper_files/paper/2016/file/59f51fd6937412b7e56ded1ea2470c25-Paper.pdf.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4829–4837, 2015. URL https://api.semanticscholar.org/CorpusID:206594470.

Adam Dziedzic, Franziska Boenisch, Mingjian Jiang, Haonan Duan, and Nicolas Papernot. Sentence embedding encoders are easy to steal but hard to defend. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023. URL https://openreview.net/forum?id=XN5qOxI8gkz.

Dylan J. Foster, Sham M. Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making, 2021.

Nika Haghtalab, Yanjun Han, Abhishek Shetty, and Kunhe Yang. Oracle-efficient online learning for smoothed adversaries. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=SbHxPRHPc2u.

D. Haussler, N. Littlestone, and M.K. Warmuth. Predicting (0, 1)-functions on randomly drawn points. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 100–109, 1988. doi: 10.1109/SFCS.1988.21928.

David Haussler and Philip M Long. A generalization of sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71(2):219–240, 1995. ISSN 0097-3165. doi: https://doi.org/10.1016/0097-3165(95)90001-2. URL https://www.sciencedirect.com/science/article/pii/0097316595900012.

Lunjia Hu and Charlotte Peale. Comparative Learning: A Sample Complexity Theory for Two Hypothesis Classes. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:30, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-263-1. doi: 10.4230/LIPIcs.ITCS.2023.72. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.72.

Samir Khuller and Vijay V. Vazirani. Planar graph coloring is not self-reducible, assuming p not np. *Theoretical Computer Science*, 88(1):183–189, 1991. ISSN 0304-3975. doi: https://doi.org/10.1016/0304-3975(91)90081-C. URL https://www.sciencedirect.com/science/article/pii/030439759190081C.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byl5NREFDr.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Arjen K. Lenstra. *Integer Factoring*, pages 611–618. Springer US, Boston, MA, 2011. ISBN 978-1-4419-5906-5. doi: 10.1007/978-1-4419-5906-5_455. URL https://doi.org/10.1007/978-1-4419-5906-5_455.

David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006. URL http://scholar.google.com/scholar.bib?q=info:3wf9IU94tyMJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&ct=citation&cd=0.

Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability. 2003. URL https://api.semanticscholar.org/CorpusID:9780485.

Philip M. Long. On agnostic learning with{0, *, 1}-valued and real-valued hypotheses. In *COLT/EuroCOLT*, 2001. URL https://api.semanticscholar.org/CorpusID:39397825.

Zakaria Mhammedi, Dylan J. Foster, and Alexander Rakhlin. Representation learning with multi-step inverse kinematics: an efficient and optimal approach to rich-observation rl. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. Language model inversion, 2023.

Indraneel Mukherjee and Robert E. Schapire. A theory of multiclass boosting. *J. Mach. Learn. Res.*, 14(1):437–497, feb 2013. ISSN 1532-4435.

Steffen Reith and Heribert Vollmer. Optimal satisfiability for propositional calculi and constraint satisfaction problems. *Inf. Comput.*, 186(1):1–19, oct 2003. ISSN 0890-5401. doi: 10.1016/S0890-5401(03)00092-0. URL https://doi.org/10.1016/S0890-5401(03)00092-0.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012. ISBN 0262017180.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 80–91, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 1581130570. doi: 10.1145/279943.279960. URL https://doi.org/10.1145/279943.279960.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

David Simchi-Levi and Yunzong Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Math. Oper. Res.*, 47(3):1904–1931, aug 2022. ISSN 0364-765X. doi: 10.1287/moor.2021.1193. URL https://doi.org/10.1287/moor.2021.1193.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 601–618, USA, 2016. USENIX Association. ISBN 9781931971324.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979).

Vladimir Vapnik and A Ya Chervonenkis. Algorithms with complete memory and recurrent algorithms in the problem of learning pattern recognition. *Avtomat. i Telemekh*, 4:95–106, 1968.

Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5531–5546, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.446. URL `https://aclanthology.org/2020.emnlp-main.446`.

Appendix A. Additional Preliminaries

In this section, we give some additional preliminaries which will be useful in the proofs. Our techniques will involve the use of sample compression schemes, which we proceed to define.

Definition A.1 (Sample compression scheme; (Littlestone and Warmuth, 2003; David et al., 2016)). Fix a domain $\mathcal{X}$ and a label set $\mathcal{Y}$. A *compression scheme* for the tuple $(\mathcal{X},\mathcal{Y})$ is a pair (κ,ρ), consisting of a compression function $\kappa : (\mathcal{X}\times\mathcal{Y})^\star \to (\mathcal{X}\times\mathcal{Y})^\star \times \{0,1\}^\star$ and a *reconstruction function* $\rho : (\mathcal{X}\times\mathcal{Y})^\star \times \{0,1\}^\star \to \mathcal{Y}^{\mathcal{X}}$, satisfying the following property. For any sequence $S \in (\mathcal{X}\times\mathcal{Y})^\star$, $\kappa(S)$ evaluates to some tuple $(S',B) \in (\mathcal{X}\times\mathcal{Y})^\star \times \{0,1\}^\star$, where S' is a sequence of elements of S.

For $S \in (\mathcal{X}\times\mathcal{Y})^m$, writing $(S',B) := \kappa(S)$, define $|\kappa(S)| := |S'|+|B|$, i.e., to denote the sum of the number of samples in S' and the length of B. The *size* of the compression scheme (κ,ρ) for *m-sample datasets* is $|\kappa| := \max_{S\in(\mathcal{X}\times\mathcal{Y})^{\leq m}} |\kappa(S)|$.

For a (partial, multiclass, or real-valued) concept class $\mathcal{H}$, a *sample compression scheme for $\mathcal{H}$* is a compression scheme (κ,ρ), so that for every $\mathcal{H}$-realizable sequence $S \in (\mathcal{X}\times\mathcal{Y})^m$, $\rho(\kappa(S))$ correctly classifies every point in S, i.e., $\frac{1}{n}\sum_{(x,y)\in S} \ell(\rho(\kappa(S)),y) = 0$, where $\ell \in \{\ell^{\mathsf{bin}},\ell^{\mathsf{mc}},\ell^{\mathsf{abs}}\}$ is the appropriate loss function corresponding to $\mathcal{H}$.

Lemma A.1 below shows that compression schemes of bounded size generalize.

Lemma A.1 (Generalization-by-compression; Theorem 2.1 of (David et al., 2016)). *There is a constant $C>0$ so that the following holds. Consider any domain $\mathcal{X}$ and label set $\mathcal{Y}$, together with a loss function $\ell : \mathcal{Y}\times\mathcal{Y}\to[0,1]$. For any compression scheme (κ,ρ), for any $n\in\mathbb{N}$, and $\delta\in(0,1)$, for any distribution $P\in\Delta(\mathcal{X}\times\{0,1\})$, the following holds with probability $1-\delta$ over $S\sim P^n$:*

$$\begin{aligned}
&|\mathrm{er}_{P,\ell}(\rho(\kappa(S))) - \widehat{\mathrm{er}}_{S,\ell}(\rho(\kappa(S)))| \\
&\leq C\sqrt{\widehat{\mathrm{er}}_{S,\ell}(\rho(\kappa(S)))\cdot\frac{1}{n}\left(|\kappa(S)|\log(n)+\log\frac{1}{\delta}\right)} + C\cdot\frac{1}{n}\left(|\kappa(S)|\log(n)+\log\frac{1}{\delta}\right).
\end{aligned}$$

In particular, if $\widehat{\mathrm{er}}_{S,\ell}(\rho(\kappa(S)))=0$, then

$$\mathrm{er}_{P,\ell}(\rho(\kappa(S))) \leq C\cdot\frac{1}{n}\left(|\kappa(S)|\log(n)+\log\frac{1}{\delta}\right).$$

Appendix B. Helpful lemmas

In this section we collect various probabilistic lemmas which are used throughout the proofs. Fix $n\in\mathbb{N}$, and consider the hypercube $V_n = \{0,1\}^n$. For some $v\in V$, we consider the lazy random walk on V_n, denoted $Z_v^{(0)},Z_v^{(1)},\ldots$, where $Z_v^{(0)}$, and $Z_v^{(t)}$ is generated from $Z_v^{(t-1)}$ by picking $i\in[n]$ uniformly at random and flipping the ith coordinate of $Z_v^{(t-1)}$ with probability $1/2$.

Lemma B.1 (Mixing time of the hypercube; (Levin et al., 2006)). *Consider $n\in\mathbb{N}$, $v\in V_n$, and let $Z_v^{(0)},Z_v^{(1)},\ldots$ denote the lazy random walk on the hypercube V_n. Let U be a uniformly distributed random variable on V_n. Then for any $\epsilon\in(0,1)$ and $t\geq n\log n + n\log(1/\epsilon)$, it holds that $D_{\mathsf{TV}}(Z_v^{(t)},U)\leq\epsilon$.*

Lemma B.2 (Sauer-Shelah; (Shalev-Shwartz and Ben-David, 2014)). *If $\mathcal{H}\subset\{0,1,*\}^{\mathcal{X}}$ is a partial concept class with VC dimension d and $X\in\mathcal{X}^m$, then $|\mathcal{H}|_X|\leq(em/d)^d$.*

The following result is a corollary of Freedman's inequality.

Lemma B.3 (Lemma A.3 of (Foster et al., 2021)). *Let $(X_t)_{t\in[T]}$ be a sequence of random variables adapted to a filtration $(\mathcal{F}_t)_{t\in[T]}$. If $0\le X_t\le R$ almost surely for all $t\in[T]$, then with probability at least $1-\delta$,*

$$\sum_{t=1}^{T}\mathbb{E}[X_t\,|\,\mathcal{F}_{t-1}]\le 2\sum_{t=1}^{T}X_t+8R\log(2/\delta).$$

Appendix C. Manipulating a weak ERM oracle

In this section, we prove some lemmas showing that a weak ERM oracle can be used to implement a slightly stronger oracle which, given a dataset $S=\{(x_i,y_i)\}_{i\in[n]}\in(\mathcal{X}\times\mathcal{Y})^n$, gives the values of $h^\star(x_i)$ for $i\in[n]$, where $h^\star$ is a risk-minimizing element of $\mathcal{H}$.

C.1. Weak ERM oracle: binary-valued labels

Algorithm 2 Finding the ERM minimizer on a sample from a weak ERM oracle

Require: Concept class $\mathcal{H}\subset\mathcal{Y}^{\mathcal{X}}$, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$, binary-valued loss function $\ell:\mathcal{Y}\times\mathcal{Y}\to\{0,1\}$, sample $S=\{(x_i,y_i)\}_{i=1}^{n}\in(\mathcal{X}\times\mathcal{Y})^n$.

function $\textsc{SampleERM.Binary}(S,\ell,\mathcal{O}^{\mathsf{erm,w}})$
- Set $\mathcal{I}\leftarrow[n]$.
- **while** There is $i\in\mathcal{I}$ so that $\mathcal{O}^{\mathsf{erm,w}}(\{x_j,y_j'\})\}_{j\in\mathcal{I}}>\mathcal{O}^{\mathsf{erm,w}}(\{(x_j,y_j')\}_{j\in\mathcal{I}\setminus\{i\}})$ **do**
  - Remove such i from $\mathcal{I}$.
- For each $i\in[n]$, set $z_i\leftarrow\mathbb{1}\{i\notin\mathcal{I}\}$.
- Return $(z_1,\ldots,z_n)$.

We begin with the case of binary-valued loss functions.

Lemma C.1. *Consider a concept class $\mathcal{H}\subset\mathcal{Y}^{\mathcal{X}}$ and a binary-valued loss function $\ell:\mathcal{Y}\times\mathcal{Y}\to\{0,1\}$. Let $\mathcal{O}^{\mathsf{erm,w}}$ be a weak ERM oracle for the class $\mathcal{H}$ and loss function ℓ (Definition 2.5). Then for any dataset $S=\{(x_i,y_i)\}_{i\in[n]}\in(\mathcal{X}\times\mathcal{Y})^n$, the algorithm $\texttt{SampleERM.Binary}(S,\ell,\mathcal{O}^{\mathsf{erm,w}})$ (Algorithm 2) makes $O(n^2)$ calls to $\mathcal{O}^{\mathsf{erm,w}}$ and outputs a vector $(z_1,\ldots,z_n)\in\{0,1\}^n$ so that, for some empirical minimizer $h^\star=\mathrm{argmin}_{h\in\mathcal{H}}\sum_{i=1}^{n}\ell(h(x_i),y_i)$, we have $z_i=\ell(h^\star(x_i),y_i)$ for all $i\in[n]$.*

Proof. Fix $\mathcal{H},\ell,\mathcal{O}^{\mathsf{erm,w}}$, and S.

Let the number of iterations of the while loop be denoted N. For $0\le t\le N$, let $\mathcal{I}_t$ denote the value of the set $\mathcal{I}$ in $\texttt{SampleERM.Binary}(S,\ell,\mathcal{O}^{\mathsf{erm,w}})$ directly after round t (so that, in particular, $\mathcal{I}_0=[n]$). Note that each iteration of the while loop, $\mathcal{I}$ decreases in size by 1. Moreover, on round t of the while loop, letting i_t denote the chosen $i\in\mathcal{I}_{t-1}$, we have

$$\min_{h\in\mathcal{H}}\sum_{j\in\mathcal{I}_{t-1}}\ell(h(x_j),y_j)>\min_{h\in\mathcal{H}}\sum_{j\in\mathcal{I}_{t-1}\setminus i_t}\ell(h(x_j),y_j)=\min_{h\in\mathcal{H}}\sum_{j\in\mathcal{I}_t}\ell(h(x_j),y_j)\ge\min_{h\in\mathcal{H}}\sum_{j\in\mathcal{I}_{t-1}}\ell(h(x_j),y_j)-1.$$

It follows that for $1\le t\le N$, $\min_{h\in\mathcal{H}}\sum_{j\in\mathcal{I}_t}\ell(h(x_j),y_j)=\min_{h\in\mathcal{H}}\sum_{j\in[n]}\ell(h(x_j),y_j)-t$.

Note also that we must have $\min_{h \in \mathcal{H}} \sum_{j \in \mathcal{I}_N} \ell(h(x_j), y_j) = 0$, as otherwise we could remove some i from $\mathcal{I}_N$ and decrease the empirical loss. Thus, $N = \min_{h \in \mathcal{H}} \sum_{j \in [n]} \ell(h(x_j), y_j)$. Moreover, there is some $h^\star \in \mathcal{H}$ so that $\ell(h(x_j), y_j) = 0$ for each $j \in \mathcal{I}_N$.

If any $i \in [n] \backslash \mathcal{I}_N$ satisfies $\ell(h^\star(x_i), y_i) = 0$, then we would have $\sum_{j \in [n]} \ell(h(x_j), y_j) < n - |\mathcal{I}_N| = N$, which is a contradiction. Thus, $\ell(h^\star(x_i), y_i) = \mathbb{1}\{i \notin \mathcal{I}_N\}$ for all $i \in [n]$, as desired.

The total number of oracle calls made in Algorithm 2 is at most $2n^2$: we certainly have $N \leq n$, and each round of the while loop requires at most $|\mathcal{I}| + 1 \leq 2n$ calls to $\mathcal{O}^{\mathsf{erm,w}}$. $\square$

C.2. Weak ERM oracle: real-valued labels

Next, we prove an analogue of Lemma C.1 for real-valued loss functions; to keep the oracle complexity bounded, we need to tolerate some approximation error.

Lemma C.2. *Consider a concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$. Let $\mathcal{O}^{\mathsf{erm,w}}$ be a weak ERM oracle for the class $\mathcal{H}$ and absolute loss function ℓ^{abs} (Definition 2.5). Then for any dataset $S = \{(x_i, y_i)\}_{i \in [n]} \in (\mathcal{X} \times [0,1])^n$ and $\gamma \in (0,1)$, the algorithm $\texttt{SampleERM.Real}(S, \ell, \mathcal{O}^{\mathsf{erm,w}}, \gamma)$ (Algorithm 3) makes $O(n/\gamma)$ calls to $\mathcal{O}^{\mathsf{erm,w}}$ of length $O(n)$ and outputs a vector $(z_1, \ldots, z_n) \in [0,1]^n$ so that, for some $h^\star \in \mathcal{H}$ satisfying $\sum_{i=1}^n \ell^{\mathsf{abs}}(h^\star(x_i), y_i) \leq \inf_{h \in \mathcal{H}} \sum_{i=1}^n \ell^{\mathsf{abs}}(h(x_i), y_i)$, we have $|z_i - \ell(h^\star(x_i), y_i)| \leq \gamma$ for all $i \in [n]$.*

Proof. Fix $\mathcal{H}, \mathcal{O}^{\mathsf{erm,w}}, S, \gamma$. For $0 \leq i \leq n$, let $\tilde{S}^{(i)}$ denote the value of $\tilde{S}$ directly after round i, so that, in particular, $\tilde{S}^{(0)} = \emptyset$, and $\tilde{S}^{(i)}$ consists of $n \cdot \lceil 1/\gamma \rceil$ copies of (x_j, y_j'), for each $j \in [i]$. For $1 \leq i \leq n$, let $\Delta^{(i)}$ denote the value of Δ defined on round i. We show the following claim:

Lemma C.3. *For each $0 \leq i \leq n$, the following properties hold for any empirical risk minimizer $h^{(i)} \in \operatorname{argmin}_{h \in \mathcal{H}} \sum_{(x,y) \in S^{(i)} \cup \{(x_j, y_j)\}_{j \in [n]}} \ell^{\mathsf{abs}}(h(x), y)$:*

1. *For each $j \leq i$, $h^{(i)}(x_j) \in [y_j', y_j' + \gamma]$.*
2. *$h^{(i)}$ is an empirical risk minimizer for $S^{(i-1)} \cup \{(x_j, y_j)\}_{j \in [n]}$.*

Proof of Lemma C.3. We prove the claim by induction on i, noting that there is nothing else to establish for the base case $i = 0$. To establish the inductive step, suppose that both claims hold at steps $j < i$, for some $i \in [n]$. Let us write $V_{i-1} := \sum_{(x,y) \in \tilde{S}^{(i-1)} \cup \{(x_j, y_j)\}_{j \in [n]}} \ell^{\mathsf{abs}}(h^{(i-1)}(x), y)$. Taking $\ell = \lfloor h^{(i-1)}(x_i)/\gamma \rfloor$ yields

$$V_{i,\ell} \leq \sum_{(x,y) \in \tilde{S}^{(i-1)} \cup \{(x_j, y_j)\}_{j \in [n]} \cup \{(x_i, \gamma\ell), (x_i, \gamma(\ell+1))\}} \ell^{\mathsf{abs}}(h^{(i-1)}(x), y) \leq V_{i-1} + \gamma,$$

which yields that $V_{i,\ell_i^\star} \leq V_{i-1} + \gamma$. On the other hand, since any function h satisfies $\sum_{(x,y) \in \{(x_i, \gamma\ell), (x_i, \gamma(\ell+1))\}} \ell^{\mathsf{abs}}(h(x), y) \geq \gamma$, we must have that $V_{i,\ell} \geq V_{i-1} + \gamma$ for each ℓ. It follows that $V_i = V_{i,\ell_i^\star} = V_{i-1} + \gamma$, and that any empirical risk minimizer $h^{(i)}$ for $\tilde{S}^{(i)} \cup \{(x_j, y_j)\}_{j \in [n]}$ satisfies the following two properties:

- $h^{(i)}(x_i) \in [\gamma \ell_i^\star, \gamma(\ell_i^\star + 1)] = [y_i', y_i' + \gamma]$.
- $h^{(i)}$ is an empirical risk minimizer on $\tilde{S}^{(i-1)} \cup \{(x_j, y_j)\}_{j \in [n]}$.

Thus the second claim of the lemma statement holds at step i. Moreover, using the inductive hypothesis together with the first item above, we see that $h^{(i)}(x_j) \in [y_j', y_j' + \gamma]$ for all $j < i$. $\square$

Algorithm 3 Finding the ERM minimizer on a sample from a weak ERM oracle

Require: Concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$, sample $S = \{(x_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times [0,1])^n$, accuracy parameter γ so that $1/\gamma \in \mathbb{N}$.

```
function SampleERM.Real(S, γ, O^{erm,w})
    Initialize S̃ ← ∅.
    for 1 ≤ i ≤ n do
        for 0 ≤ ℓ ≤ 1/α − 1 do
            Set V_{i,ℓ} ← O^{erm,w}(S̃ ∪ {(x_j, y_j)}_{j∈[n]} ∪ {(x_i, αℓ), (x_i, α(ℓ+1))}).
        Define ℓ*_i := argmin_{0≤ℓ≤1/α−1l} {V_{i,ℓ}}, and y'_i := α·ℓ*_i.
        Add (x_i, y'_i) and (y'_i + α) to S̃.
    return the vector (y'_1, ..., y'_n).
```

Algorithm 4 Implementing a real-valued consistency oracle with range queries

Require: Concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$, sample $S = \{(x_i, \ell_i, u_i)\}_{i\in[n]} \in (\mathcal{X} \times [0,1]^2)^n$, $i \in [n]$.

```
function SampleCon.Real((S, O^{erm,w}))
    Set S' := ⋃_{i∈[n]} {(x_i, ℓ_i), (x_i, u_i)}.
    Set V ← O^{erm,w}(S').
    return True if V ≤ Σ_{i=1}^n (u_i − ℓ_i), else False.
```

By Lemma C.3, any empirical risk minimizer $h^{(n)}$ for $S^{(n)} \cup \{(x_j, y_j)\}_{j\in[n]}$ satisfies $h^{(n)}(x_i) \in [y'_i, y'_i + \gamma]$ for each $i \in [n]$ and moreover is also an empirical risk minimizer for S. This establishes the claim of Lemma C.2. □

Lemma C.4. *Consider a concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ equipped with a weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$. Then for any $S = \{(x_i, \ell_i, u_i)\}_{i\in[n]} \in (\mathcal{X} \times [0,1]^2)^n$ with $\ell_i \le u_i$ for all i, the algorithm* `SampleCon.Real`$(S, \mathcal{O}^{\mathsf{erm,w}})$ *(Algorithm 4) outputs* True *if and only if there is some $h \in \mathcal{H}$ satisfying $\ell_i \le h(x) \le u_i$ for all $i \in [n]$.*

Proof. The lemma statement is immediate from the fact that there is $h \in \mathcal{H}$ satisfying $\ell_i \le h(x) \le u_i$ if and only if

$$\inf_{h\in\mathcal{H}} \sum_{i=1}^{n} |h(x_i) - \ell_i| + |h(x_i) - u_i| = \sum_{i=1}^{n} (u_i - \ell_i).$$

□

Appendix D. Proof of Theorem 3.2

D.1. Properties of the generating function

Given $m, \mathcal{S} \subset V_m = \{0,1\}^m$, and $v \in V_m$, recall the definition of the hitting time $\tau_{\mathcal{S},v}$ in (4). We begin by proving the following basic recursive property of the generating function $M_{\mathcal{S},v}(\gamma)$ of the random walk on the hypercube graph G_m defined in Section 3.1.

Lemma D.1. *Suppose $\mathcal{W} \subset \{0,1\}^m$ is given, and consider the m-dimensional hypercube graph $G_m = (V,E)$. Then the following holds for all $v \in \mathcal{W}$:*

$$M_{\mathcal{W}^c,v}(\gamma) = \frac{\gamma}{(2-\gamma)m} \sum_{i\in[m]} M_{\mathcal{W}^c,v^{\oplus i}}(\gamma).$$

Proof. For any $v \in \mathcal{W}$ and $t > 0$, we have

$$\Pr(\tau_{\mathcal{W}^c,v} = t) = \frac{1}{2} \cdot \Pr(\tau_{\mathcal{W}^c,v} = t-1) + \frac{1}{2}\sum_{i=1}^m \frac{1}{m} \cdot \Pr(\tau_{\mathcal{W}^c,v^{\oplus i}} = t-1),$$

where we have used the fact that for $v \in \mathcal{W}$, each of the m edges containing v, indexed by $i \in [m]$ has two vertices, namely v and $v^{\oplus i}$. Moreover, for $v \in \mathcal{W}$, we have that $\Pr(\tau_{\mathcal{W}^c,v} = 0) = 0$. Thus, for $\gamma \in (0,1)$, we have

$$\begin{aligned}
M_{\mathcal{W}^c,v}(\gamma) &= \sum_{t\geq 1} \gamma^t \cdot \left(\frac{1}{2} \cdot \Pr(\tau_{\mathcal{W}^c,v} = t-1) + \frac{1}{2}\sum_{i=1}^m \frac{1}{m} \cdot \Pr(\tau_{\mathcal{W}^c,v^{\oplus i}} = t-1) \right) \\
&= \frac{\gamma}{2} \cdot M_{\mathcal{W}^c,v}(\gamma) + \frac{\gamma}{2m} \sum_{i=1}^m \sum_{t\geq 1} \gamma^{t-1} \cdot \Pr(\tau_{\mathcal{W}^c,v^{\oplus i}} = t-1) \\
&= \frac{\gamma}{2} \cdot M_{\mathcal{W}^c,v}(\gamma) + \frac{\gamma}{2m} \sum_{i=1}^m M_{\mathcal{W}^c,v^{\oplus i}}(\gamma).
\end{aligned}$$

Rearranging, we see that

$$M_{\mathcal{W}^c,v}(\gamma) = \frac{\gamma}{(2-\gamma)m} \sum_{i\in[m]} M_{\mathcal{W}^c,v^{\oplus i}}(\gamma),$$

as desired. $\square$

Lemma D.2 establishes an upper bound on the outdegree of the orientation $\sigma_{F,\lambda}$ (defined in (1)) induced by the function $F(v) := M_{\mathcal{W}^c,v}(\gamma)$.

Lemma D.2. *Given $m \in \mathbb{N}$, $\gamma \in (0,1)$, $\lambda \in [0,1]$, and $\mathcal{W} \subset \{0,1\}^m$, write $F(v) := M_{\mathcal{W}^c,v}(\gamma)$ for $v \in \{0,1\}^m$. Then the induced orientation $\sigma_{F,\lambda}$ satisfies*

$$\mathsf{outdeg}(\sigma_{F,\lambda}) \leq \frac{m}{2} - (1-\gamma)\lambda m \cdot \min_{v\in V} F(v).$$

Proof. Consider any $v \in \mathcal{W}$. As a consequence of Lemma D.1, we have

$$m \cdot F(v) - \sum_{i\in[m]} F(v^{\oplus i}) = -m \cdot \frac{2(1-\gamma)}{\gamma} \cdot F(v). \tag{7}$$

We compute

$$
\begin{aligned}
\mathsf{outdeg}(v;\sigma_{F,\lambda}) &= \sum_{i\in[m]} (1-\sigma_{F,\lambda}((i,e_{i,v}))(v)) \\
&= \sum_{i\in[m]} \frac{1-\lambda\cdot(F(v^{\oplus i})-F(v))}{2} \\
&= \frac{m}{2}+\frac{\lambda m}{2}\cdot F(v)-\frac{\lambda}{2}\sum_{i\in[m]} F(v^{\oplus i}) \\
&= \frac{m}{2}-\frac{(1-\gamma)\lambda m}{\gamma}\cdot F(v) \le \frac{m}{2}-(1-\gamma)\lambda m\cdot F(v),
\end{aligned} \tag{8}
$$

where the first equality uses that if $v^{\oplus i}\in\mathcal{W}^{\mathrm{c}}$, then $1-\sigma_{F,\lambda}((i,e_{i,v}))(v)=0\le\frac{1-\lambda(1-F(v))}{2}$, and the final equality uses (7). □

Next, Lemma D.3 lower bounds the function $M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)$, which is needed to apply Lemma D.2.

Lemma D.3. *Let $m\ge 4$ be an integer, $\gamma\in(0,1)$, and $\mathcal{W}\subset\{0,1\}^m$ be given. Then, if $\frac{1}{1-\gamma}\ge 4m\log m$ and $|\mathcal{W}|\le\frac{1}{m}2^{m-2}$,*

$$\min_{v\in\mathcal{W}} M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)\ge\frac{1}{4e}.$$

Proof. Suppose for the purpose of contradiction that there is some $v\in\mathcal{W}$ for which $M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)<1/(4e)$. Let $X_v^{(0)},X_v^{(1)},\ldots\in V$ denote the lazy random walk on the m-dimensional hypercube, G_m, started at v. Since $\gamma^{1/(1-\gamma)}\ge 1/e$ for all $\gamma<1$, we have that

$$\frac{1}{e}\Pr(\tau_{\mathcal{W}^{\mathrm{c}},v}\le\lfloor 1/(1-\gamma)\rfloor)=\sum_{0\le t\le\lfloor 1/(1-\gamma)\rfloor}\gamma^t\cdot\Pr(\tau_{\mathcal{W}^{\mathrm{c}},v}=t)\le M_{\mathcal{W}^{\mathrm{c}},v}(\gamma). \tag{9}$$

Let us write $L:=\lfloor 1/(1-\gamma)\rfloor$ and $\tau=\tau_{\mathcal{W}^{\mathrm{c}},v}$. Note that the distribution of $X_v^{(0)},\ldots,X_v^{(\tau)}$ is exactly the distribution of a lazy random walk $Y_v^{(0)},\ldots,Y_v^{(\tau)}$ on the hypercube $\{0,1\}^m$, up to the stopping time τ. Let U denote a uniformly distributed random variable on $\{0,1\}^m$. By Lemma B.1 together with the fact that $L\ge 1/(2(1-\gamma))\ge 2m\log(m)\ge m\log m+m\log(4)$, we have that $D_{\mathsf{TV}}(Y_v^{(L)},U)\le 1/4$. Let $\bar{X}_v^{(t)}:=X_v^{(t\wedge\tau)}$ denote the stopped random walk, with respect to the stopping time τ.

Consider a coupling between the distributions $\{\bar{X}_v^{(t)}\}_{t\ge 0}$ and $\{Y_v^{(t)}\}_{t\ge 0}$ so that $\bar{X}_v^{(t)}=Y_v^{(t)}$ for all $t\le\tau$, almost surely. Since $\tau>L$ with probability at least $1-e\cdot M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)$ by (9), we have that $\Pr(\bar{X}_v^{(L)}=Y_v^{(L)})\ge\Pr(\tau\ge L)\ge 1-e\cdot M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)$, where the probability is with respect to the coupling. It follows that $D_{\mathsf{TV}}(\bar{X}_v^{(L)},Y_v^{(L)})\le e\cdot M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)<1/4$. By the triangle inequality, we have

$$D_{\mathsf{TV}}(\bar{X}_v^{(L)},U)\le e\cdot M_{\mathcal{W}^{\mathrm{c}},v}(\gamma)+1/4<1/2.$$

Let $\bar{N}(\mathcal{W}):=\{v\in\{0,1\}^m\,:\,v\in\mathcal{W}\text{ or }\exists i\text{ s.t. }v^{\oplus i}\in\mathcal{W}\}$ denote the union of $\mathcal{W}$ and its neighborhood. Thus, we must have $\mathrm{supp}(\bar{X}_v^{(L)})>2^{m-1}$, which contradicts $|\mathcal{W}|\le\frac{1}{m}2^{m-2}$ since $\mathrm{supp}(\bar{X}_v^{(L)})\subset\bar{N}(\mathcal{W})$, and $\bar{N}(\mathcal{W})\le 2m|\mathcal{W}|\le 2^{m-1}$.

□

D.2. Transductive learning guarantee

Proof of Theorem 3.2. Let us write $\gamma := 1 - \frac{1}{C_1 m \log m}$. Set $\epsilon = \frac{1-\gamma}{16e}$ and $L := \lceil \log(2/\epsilon)/\log(1/\gamma) \rceil = \lceil \log(32e/(1-\gamma))/\log(1/\gamma) \rceil \leq O(\log(1/(1-\gamma))/(1-\gamma))$. Moreover, write $\delta = \epsilon/2$ and $U = C_1 m^2 \log^3 m$; note that $U = \Theta(\frac{\log(1/\delta)}{\epsilon^2})$.

Let us write $X = (x_1, \ldots, x_m) \in \mathcal{X}^m$, $y = (y_1, \ldots, y_m) \in \{0,1\}^m$. Let $\mathcal{W} := \mathcal{H}|_X$ be the projection of $\mathcal{H}$ onto $\mathcal{X}$ and $G_m = (V, E)$ denote the m-dimensional hypercube graph. For $y' \in \mathcal{W}$, recall the definition of the stopping time $\tau_{\mathcal{W}^c, y'}$ in (4). Note that, for any $y' \in V$ and each $u \in U$, the random variable T_u constructed in $\texttt{EstimatePotential}(X, y', U, \gamma, \mathcal{O}^{\mathsf{con,w}})$ is distributed exactly according to $L \wedge \tau_{\mathcal{W}^c, y'}$. Thus, for any δ, U satisfying $U \geq C \log(1/\delta)/\epsilon^2$ for a sufficiently large constant C, we have from Hoeffding's inequality that with probability $1-\delta$,

$$\left| \frac{1}{U} \sum_{u=1}^{U} \gamma^{T_u} - \mathbb{E}[\gamma^{L \wedge \tau_{\mathcal{W}^c, y'}}] \right| \leq \epsilon/2. \tag{10}$$

Moreover, by our choice of L, we have that, almost surely,

$$\left| \gamma^{L \wedge \tau_{\mathcal{W}^c, y'}} - \gamma^{\tau_{\mathcal{W}^c, y'}} \right| \leq \gamma^L \leq \epsilon/2. \tag{11}$$

Thus, combining Eqs. (10) and (11), with probability at least $1-\delta$, the output of $\texttt{EstimatePotential}(X, y', U, \gamma, \mathcal{O}^{\mathsf{con,w}})$ satisfies

$$\left| \frac{1}{U} \sum_{u=1}^{U} \gamma^{T_u} - M_{\mathcal{W}^c, y'}(\gamma) \right| = \left| \frac{1}{U} \sum_{u=1}^{U} \gamma^{T_u} - \mathbb{E}[\gamma^{\tau_{\mathcal{W}^c, y'}}] \right| \leq \epsilon. \tag{12}$$

For $y' \in \mathcal{W}$, define $F(y') := M_{\mathcal{W}^c, y'}(\gamma)$. For each $i \in [m]$, write $y^{i,0} = (y_{-i}, 0)$ and $y^{i,1} = (y_{-i}, 1)$. Now consider $i \in [m]$ for which $y^{i,0}, y^{i,1} \in \mathcal{W}$. Note that $\texttt{WeakRealizable}(S_{-i}, x_i, \gamma, \lambda, \mathcal{O}^{\mathsf{con,w}})$ calls $\texttt{EstimatePotential}(X, y^0, U, \gamma, \mathcal{O}^{\mathsf{con,w}})$ and $\texttt{EstimatePotential}(X, y^1, U, \gamma, \mathcal{O}^{\mathsf{con,w}})$. These calls return values $\hat{F}(y^{i,0}), \hat{F}(y^{i,1}) \in [0,1]$ respectively. Since can ensure, by our choices of the values U, δ above, that $U \geq C \log(1/\delta)/\epsilon^2$ (by making C_1 sufficiently large), it follows by (12) and a union bound that with probability at least $1 - 2\delta$, for each $b \in \{0,1\}$,

$$\left| \hat{F}(y^{i,b}) - F(y^{i,b}) \right| = \left| \hat{F}(y^{i,b}) - M_{\mathcal{W}^c, y^{i,b}}(\gamma) \right| \leq \epsilon.$$

Thus, with probability at least $1-2\delta$, the output $\hat{y}_i := \frac{1 + (\hat{F}(y^{i,0}) - \hat{F}(y^{i,1}))}{2}$ of $\texttt{WeakRealizable}(S_{-i}, x_i, \gamma, 1, \mathcal{O}^{\mathsf{con,w}})$ satisfies

$$\begin{aligned} |\hat{y}_i - \sigma_{F,1}(e_{i,y})(y^{i,1})| &= \left| \frac{1 + (\hat{F}(y^{i,0}) - \hat{F}(y^{i,1}))}{2} - \frac{1 + (F(y^{i,0}) - F(y^{i,1}))}{2} \right| \\ &\leq \frac{1}{2} \cdot |\hat{F}(y^{i,0}) - F(y^{i,0})| + \frac{1}{2} \cdot |\hat{F}(y^{i,1}) - F(y^{i,1})| \leq \epsilon, \end{aligned}$$

and thus, using that $y = y^{i, y_i}$,

$$|\hat{y}_i - y_i| = |\hat{y}_i - \sigma_{F,1}(e_{i,y})(y^{i,1})| + |\sigma_{F,1}(e_{i,y})(y^{i,1}) - y_i| \leq \epsilon + (1 - \sigma_{F,1}(e_{i,y})(y)).$$

By our choice of $\delta = \epsilon/2$, it follows that, for each $i \in [m]$,

$$\mathbb{E}\left[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i};x_i),y_i)\right] = \mathbb{E}[|\hat{y}_i - y_i|] \le 2\delta + \epsilon + (1 - \sigma_{F,1}(e_{i,y})(y)) = 2\epsilon + (1 - \sigma_{F,1}(e_{i,y})(y)). \tag{13}$$

Next, the Sauer-Shelah lemma (Lemma B.2) gives that $|\mathcal{W}| = |\mathcal{H}|_X| \le (em/d)^d \le \frac{1}{m}2^{m-2}$, since we have chosen $m \ge C_1 d \log d$ for a sufficiently large constant C_1. Thus, by our choice of $\gamma = 1 - \frac{1}{C_1 m \log m}$ and Lemma D.3 we have that $\min_{v \in \mathcal{W}} F(v) = \min_{v \in \mathcal{W}} M_{\mathcal{W}^c,v}(\gamma) \ge 1/(4e)$. We may now compute

$$\begin{aligned}\sum_{i \in [m]} \mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i};x_i),y_i)] &\le 2\epsilon m + \sum_{i \in [m]} (1 - \sigma_{F,1}(e_{i,y})(y)) \\ &= 2\epsilon m + \mathsf{outdeg}(y;\sigma_{F,1}) \\ &\le 2\epsilon m + \frac{m}{2} - \frac{(1-\gamma)m}{4e} \le \frac{m}{2} - \frac{(1-\gamma)m}{16e},\end{aligned}$$

where the first inequality uses (13), the second inequality uses Lemma D.2, and the final inequality uses the choice of $\epsilon = \frac{1-\gamma}{16e}$. □

Finally, for use in applying Theorem 3.2, we state the following standard lemma, which relates the transductive error of a learning algorithm $\mathscr{A}$ to its expected error with respect to any realizable distribution.

Lemma D.4 (Leave-one-out). *Let $P \in \Delta(\mathcal{X} \times \{0,1\})$ be $\mathcal{H}$-realizable and $m \in \mathbb{N}$ be given. Furthermore, let $\mathscr{A}(\cdot,\cdot) : (\mathcal{X} \times \{0,1\})^{m-1} \times \mathcal{X} \to \{0,1\}$ be a (possibly randomized) mapping which takes as input a dataset of size $m-1$ and a point in $\mathcal{X}$, and outputs a real number. Then*

$$\mathop{\mathbb{E}}_{S' \sim P^{m-1}} \mathop{\mathbb{E}}_{(X,Y) \sim P} \mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S',X),Y)] = \mathop{\mathbb{E}}_{\substack{S \sim P^m \\ S = \{(x_i,y_i)\}_{i \in [m]}}} \left[\frac{1}{m}\sum_{i=1}^m \mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i},x_i),y_i)]\right],$$

where the inner expectation is over the randomness in $\mathscr{A}$.

Proof. Let $(x_1,y_1),\dots,(x_m,y_m),(X,Y)$ denote i.i.d. samples from P, and write $S' = \{(x_i,y_i)\}_{i \in [m-1]}, S = \{(x_i,y_i)\}_{i \in [m]}$. By exchangeability of these samples and linearity of expectation, we have

$$\begin{aligned}\mathbb{E}_{S'}\mathbb{E}_{(X,Y)\sim P}\mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S',X),Y)] &= \mathbb{E}_{S'}\mathbb{E}_{(x_m,y_m)\sim P}\mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S',x_m),y_m)] \\ &= \frac{1}{m}\sum_{i=1}^m \mathbb{E}_S \mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i},x_i),y_i)] \\ &= \mathbb{E}_S\left[\frac{1}{m}\sum_{i=1}^n \mathbb{E}[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i},x_i),y_i)]\right].\end{aligned}$$

□

Appendix E. Boosting

In this section, we discuss the technique of boosting, which is used to upgrade a weak learner (in the sense of Definition 3.1) to a strong learner, i.e., one which achieves error at most an arbitrary threshold $\epsilon \in (0,1)$ with high probability. Notice that we allow a weak learner to only possess its guarantee (2) *in expectation*, rather than with high probability. To account for this weaker assumption, it is necessary to slightly modify standard boosting results (Schapire and Freund, 2012, Chapters 3 & 4), as stated in Lemma E.1 below. The main difference in the proofs is the use of an appropriate martingale concentration inequality (namely, Lemma B.3) to deal with the deviations in errors exhibited by the individual calls to the weak learner by the boosting algorithm. Lemma E.1 gives a bound on the training error of the `Adaboost` algorithm (Algorithm 5).

Algorithm 5 `Adaboost` (Algorithm 1.1 of (Schapire and Freund, 2012))

Require: Input dataset $\{(x_i,y_i)\}_{i\in[n]}\subset(\mathcal{X}\times\{0,1\})^n$, randomized weak learner $\mathscr{A}$, number of time steps T.

1: Initialize $D_1:=\mathrm{Unif}([n])\in\Delta([n])$.
2: **for** $1\leq t\leq T$ **do**
3: Sample a fresh string of uniform bits R_t for use in $\mathscr{A}$ and to sample S_t in Line 4 below.
4: Sample an i.i.d. dataset S_t of size m from the distribution of x_i, $i\sim D_t$, so that $S_t\in(\mathcal{X}\times\{0,1\})^m$.
5: Let $h_t:\mathcal{X}\to\{0,1\}$ be the output of $\mathscr{A}_{R_t}(S_t,\cdot)$.
6: Define $\epsilon_t:=\Pr_{i\sim D_t}(h_t(x_i)\neq y_i)$, and $\alpha_t:=\frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
7: For $i\in[n]$, define

$$D_{t+1}(i):=\frac{D_t(i)\cdot\exp(-\alpha_t\cdot(2y_i-1)\cdot(2h_t(x_i)-1))}{Z_t},\tag{14}$$

where $Z_t:=\sum_{j\in[n]}D_t(j)\cdot\exp(-\alpha_t\cdot(2y_j-1)\cdot(2h_t(x_j)-1))$.
8: **return** the hypothesis $H:\mathcal{X}\to\{0,1\}$, where $H(x):=\frac{1}{2}+\frac{1}{2}\mathrm{sign}\left(\sum_{t=1}^T\alpha_t\cdot(2h_t(x)-1)\right)$.

Lemma E.1 (Training error of `Adaboost`)**.** *Let $m,n\in\mathbb{N}$ and $\eta\in(0,1)$ be given, and suppose algorithm $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathcal{H}$. Let $\bar{S}\in(\mathcal{X}\times\{0,1\})^n$ be an $\mathcal{H}$-realizable sample. Then if Algorithm 5 is run for $T\geq\lceil 16\log(2n/\delta)/\eta^2\rceil$ rounds on $\bar{S}$, the output hypothesis $H(x)$ satisfies $\widehat{\mathrm{er}}_S(H)=0$ with probability $1-\delta$.*

Moreover, for $x\in\mathcal{X}$, to compute $H(x)$, one must only call $\mathscr{A}(S_t,x)$ for T different choices of datasets $S_t\in(\mathcal{X}\times\{0,1\})^m$.

Proof. The proof follows closely to that in, e.g., (Schapire and Freund, 2012, Chapter 3), with minor modifications. We use the notation in Algorithm 5. Define $F(x):=\sum_{t=1}^T\alpha_t h_t(x)$. From the definition of Z_t in Algorithm 5 we have, for $i\in[m]$,

$$D_{T+1}(i)=\frac{D_1(i)\cdot\exp(-y_i\cdot F(x_i))}{Z_1\cdots Z_T}.$$

For all $(x,y)\in\mathcal{X}\times\{\pm1\}$, we have $\mathbb{1}\{H(x)\neq y\}\leq e^{-F(x)\cdot y}$. We can thus bound the training error over the dataset $\{(x_i,y_i)\}_{i\in[n]}$ as follows:

$$\frac{1}{n}\sum_{i=1}^n\mathbb{1}\{H(x_i)\neq y_i\}\leq\sum_{i=1}^n D_1(i)\cdot e^{-F(x_i)\cdot y_i}=\sum_{i=1}^n D_{T+1}(i)\cdot(Z_1\cdots Z_T)=Z_1\cdots Z_T.\tag{15}$$

By the choice of $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ in Line 6 of Algorithm 5, we have

$$Z_t = \sum_{j=1}^{n} D_t(j)\cdot e^{-\alpha_t \cdot y_j h_t(x_j)} = (1-\epsilon_t)\cdot e^{-\alpha_t} + \epsilon_t \cdot e^{\alpha_t} = 2\sqrt{\epsilon_t(1-\epsilon_t)} = \sqrt{1-4\gamma_t^2},$$

where the second equality uses the fact that $\sum_{j: y_j h_t(x_j)=1} D_t(j) = 1-\epsilon_t$ and $\sum_{j: y_j h_t(x_j)=-1} D_t(j) = \epsilon_t$, and we have written $\gamma_t := 1/2 - \epsilon_t$ for the final equality.

For each $t \in [T]$, let $\mathcal{F}_t$ denote the sigma-algebra generated by $\{(S_s, h_s)\}_{1\le s\le t}$. Note that D_t is measurable with respect to $\mathcal{F}_{t-1}$. Since the distribution over (x_i, y_i), for $i \sim D_t$, is $\mathcal{H}$-realizable, the fact that $\mathscr{A}$ is an m-sample weak learner with margin η yields that for each t, $\mathbb{E}[\epsilon_t \,|\, \mathcal{F}_{t-1}] \le \frac{1}{2} - \eta$, i.e., $\mathbb{E}[\gamma_t \,|\, \mathcal{F}_{t-1}] \ge \eta$. By Jensen's inequality it follows that $\mathbb{E}[\gamma_t^2 \,|\, \mathcal{F}_{t-1}] \ge \eta^2$. Note that $\gamma_t^2 \in [0,1]$ for all $t \in [T]$. Then by Lemma B.3 with $R=1$, there is an event $\mathcal{E}$ occurring with probability $1-\delta$ so that, under $\mathcal{E}$, $\sum_{t=1}^{T} \gamma_t^2 \ge \frac{1}{2} T\eta^2 - 4\log(2/\delta)$. Thus, under the event $\mathcal{E}$, we have

$$Z_1 \cdots Z_T = \left(\prod_{t=1}^{T}(1-4\gamma_t^2)\right)^{1/2} \le e^{-2(\gamma_1^2+\cdots+\gamma_T^2)} \le e^{-T\eta^2 + 8\log(2/\delta)} \le e^{-T\eta^2/2} \le \frac{1}{2n}, \tag{16}$$

where the second-to-last inequality above holds since we have chosen T to satisfy $T \ge \frac{16\log(2/\delta)}{\eta^2}$, and the final inequality holds since T also satisfies $T \ge \frac{2\log(2n)}{\eta^2}$. Combining the above display and (15), we obtain that, under $\mathcal{E}$, $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{H(x_i) \ne y_i\} \le 1/(2n)$, which implies that $H(x_i) = y_i$ for all $i \in [n]$. $\square$

E.1. Generalization error of `Adaboost`

Next, using the technique of *sample compression schemes*, and in particular their connection to generalization (Lemma A.1), we prove that the output hypothesis H of `Adaboost` generalizes well. Proposition E.2 carries out this argument for the realizable setting, and Proposition E.3 does so for the agnostic setting. Even in the agnostic setting, the input dataset for `Adaboost` must still be realizable by $\mathcal{H}$, as the weak learner's guarantee depends on realizability. Thus, in the statement of Proposition E.3, we assume that `Adaboost` is passed a subsample of maximum size which is $\mathcal{H}$-realizable. Our final algorithm which agnostically learns partial concept classes (Algorithm 6) will find such a subsample to pass to `Adaboost` using a weak ERM oracle.

Proposition E.2 (Generalization error of Adaboost – realizable setting)**.** *Let $m,n \in \mathbb{N}$ and $\eta,\delta \in (0,1)$ be given, and suppose algorithm $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathcal{H}$. Let $P \in \Delta(\mathcal{X}\times\{0,1\})$ be $\mathcal{H}$-realizable. Then if Algorithm 5 is run for $T = \lceil 16\log(4n/\delta)/\eta^2 \rceil$ rounds on a dataset $\bar{S} \sim P^n$, the random output hypothesis $H \in \{0,1\}^{\mathcal{X}}$ satisfies the following with probability $1-\delta$:*

$$\mathrm{er}_P(H) \le O\left(\frac{\log^2(n/\delta)\cdot(m+\log n)}{\eta^2 n}\right).$$

In particular, the probability is over the draw of $\bar{S}$.

Proof. We use a compression-based argument, following (Schapire and Freund, 2012, Chapter 4.2). Let us denote the input dataset to Algorithm 5 by $\bar{S} \in (\mathcal{X}\times\{0,1\})^n$. We consider a distribution Q over sample compression schemes on n-sample datasets, (κ,ρ), defined as follows. Given a dataset $\bar{S}$, let S_t, α_t, R_t, for

$t \in [T]$, be the random variables generated in the course of the procedure in Algorithm 5. Since the bits R_t are used for the sampling step in Line 4 and by $\mathscr{A}$ (where the portions of R_t that are used for the two tasks are independent), given a dataset $\bar{S}$, the random variable $(S_t,\alpha_t)_{t\in[T]}$ is a deterministic function of $(R_t)_{t\in[T]}$ and $\bar{S}$. Then we define (ρ,κ) to be the distributed as the following (deterministic) function of $(R_t)_{t\in[T]}$:

- κ maps a dataset $\bar{S} \in (\mathcal{X}\times\{0,1\})^n$ to $\kappa(\bar{S}) = ((S_1,...,S_T),(\alpha_1,...,\alpha_T))$, where S_t,α_t are defined as a function of $(R_t)_{t\in[T]}$ and $\bar{S}$, as described above.
- ρ maps an input of the form $((S'_1,...,S'_T),(\alpha'_1,...,\alpha'_T))$ (where $(S'_1,...,S'_T)$ is a sequence of examples in $\mathcal{X}\times\{0,1\}$ of length Tm and $(\alpha'_1,...,\alpha'_T)$ is a sequence of real numbers, encoded in binary) to the hypothesis $x \mapsto \operatorname{sign}\left(\sum_{t=1}^T \alpha_t \cdot \mathscr{A}_{R_t}(S_t,x)\right)$.

Since the values ϵ_t in Algorithm 5 lie in $\{0,1/n,2/n,...,1\}$, each parameter α_t can be encoded with $O(\log n)$ bits. Thus, with probability 1 over $(\kappa,\rho)\sim Q$, we have that the size of κ (for input samples $\bar{S}$ of size n) is $|\kappa| \le O(T\cdot(m+\log(n)))$. Next, Lemma E.1 establishes that, for any fixed $\mathcal{H}$-realizable $\bar{S}$, there is a set $\mathcal{E}$ of compression schemes satisfying $Q(\mathcal{E}) \ge 1-\delta$ so that, for all $(\kappa,\rho)\in\mathcal{E}$, $\widehat{\operatorname{er}}_{\bar{S}}(\rho(\kappa(\bar{S})))=0$. (Here we crucially use that the output hypothesis of ρ depends on the same random bits R_t used to generate the sequence $(S_t,\alpha_t)_{t\in[T]}$.) Moreover, by Lemma A.1 and our bound on the size of κ drawn from Q, there is a constant $C>0$ so that the following holds for any fixed $(\kappa,\rho)\in\operatorname{supp}(Q)$: with probability $1-\delta$ over the draw of $\bar{S}\sim P^n$,

$$\operatorname{er}_P(\rho(\kappa(\bar{S}))) \le \mathbb{1}\{\widehat{\operatorname{er}}_{\bar{S}}(\rho(\kappa(\bar{S})))=0\} + \frac{C}{n}\cdot\left(T\log(n)\cdot(m+\log n)+\log\frac{1}{\delta}\right).$$

By our choice of T together with the fact that $Q(\mathcal{E})\ge 1-\delta$, it follows that with probability $1-2\delta$ over the draw of $\bar{S}\sim P^n$, we have

$$\operatorname{er}_P(\rho(\kappa(\bar{S}))) \le C\cdot\frac{\log(2n^2)\log(n)\cdot\frac{1}{\eta^2}\cdot(m+\log n)+\log(1/\delta)}{n}.$$

Since, for fixed $\bar{S}$, the distribution of $\rho(\kappa(\bar{S}))$ is the distribution of the output hypothesis H of Algorithm 5, the claim of the proposition follows after rescaling δ. $\square$

Proposition E.3 (Generalization error of Adaboost – agnostic setting)**.** *Let $m,n\in\mathbb{N}$ and $\eta,\delta\in(0,1)$ be given, and suppose algorithm $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathcal{H}$. Let $P\in\Delta(\mathcal{X}\times\{0,1\})$ be an arbitrary distribution. Consider a procedure which samples a dataset $\bar{S}\sim P^n$, deterministically chooses a subsample $\tilde{S}\subset\bar{S}$ of maximum size which is realizable by $\mathcal{H}$, and then runs Algorithm 5 for $T=\lceil 16\log(4n/\delta)/\eta^2\rceil$ rounds on $\tilde{S}$. Then the output hypothesis $H(x)$ satisfies the following with probability $1-\delta$:*

$$\operatorname{er}_P(H) \le \inf_{h\in\mathcal{H}} \widehat{\operatorname{er}}_{\bar{S}}(h) + O\left(\sqrt{\frac{\log^2(n/\delta)\cdot(m+\log n)}{\eta^2 n}}\right).$$

Proof. The proof closely follows that of Proposition E.2. In particular, we consider the same distribution Q over sample compression schemes on datasets with (at most) n-samples. We again have that with probability 1 over $(\kappa,\rho)\sim Q$, $|\kappa|\le O(T\cdot(m+\log n))$, and that, for any fixed $\mathcal{H}$-realizable $\tilde{S}$, there is a set $\mathcal{E}$ of compression schemes satisfying $Q(\mathcal{E})\ge 1-\delta$ so that, for all $(\kappa,\rho)\in\mathcal{E}$, $\widehat{\operatorname{er}}_{\tilde{S}}(\rho(\kappa(\tilde{S})))=0$.

Next, let the deterministic mapping from samples $\bar{S} \in (\mathcal{X} \times \{0,1\})^n$ to $\tilde{S}$ be denoted by Σ. For any compression scheme $(\kappa, \rho) \in \operatorname{supp}(Q)$, note that $(\kappa \circ \Sigma, \rho)$ is a compression scheme of size $|\kappa \circ \Sigma| \leq |\kappa| \leq O(T \cdot (m + \log n))$. Thus, by Lemma A.1, there is a constant $C > 0$ so that, for any fixed $(\kappa,\rho) \in \operatorname{supp}(Q)$, with probability $1-\delta$ over the draw of $\bar{S} \sim P^n$,

$$\operatorname{er}_P(\rho(\kappa(\Sigma(\bar{S})))) \leq \widehat{\operatorname{er}}_{\bar{S}}(\rho(\kappa(\Sigma(\bar{S})))) + C\sqrt{\frac{1}{n}(T\log(n)\cdot(m+\log n)+\log(1/\delta))}.$$

The choice of Σ yields that, if $\bar{S} \in (\mathcal{X} \times \{0,1\})^n$ and $\tilde{S} = \Sigma(\bar{S})$, then for any hypothesis h' satisfying $\widehat{\operatorname{er}}_{\tilde{S}}(h') = 0$, we have

$$\widehat{\operatorname{er}}_{\bar{S}}(h') \leq \frac{n - |\tilde{S}|}{n} = \inf_{h \in \mathcal{H}} \widehat{\operatorname{er}}_{\bar{S}}(h) \tag{17}$$

Thus, by our choice of T, (17), and the fact that $Q(\mathcal{E}) \geq 1-\delta$, it follows that with probability $1-2\delta$ over the draw of $\bar{S} \sim P^n$,

$$\operatorname{er}_P(\rho(\kappa(\Sigma(\bar{S})))) \leq \inf_{h \in \mathcal{H}} \widehat{\operatorname{er}}_{\bar{S}}(h) + O\left(\sqrt{\frac{\log^2(n/\delta)\cdot(m+\log n)}{\eta^2 n}}\right).$$

Since for fixed $\bar{S}$, the distribution of $\rho(\kappa(\Sigma(\bar{S})))$ is the distribution of the output hypothesis H of Algorithm 5, the claim follows after rescaling δ. $\square$

Appendix F. Proof of Theorem 3.1

Algorithm 6 Oracle-efficient PAC learner for partial concept classes

Require: Partial concept class $\mathcal{H}$, $n \in \mathbb{N}$, sample $S = \{(x_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \{0,1\})^n$, weak learner $\mathscr{A}$ for $\mathcal{H}$ with margin $\eta \in (0,1)$, failure probability $\delta \in (0,1)$, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$, domain point $x \in \mathcal{X}$.

function $\textsc{RealizablePartial}(S, x, \mathscr{A}, \eta, \delta)$
- Call `Adaboost` (Algorithm 5) on the dataset S using the weak learner $\mathscr{A}$ with $T = \lceil 16 \log(4n/\delta)/\eta^2 \rceil$, and denote its output hypothesis by $H : \mathcal{X} \to \{0,1\}$.
- **return** $H(x)$.

function $\textsc{AgnosticPartial}(S, x, \mathscr{A}, \eta, \delta, \mathcal{O}^{\mathsf{erm,w}})$
- Let $z \in \{0,1\}^n$ be the output of `SampleERM.Binary`$(S, x, \ell^{\mathsf{bin}}, \mathcal{O}^{\mathsf{erm,w}})$. ▷ *(Algorithm 2)*
- Define $\tilde{S} := \{(x_i, y_i) : z_i = 0\}$.
- Call `Adaboost` (Algorithm 5) on the dataset $\tilde{S}$ using the weak learner $\mathscr{A}$ with $T = \lceil 16 \log(6n/\delta)/\eta^2 \rceil$, and denote its output hypothesis by $H : \mathcal{X} \to \{0,1\}$.
- **return** $H(x)$.

The guarantees of Theorem 3.1 are established with the algorithms `RealizablePartial` and `AgnosticPartial` (Algorithm 6). These algorithms call `Adaboost` on an appropriate $\mathcal{H}$-realizable dataset and use `WeakRealizable` as the weak learner. We remark that `AgnosticPartial` uses the algorithm `SampleERM.Binary` (defined in Appendix C) to find the largest subset of labels which can be realized by the class $\mathcal{H}$. Note that in the algorithms' descriptions we have stated that `Adaboost` returns a hypothesis $H:\mathcal{X}\to\{0,1\}$. `RealizablePartial` and `AgnosticPartial` never have to compute this entire hypothesis H, and instead only have to evaluate $H(x)$, which, as we shall show, can be done using few calls to the oracle $\mathcal{O}^{\mathsf{con,w}}$ or $\mathcal{O}^{\mathsf{erm,w}}$, respectively.

Proof of Theorem 3.1. Let $d_{\mathsf{VC}}\in\mathbb{N},\epsilon,\delta\in(0,1)$ be given. Let C_1,C_2 denote the constants of Theorem 3.2, $m:=C_1 d_{\mathsf{VC}}\log d_{\mathsf{VC}}$, and $\eta:=\frac{1}{C_2 m\log m}$. Let $\mathscr{A}$ denote the randomized mapping $\mathscr{A}:(\mathcal{X}\times\{0,1\})^m\times\mathcal{X}\to\{0,1\}$ which, given as input (S,x), returns `WeakRealizable`$(S,x,\frac{1}{C_1 m\log m},1,C_1 m^2\log^3 m,\mathcal{O}^{\mathsf{con,w}})$ (see Algorithm 1). By Theorem 3.2 and Lemma D.4, $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathcal{H}$.

Realizable setting. We take $n=\frac{d_{\mathsf{VC}}^3\cdot\log(1/\delta)}{\epsilon}\cdot(c\log(d_{\mathsf{VC}}\log(1/\delta)/\epsilon))^c$, for a sufficiently large constant c as specified below. The output hypothesis H of `Adaboost` in `RealizablePartial` is a deterministic function of S and the random bits R used in `Adaboost` (including in its calls to $\mathscr{A}$). By Lemma E.1, for any S, with probability $1-\delta$ over the draw of R, we have $\widehat{\mathrm{er}}_S(H)=0$. Moreover, by Proposition E.2, with probability $1-\delta$ over the draw of S,R, we have

$$\mathrm{er}_P(H)\le O\left(\frac{\log^2(n/\delta)\cdot(m+\log n)}{\eta^2 n}\right).$$

Combining the above inequality with our choice of n and rescaling ϵ,δ, we obtain that with probability $1-\delta$ over the draw of $S\sim P^n$, $\mathrm{er}_P(H)\le\epsilon$.

Finally, we analyze the oracle complexity of `RealizablePartial`: to compute the value of $H(x)$, we need to compute the values $h_t(x)$ for the hypotheses h_t, $t\in[T]$, defined in `Adaboost`. Given the value of S_t, each computation of $h_t(x)$, for any $x\in\mathcal{X}$, requires a single run of $\mathscr{A}(S_t,x)$, which requires $\tilde{O}(m^3)$ calls to $\mathcal{O}^{\mathsf{con,w}}$ with datasets of size m (Theorem 3.2). In turn, the datasets S_t are computed inductively as follows: given S_t, we can compute $h_t(x_i)$ for each $i\in[n]$, which requires $\tilde{O}(nm^3)$ calls to $\mathcal{O}^{\mathsf{con,w}}$. This in turn allows us to compute D_{t+1} (per (14)), which then allows us to sample S_{t+1}. Thus, overall, we need $\tilde{O}(nm^3\cdot T)\le\mathrm{poly}(n)$ calls to $\mathcal{O}^{\mathsf{con,w}}$ to compute $H(x)$, each of which uses a dataset of size at most n. Hence the cumulative oracle cost is $\mathrm{poly}(n)$.

Agnostic setting. We take $n=\frac{d_{\mathsf{VC}}^3\cdot\log(1/\delta)}{\epsilon^2}\cdot(c\log(d_{\mathsf{VC}}\log(1/\delta)/\epsilon))^c$, for a sufficiently large constant c as specified below. The output hypothesis H of `Adaboost` in `AgnosticPartial` is a deterministic function of S as well as the random bits R used in `Adaboost` (including in its calls to $\mathscr{A}$). By Lemma C.1, the set $\tilde{S}$ constructed from S in `AgnosticPartial` is a subset of S of maximum size which is realizable by $\mathcal{H}$. Thus, by Lemma E.1, for any S, with probability $1-\delta$ over the draw of R, the hypothesis H satisfies $\widehat{\mathrm{er}}_{\tilde{S}}(H)=0$.

Moreover, by Proposition E.3, the (random) output hypothesis H of `Adaboost` satisfies the below with probability $1-\delta$ over the draw of S,R:

$$\mathrm{er}_P(H)\le\inf_{h\in\mathcal{H}}\widehat{\mathrm{er}}_S(h)+O\left(\sqrt{\frac{\log^2(n/\delta)\cdot(m+\log n)}{\eta^2 n}}\right). \tag{18}$$

Since $n \ge \frac{C\log 1/\delta}{\epsilon^2}$ for a sufficiently large constant C, McDiarmid's inequality yields that with probability $1-\delta$ over the choice of $S \sim P^n$,

$$\inf_{h\in\mathcal{H}} \widehat{\mathrm{er}}_S(h) \le \mathbb{E}_{S'\sim P^n}\left[\inf_{h'\in\mathcal{H}} \widehat{\mathrm{er}}_{S'}(h')\right] + \epsilon \le \mathrm{er}_P(\mathcal{H}) + \epsilon. \tag{19}$$

Combining Eqs. (18) and (19) with our choice of n and rescaling ϵ,δ, we obtain that with probability $1-\delta$ over the draw of $S \sim P^n$, we have $\mathrm{er}_P(H) \le \mathrm{er}_P(\mathcal{H}) + \epsilon$.

Finally, note that, to compute the value $H(x)$, for any $x \in \mathcal{X}$, we need a single call to $\mathcal{O}^{\mathsf{erm,w}}$ to compute $\tilde{S}$ (Line 6 of Algorithm 6), and then need to compute the values $h_t(x)$ for the hypotheses h_t, $t \in [T]$, defined in `Adaboost`. The oracle complexity of these calls are analyzed in the same manner as in the realizable case, so we again obtain $\mathrm{poly}(n)$ cumulative oracle cost. □

Appendix G. Multiclass classification

In this section, we generalize our results on binary classification to the setting of multiclass classification. We begin by establishing that a variant of `WeakRealizable` yields a weak learner in the multiclass setting, in an appropriate sense. It is less clear how to define a "weak learner" in the multiclass setting than in the binary setting, and the literature on multiclass boosting has identified several possible definitions (see (Schapire and Freund, 2012, Chapter 10) as well as many more recent works (Mukherjee and Schapire, 2013; Brukhim et al., 2023a, 2021, 2022, 2023b)). Our approach proceeds by defining a *partial binary* concept class given any multiclass classification problem, in Definition G.1 below. We will then feed our weak learner for partial binary classes (`WeakRealizable`; Algorithm 1) into `Adaboost`, and finally show how to translate good performance of the boosted learner back to good performance for the original multiclass problem.

Our approach is similar to the one taken in (Schapire and Freund, 2012, Chapter 11) (which originally appeared in (Schapire and Singer, 1998)), where multiclass classification is reduced to boosting with rankings and the `Adaboost.MR` algorithm is used. However, our approach is syntactically different since the weak learner for a menu class (in the sense of Definition G.1) takes as input *two labels* together with the covariate x, and must determine which of them is the correct label, in contrast to (Schapire and Singer, 1998) where the weak learner takes as input a single label together with x and outputs a scalar indicating how likely the label is to be correct.

Definition G.1 (Menu class)**.** Consider a hypothesis class $\mathcal{H} \subset [K]^{\mathcal{X}}$, and let $\mathcal{P}_2(K)$ denote the set of all length-2 vectors consisting of distinct elements of $[K]$. We refer to the elements of $\mathcal{P}_2(K)$ as *menus*.[8] Given $\mu \in \mathcal{P}_2(K), k \in [K]$, we write $k \in \mu$ to mean that k is one of the two elements of μ.

Given $h \in \mathcal{H}$ and $(x,(\ell,k)) \in \mathcal{X} \times \mathcal{P}_2(K)$, we define

$$h^{\mathsf{menu}}(x,(\ell,k)) := \begin{cases} 0 & : h(x) = \ell \\ 1 & : h(x) = k \\ * & : \text{otherwise.} \end{cases}$$

we let $\mathscr{M}_2(\mathcal{H}) \subset \{0,1,*\}^{\mathcal{X}\times\mathcal{P}_2(K)}$ denote the (binary) partial hypothesis class defined by

$$\mathscr{M}_2(\mathcal{H}) := \{(x,\mu) \mapsto h^{\mathsf{menu}}(x,\mu) \,:\, h \in \mathcal{H}\}.$$

8. This terminology is inspired by (Brukhim et al., 2022, 2023b), which considered such menus, though used them together with techniques distinct from ours.

We often refer to $\mathscr{M}_2(\mathcal{H})$ as the *menu class* of $\mathcal{H}$.

The below definition gives a procedure to map hypotheses in $\{0,1,*\}^{\mathcal{X}\times\mathcal{P}_2(K)}$ to multiclass hypotheses in $[K]^{\mathcal{X}}$.

Definition G.2 (Multiclass decoder)**.** We define a map $\mathsf{Dec}^{\mathsf{mc}}:\{0,1,*\}^{\mathcal{X}\times\mathcal{P}_2(K)}\to[K]^{\mathcal{X}}$, as follows. For $J:\mathcal{X}\times\mathcal{P}_2(K)\to\{0,1,*\}$, we define $\mathsf{Dec}^{\mathsf{mc}}(J):=H$, where $H(x)$ is defined to be the unique value $k\in[K]$ for which $J(x,(k,\ell))=0$ and $J(x,(\ell,k))=1$ for all $\ell\in[K]\backslash\{k\}$, if such k exists. Otherwise $H(x)$ is defined to be 1.

Note that $\mathsf{Dec}^{\mathsf{mc}}$ depends on $K,\mathcal{X}$; for simplicity, we omit this dependence in our notation. Lemma G.1 bounds the VC dimension of the menu class of $\mathcal{H}$ in terms of the Natarajan dimension of $\mathcal{H}$.

Lemma G.1. *For any $\mathcal{H}\subset[K]^{\mathcal{X}}$, it holds that $d_{\mathsf{VC}}(\mathscr{M}_2(\mathcal{H}))\le O(d_{\mathsf{N}}(\mathcal{H})\log K)$.*

Proof. Let us write $d:=d_{\mathsf{VC}}(\mathscr{M}_2(\mathcal{H}))$, and let $(x_1,\mu_1),\dots,(x_d,\mu_d)$ be shattered by $\mathscr{M}_2(\mathcal{H})$. Note that the values $x_1,\dots,x_d$ are all distinct, since no hypothesis h^{menu} can shatter the points $(x,\mu),(x,\mu')$ for distinct menus $\mu,\mu'\in\mathcal{P}_2(K)$. Thus, the number of vectors $y\in[K]^n$ for which there exists $h\in\mathcal{H}$ so that $y_i=h(x_i)$, $i\in[n]$, is at least 2^d. But the number of such y may also be upper bounded by $(K^2ed/d_{\mathsf{N}}(\mathcal{H}))^{d_{\mathsf{N}}(\mathcal{H})}$, by (Haussler and Long, 1995). It follows that $2^d\le(K^2ed/d_{\mathsf{N}}(\mathcal{H}))^{d_{\mathsf{N}}(\mathcal{H})}$, i.e., $d\le d_{\mathsf{N}}(\mathcal{H})\cdot O(\log(Kd/d_{\mathsf{N}}(\mathcal{H})))$, which yields $d\le O(d_{\mathsf{N}}(\mathcal{H})\cdot\log(K))$. □

Algorithm 7 Multiclass learner

Require: Concept class $\mathcal{H}\subset[K]^{\mathcal{X}}$, sample $\{(x_i,y_i)\}_{i\in[n]}\subset(\mathcal{X}\times[K])^n$, domain point $x\in\mathcal{X}$, weak learner $\mathscr{A}$ for $\mathscr{M}_2(\mathcal{H})$ with margin η, failure probability δ, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$.

function MULTICLASSREALIZABLE$(S,x,\mathscr{A},\eta,\delta)$

Define a dataset $\tilde{S}\subset(\mathcal{X}\times\mathcal{P}_2(K)\times\{0,1,*\})^{2n(K-1)}$ as follows:

$$\tilde{S}=\{(x_i,(y_i,\ell),0)\ :\ i\in[n],\ell\in[K]\backslash\{y_i\}\}\cup\{(x_i,(\ell,y_i),1)\ :\ i\in[n],\ell\in[K]\backslash\{y_i\}\}. \tag{20}$$

Call `Adaboost` (Algorithm 5) on the dataset $\tilde{S}$ using the weak learner $\mathscr{A}$ with $T=\lceil 16\log(4nK/\delta)/\eta^2\rceil$, and denote the resulting hypothesis by $J:\mathcal{X}\times\mathcal{P}_2(K)\to\{0,1\}$.

Define the hypothesis $H:=\mathsf{Dec}^{\mathsf{mc}}(J)\in[K]^{\mathcal{X}}$ (see Definition G.2).

return $H(x)$.

function MULTICLASSAGNOSTIC$(S,\mathscr{A},\eta,\delta)$

Let $z\in\{0,1\}^n$ denote the output of `SampleERM.Binary`$(S,\ell^{\mathsf{mc}},\mathcal{O}^{\mathsf{erm,w}})$. ▷ *(Algorithm 2)*

Define $\tilde{S}:=\{(x_i,y_i)\ :\ i\in[n],z_i=0\}$.

Set $H:=$`MulticlassRealizable`$(\tilde{S},\mathscr{A},\eta,\delta)$.

return $H(x)$.

Note that, given a weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ for the class $\mathcal{H}\subset[K]^{\mathcal{X}}$, we immediately obtain a weak consistency oracle $\mathcal{O}^{\mathsf{con,menu}}$ for the class $\mathscr{M}_2(\mathcal{H})$: for a $\mathscr{M}_2(\mathcal{H})$-realizable dataset $S=\{(x_i,\mu_i,y_i)\}_{i\in[n]}$, the oracle $\mathcal{O}^{\mathsf{con,menu}}(S)$ defines $y_i'=(\mu_i)_{y_i+1}$, and returns $\mathcal{O}^{\mathsf{con,w}}(\{(x_i,y_i')\}_{i\in[n]})$. Using this observation, we may obtain a weak learner for the class $\mathscr{M}_2(\mathcal{H})$ as a corollary of Theorem 3.2:

Corollary G.2. *There are constants C_1, C_2 so that the following holds. Consider a multiclass concept class $\mathcal{H} \subset [K]^{\mathcal{X}}$ of Natarajan dimension d_{N}, and suppose $m \geq C_1 d_{\mathsf{N}} \log K \cdot \log(d_{\mathsf{N}} \log K)$. Let $S = \{((x_i, \mu_i), y_i)\} \in (\mathcal{X} \times \mathcal{P}_2(K) \times \{0,1\})^m$ be $\mathscr{M}_2(\mathcal{H})$-realizable. and let $\mathscr{A}(S_{-i}, (x_i, \mu_i))$ be the output of* `WeakRealizable`$(S_{-i}, (x_i, \mu_i), 1 - \frac{1}{C_1 m \log m}, 1, C_1 m^2 \log^3 m, \mathcal{O}^{\mathsf{con,menu}})$ *(Algorithm 1) for each $i \in [m]$. Then*

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\mathscr{A}} \left[\ell^{\mathsf{bin}}(\mathscr{A}(S_{-i}, (x_i, \mu_i)), y_i) \right] \leq \frac{1}{2} - \frac{1}{C_2 m \log m},$$

where the expectation is taken over the randomness in the runs of `WeakRealizable` *as well as the sampling of $\hat{y}$ from its output. Moreover,* `WeakRealizable` *makes at most $\tilde{O}(m^3)$ calls to $\mathcal{O}^{\mathsf{con,menu}}$, each with a dataset of size $m-1$.*

Proof. The corollary follows immediately from Theorem 3.2 applied to the class $\mathscr{M}_2(\mathcal{H})$, which has VC dimension bounded by $O(d_{\mathsf{N}} \log K)$ by Lemma G.1, together with the fact observed above that a weak consistency oracle for $\mathscr{M}_2(\mathcal{H})$ can be implemented using a weak consistency oracle for $\mathcal{H}$. □

Lemma G.3 (Training error of `Adaboost` in multiclass setting)**.** *Let $m, n \in \mathbb{N}$ and $\eta, \delta \in (0,1)$ be given, and suppose that algorithm $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathscr{M}_2(\mathcal{H})$. Let $x \in \mathcal{X}$ and $\bar{S} \in (\mathcal{X} \times [K])^n$ be an $\mathcal{H}$-realizable sample. Then the hypothesis H defined in Line 4 of* `MulticlassRealizable`$(\bar{S}, x, \mathscr{A}, \eta, \delta)$ *(Algorithm 7) satisfies $\widehat{\mathrm{er}}_{\bar{S}}(H) = 0$ with probability $1 - \delta$.*

Note that the domain point x plays no role in Lemma G.3.

Proof of Lemma G.3. By Lemma E.1 applied to the dataset $\tilde{S}$ (defined in (20)) together with the guarantee on the weak learner $\mathscr{A}$, the output hypothesis $J : \mathcal{X} \times \mathcal{P}_2(K) \to \{0,1\}$ of `Adaboost` satisfies the following with probability $1 - \delta$:

$$\forall i \in [n],\, \ell \in [K] \backslash \{y_i\}, \quad J(x, (y_i, \ell)) = 0,\, J(x, (\ell, y_i)) = 1. \tag{21}$$

For each $(x, y) \in \mathcal{X} \times [K]$, we have

$$\ell^{\mathsf{mc}}(H(x), y) = \mathbb{1}\{H(x) \neq y\} \leq \sum_{\ell \in [K] \backslash \{y\}} \mathbb{1}\{J(x, (y, \ell)) = 1\} + \mathbb{1}\{J(x, (\ell, y)) = 0\},$$

since if $H(x) \neq y$, then there must be some $\ell \neq y$ so that $J(x, (y, \ell)) = 1$ or $J(x, (\ell, y)) = 0$. The guarantee (21) on J yields that we must have $H(x_i) = y_i$ for all $i \in [n]$, with probability $1 - \delta$.

□

Proof of Theorem 4.1. Let $d_{\mathsf{N}} \in \mathbb{N}$ be given and consider a concept class $\mathcal{H} \subset [K]^{\mathcal{X}}$ with $d_{\mathsf{N}}(\mathcal{H}) \leq d_{\mathsf{N}}$, together with a weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ and a weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$. Let C_1, C_2 denote the constants in the statement of Corollary G.2, and define $m := C_1 d_{\mathsf{N}} \log K \cdot \log(d_{\mathsf{N}} \log K)$ and $\eta := \frac{1}{C_2 m \log m}$. Write $T = \lceil 16 \log(4nK/\delta)/\eta^2 \rceil$ as in Algorithm 7. Let $\mathscr{A} : (\mathcal{X} \times \mathcal{P}_2(K) \times \{0,1\})^m \times (\mathcal{X} \times \mathcal{P}_2(K)) \to \{0,1\}$ denote the learner of Corollary G.2 (in particular, $\mathscr{A}(S, (x, \mu))$ calls `WeakRealizable` with inputs $S, (x, \mu), \mathcal{O}^{\mathsf{con,w}}$, and an appropriate choice of the parameters). Then by Corollary G.2 and Lemma D.4, $\mathscr{A}$ is an m-sample weak learner for the partial concept class $\mathscr{M}_2(\mathcal{H})$ with margin η.

Realizable case. We begin by proving the first statement of the theorem. Given $\epsilon,\delta \in (0,1)$ and an $\mathcal{H}$-realizable distribution $P \in \Delta(\mathcal{X} \times [K])$, consider a sample $S \sim P^n$, where n will be chosen below. We will show that the algorithm which, given S and some $x \in \mathcal{X}$ as input, returns the output $H(x)$ of $\texttt{MulticlassRealizable}(S,x,\mathscr{A},\eta,\delta)$ satisfies the requirements of the theorem. Let $H:\mathcal{X} \to [K]$ denote the hypothesis constructed in Line 4 of `MulticlassRealizable`; we wish to show that with probability $1-\delta$ over S and the random bits of `MulticlassRealizable`, $\mathrm{er}_P(H) \leq \epsilon$. To do so, we will define a distribution Q over sample compression schemes (ρ,κ) on n-sample datasets, so that the distribution of $\rho(\kappa(S))$ is the same as the distribution of H. The call to `Adaboost` (Algorithm 5) in Line 3 of Algorithm 7 generates a sequence of i.i.d. datasets $\tilde{S}_1,\ldots,\tilde{S}_T \in (\mathcal{X} \times \mathcal{P}_2(K) \times \{0,1\})^m$, each consisting of examples in $\tilde{S}$ (defined in (20)), together with a sequence of parameters $\alpha_1,\ldots,\alpha_T \in \mathbb{R}$ and a sequence of random bitstrings $R_1,\ldots,R_T$ for use in the weak learner $\mathscr{A}$ and in the sampling step on Line 4 of Algorithm 5. The values of $\tilde{S}_t,\alpha_t,R_t$ satisfy the following: the output hypothesis $J:\mathcal{X} \times \mathcal{P}_2(K) \to \{0,1\}$ of `Adaboost` is given by

$$J(x,\mu) := \frac{1}{2} + \frac{1}{2}\mathrm{sign}\left(\sum_{t=1}^T \alpha_t \cdot (2\mathscr{A}_{R_t}(\tilde{S}_t,x,\mu)-1)\right). \tag{22}$$

For $j \in [m]$, the jth example in $\tilde{S}_t$ may be written as $(x_{i_{t,j}},(\ell_{t,j},k_{t,j}),b_{t,j})$, for some $i_{t,j} \in [n], \ell_{t,j} \in [K], k_{t,j} \in [K], b_{t,j} \in \{0,1\}$. We then define the (random) dataset $S_t := \{(x_{i_{t,j}}, y_{i_{t,j}})\}_{j \in [m]}$.

Note that, for fixed $S \in (\mathcal{X} \times [K])^n$, the resulting random variables $(\tilde{S}_t, S_t, \alpha_t)_{t \in [T]}$ are a deterministic function of S and $(R_t)_{t \in [T]}$. We now define $(\rho,\kappa) \sim Q$ to be distributed as follows:

- κ maps the dataset $S \in (\mathcal{X} \times [K])^n$ to $\kappa(S) = (S_1,\ldots,S_T), \left(\alpha_t, (\ell_{t,j},k_{t,j},b_{t,j})_{j \in [m]}\right)_{t \in [T]}$, where $S_t,\alpha_t,\ell_{t,j},k_{t,j},b_{t,j}$ are defined as a function of $(R_t)_{t \in [T]}$ and S as described above.
- ρ proceeds as follows, given an input of the form $(S_1',\ldots,S_T'), \left(\alpha_t', (\ell_{t,j}',k_{t,j}',b_{t,j}')_{j \in [m]}\right)_{t \in [T]}$ (where $S_1',\ldots,S_T'$ is a sequence of examples in $\mathcal{X} \times [K]$ of length Tm and the supplemental information $\alpha_t' \in \mathbb{R}$, $\ell_{t,j}',k_{t,j}' \in [K], b_{t,j}' \in \{0,1\}$ are encoded in binary). Denoting $S_t' = \{(x_{t,j}', y_{t,j}')\}_{j \in [m]}$, let us define $\tilde{S}_t' := \{(x_{t_j}', (\ell_{t,j}',k_{t,j}'), b_{t,j}')\}_{j \in [m]}$; then ρ outputs the hypothesis $x \mapsto \mathsf{Dec}^{\mathsf{mc}}(J')$, where $J'(x,\mu) := \left(\frac{1}{2} + \frac{1}{2}\mathrm{sign}\left(\sum_{t=1}^T \alpha_t' \cdot (2\mathscr{A}_{R_t}(\tilde{S}_t',x,\mu)-1)\right)\right)$.

Since the values α_t defined in `Adaboost` can be encoded with $O(\log n)$ bits (as $\epsilon_t \in \{0,1/n,\ldots,1\}$) and the list $((\ell_{t,j},k_{t,j},b_{t,j}))_{j \in [m]}$ can be encoded with $O(m\log K)$ bits, with probability 1 over $(\kappa,\rho) \sim Q$, the size of κ for inputs samples S of size n is bounded by $|\kappa| \leq O(T \cdot (m\log K + \log n))$. Next, Lemma G.3 together with the definition of Q above (and in particular the fact that ρ uses the same bits R_t as in the definition of κ), gives that for any $\mathcal{H}$-realizable dataset S, there is a subset $\mathcal{E}$ of compression schemes for which $Q(\mathcal{E}) \geq 1-\delta$ so that for all $(\kappa,\rho) \in \mathcal{E}$, $\widehat{\mathrm{er}}_S(\rho(\kappa(S))) = 0$. By Lemma A.1 applied to the multiclass loss function, there is a constant $C>0$ so that for each $(\kappa,\rho) \in \mathrm{supp}(Q)$, with probability $1-\delta$ over the draw of $S \sim P^n$,

$$\mathrm{er}_P(\rho(\kappa(S))) \leq \mathbb{1}\{\widehat{\mathrm{er}}_S(\rho(\kappa(S)))=0\} + C \cdot \frac{T \cdot (m\log K + \log n) \cdot \log n + \log(1/\delta)}{n},$$

for some sufficiently large constant C. By our choice of T together with the fact that $Q(\mathcal{E}) \geq 1-\delta$, it follows that with probability $1-2\delta$ over the draw of $S \sim P^n$ and the draw of $(\rho,\kappa) \sim Q$, we have

$$\mathrm{er}_P(\rho(\kappa(S))) \leq C \cdot \frac{\log(n)\log(nK/\delta) \cdot \frac{1}{\eta^2} \cdot (m\log K + \log n)}{n}.$$

For fixed S, the distribution of $\rho(\kappa(S))$ is the same as the distribution of the output hypothesis H of `MulticlassRealizable`. Thus, by our choices of m,η, after rescaling δ, we can ensure that $\mathrm{er}_P(H) \le \epsilon$ with probability $1-\delta$ as long as we take $n = \frac{d_{\mathsf{N}}^3 \log^4(K/\delta)}{\epsilon} \cdot \mathrm{poly}(\log 1/\epsilon, \log d_{\mathsf{N}}, \log\log K)$.

Note that the dataset $\tilde{S}$ passed to `Adaboost` in `MulticlassRealizable` is of size $|\tilde{S}| = O(nK)$. Since the number of rounds of `Adaboost` is $T \le \mathrm{poly}(n)$ and the weak learner $\mathscr{A}$ makes $\mathrm{poly}(m)$ calls to $\mathcal{O}^{\mathsf{con,w}}$, which can simulate $\mathcal{O}^{\mathsf{con,menu}}$ (Corollary G.2), the same argument as in the proof of Theorem 3.1 shows that the cumulative query cost of `MulticlassRealizable` for the oracle $\mathcal{O}^{\mathsf{con,w}}$ is $\tilde{O}(Knm^3 \cdot T) \le K \cdot \mathrm{poly}(n)$.

Agnostic case. Let $P \in \Delta(\mathcal{X} \times [K])$ be a distribution. For n sufficiently large (as specified below), we will show that the algorithm which takes as input a sample $S \sim P^n$ and $x \in \mathcal{X}$, and which returns the output $H(x)$ of `MulticlassAgnostic`$(S, x, \mathscr{A}, \eta, \delta)$ (Algorithm 7) satisfies the requirements of the theorem. Let $H : \mathcal{X} \to [K]$ denote the hypothesis returned by `MulticlassRealizable` in `MulticlassAgnostic`; we want to show that with probability $1-\delta$ over S and the random bits of `MulticlassAgnostic`, $\mathrm{er}_P(H) \le \min_{h \in \mathcal{H}} \mathrm{er}_P(h) + \epsilon$. To do so, we consider the same distribution Q over sample compression schemes on datasets with (at most) n samples. Let Σ denote the mapping which takes as input $S \in (\mathcal{X} \times [K])^n$ and outputs the dataset $\tilde{S} \in (\mathcal{X} \times [K])^{\le n}$ as defined in `MulticlassAgnostic`. For any compression scheme $(\kappa, \rho) \in \mathrm{supp}(Q)$, $(\kappa \circ \Sigma, \rho)$ is a compression scheme of size $|\kappa \circ \Sigma| \le |\kappa| \le O(T \cdot (m \log K + \log n))$. Thus, by Lemma A.1, there is a constant $C > 0$ so that, for any fixed $(\kappa, \rho) \in \mathrm{supp}(Q)$, with probability $1-\delta$ over the draw of $S \sim P^n$,

$$\mathrm{er}_P(\rho(\kappa(\Sigma(S)))) \le \widehat{\mathrm{er}}_S(\rho(\kappa(\Sigma(S)))) + C\sqrt{\frac{1}{n}(T\log(n) \cdot (m\log K + \log n) + \log(1/\delta))}. \tag{23}$$

By Lemma C.1 with loss function ℓ^{mc}, for any $S \in (\mathcal{X} \times [K])^n$, the dataset $\tilde{S} = \Sigma(S)$ satisfies $\inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h) = \frac{n - |\tilde{S}|}{n}$. The lemma also guarantees that $\tilde{S}$ is $\mathcal{H}$-realizable, and thus there is a set $\mathcal{E}$ of compression schemes satisfying $Q(\mathcal{E}) \ge 1-\delta$ so that for all $(\kappa, \rho) \in \mathcal{E}$, $\widehat{\mathrm{er}}_{\tilde{S}}(\rho(\kappa(\tilde{S}))) = 0$. Using this fact and (23), it follows that with probability $1-2\delta$ over the draw of $S \sim P^n$,and $(\rho, \kappa) \sim Q$,

$$\begin{aligned} \mathrm{er}_P(\rho(\kappa(\Sigma(S)))) &\le \frac{n - |\tilde{S}|}{n} + C\sqrt{\frac{1}{n}(T\log(n) \cdot (m\log K + \log n) + \log(1/\delta))} \\ &= \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{\tilde{S}}(h) + C\sqrt{\frac{\log(n)\log(nK/\delta) \cdot \frac{1}{\eta^2} \cdot (m\log K + \log n)}{n}}, \end{aligned} \tag{24}$$

where the equality uses our choice of T. Moreover McDiarmid's inequality yields that with probability $1-\delta$ over the choice of $S \sim P^n$,

$$\inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h) \le \mathbb{E}_{S' \sim P^n}\left[\inf_{h' \in \mathcal{H}} \widehat{\mathrm{er}}_{S'}(h')\right] + C\sqrt{\frac{\log 1/\delta}{n}} \le \inf_{h \in \mathcal{H}} \mathrm{er}_P(h) + C\sqrt{\frac{\log 1/\delta}{n}}, \tag{25}$$

for a sufficiently large constant C. Combining Eqs. (24) and (25), using our choice of m, η, and choosing $n = \frac{d_{\mathsf{N}}^3 \log^4(K/\delta)}{\epsilon^2} \cdot \mathrm{poly}(\log 1/\epsilon, \log d_{\mathsf{N}}, \log\log K)$ yields the claimed statement of the theorem.

To analyze the oracle complexity of `MulticlassAgnostic`, we first note that the call to `SampleERM.Binary` makes $O(n^2)$ calls to $\mathcal{O}^{\mathsf{erm,w}}$ (Lemma C.1). The remainder of the analysis of oracle complexity follows the realizable case exactly. $\square$

Appendix H. Regression

In this section, we consider another generalization of our results on binary classification, namely to the setting of regression, in which hypotheses and labels take values in $[0,1]$. Similar to Appendix G, our approach proceeds via reducing to oracle-efficient learning of partial function classes. A similar approach was used in (Bartlett and Long, 1998; Aden-Ali et al., 2023). We thus define a partial concept class associated to any real-valued concept class in Definition H.1 below:

Definition H.1 (Threshold class)**.** Consider a hypothesis class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ and $\gamma \in (0,1)$. We let $\mathcal{D}_\gamma := \{0,\gamma,2\gamma,...,\lfloor 1/\gamma \rfloor \gamma\}$, Given $h \in \mathcal{H}$ and $\tau \in \mathcal{D}_\gamma$, we define

$$h_\gamma^{\mathsf{thr}}(x,\tau) := \begin{cases} 1 & : h(x) \geq \tau + \gamma \\ 0 & : h(x) \leq \tau - \gamma \\ * & : |h(x) - \tau| < \gamma. \end{cases}$$

We then let $\mathscr{T}_\alpha(\mathcal{H}) \subset \{0,1,*\}^{\mathcal{X} \times \mathcal{D}_\gamma}$ denote the (binary) partial hypothesis class defined by

$$\mathscr{T}_\alpha(\mathcal{H}) := \{(x,\tau) \mapsto h_\gamma^{\mathsf{thr}}(x,\tau) \ : \ h \in \mathcal{H}\}.$$

We will refer to $\mathscr{T}_\alpha(\mathcal{H})$ as the *threshold class* of $\mathcal{H}$.

Lemma H.1. *For any $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ and $\gamma \in (0,1)$, it holds that $d_{\mathsf{VC}}(\mathscr{T}_\alpha(\mathcal{H})) \leq d_{\mathsf{fat},\gamma}(\mathcal{H})$.*

Proof. Let us write $d := d_{\mathsf{fat},\gamma}(\mathcal{H})$, and let $(x_1,\tau_1),...,(x_d,\tau_d)$ be shattered by $\mathscr{T}_\alpha(\mathcal{H})$, so that $x_i \in \mathcal{X}$ and $\tau_i \in \mathcal{D}_\gamma$ for each $i \in [d]$. Then by the definition of $\mathscr{T}_\alpha(\mathcal{H})$ (Definition H.1), for each sequence $b \in \{0,1\}^d$, there is some $h \in \mathcal{H}$ so that, for each $i \in [d]$, $h(x) \geq \tau_i + \gamma$ if $b_i = 1$ and $h(x) \leq \tau_i - \gamma$ if $b_i = 0$. Thus $\mathcal{H}$ shatters the points $(x_1,...,x_d)$ as witnessed by $(\tau_1,...,\tau_d)$, i.e., $d_{\mathsf{fat},\gamma}(\mathcal{H}) \geq d$. □

An immediate consequence of Lemma C.4 is that given a weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for the class $\mathcal{H}$, we immediately obtain a weak consistency oracle for the class $\mathscr{T}_\alpha(\mathcal{H})$. Using this observation, we obtain the following corollary of Theorem 3.2:

Corollary H.2. *There are constants C_1, C_2 so that the following holds. Consider a real-valued concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ and $\gamma \in (0,1)$, write $d := d_{\mathsf{fat},\gamma}(\mathcal{H})$, and suppose $m \geq C_1 d \log d$. Let $S = \{((x_i,\tau_i),y_i)\} \in (\mathcal{X} \times \mathcal{D}_\gamma \times \{0,1\})^m$ be $\mathscr{T}_\alpha(\mathcal{H})$-realizable. Then there is an algorithm $\mathscr{A} : (\mathcal{X} \times \mathcal{D}_\gamma \times \{0,1\})^{m-1} \times (\mathcal{X} \times \mathcal{D}_\gamma) \to \Delta(\{0,1\})$ which makes $\tilde{O}(m^3)$ calls to a range consistency oracle $\mathcal{O}^{\mathsf{range}}$ for $\mathcal{H}$ and which satisfies*

$$\frac{1}{m} \sum_{i=1}^m \mathbb{E}_{\hat{y} \sim \mathscr{A}(S_{-i},(x_i,\tau_i))} \left[\ell^{\mathsf{bin}}(\hat{y},y_i) \right] \leq \frac{1}{2} - \frac{1}{C_2 m \log m}.$$

Proof. The corollary follows immediately from Theorem 3.2 applied to the class $\mathscr{T}_\alpha(\mathcal{H})$, which has VC dimension bounded by $d_{\mathsf{fat},\gamma}(\mathcal{H})$ by Lemma H.1, together with the fact that a single weak ERM oracle call for the class $\mathscr{T}_\alpha(\mathcal{H})$ can be implemented by a single range consistency oracle call for the class $\mathcal{H}$ of the same length, using `SampleCon.Real` (Lemma C.4). □

Lemma H.3 (Training error for regression)**.** *Let $m,n \in \mathbb{N}$, $x \in \mathcal{X}$, and $\eta,\delta,\gamma,\beta \in (0,1)$ be given, and suppose that algorithm $\mathscr{A}$ is an m-sample weak learner with margin η for the class $\mathscr{T}_\alpha(\mathcal{H})$. Let $\bar{S} \in (\mathcal{X} \times [0,1])^n$ be a sample so that for some $h^\star \in \mathcal{H}$, each $(x_i,y_i) \in \bar{S}$ satisfies $|y_i - h^\star(x_i)| \leq \beta - \gamma$. Then the hypothesis $H(\cdot) := \gamma \cdot \sum_{\tau \in \mathcal{D}_\gamma} J(\cdot,\tau)$, where J is defined on Line 4 of `RegRealizable`$(\bar{S},x,\mathscr{A},\eta,\delta,\gamma,\beta)$, satisfies $\widehat{\mathrm{er}}_{\bar{S}}(H) \leq 3\beta$ with probability $1-\delta$.*

Algorithm 8 Real-valued learner

Require: Concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$, sample $S = \{(x_i,y_i)\}_{i\in[n]} \subset (\mathcal{X} \times [0,1])^n$, point $x \in \mathcal{X}$, failure probability δ, discretization parameters $\gamma,\beta \in (0,1)$, weak learner $\mathscr{A}$ for $\mathscr{T}_\alpha(\mathcal{H})$ with margin η, weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$.

function $\textsc{RegRealizable}(S,x,\mathscr{A},\eta,\delta,\gamma,\beta)$

  For $i \in [n]$ and $\tau \in \mathcal{D}_\gamma$, define $y'_{i,\tau} := \begin{cases} 1 & : y_i \geq \tau+\beta \\ 0 & : y_i \leq \tau-\beta \\ * & : |y_i-\tau| < \beta. \end{cases}$

  Define a dataset $\tilde{S} \subset (\mathcal{X} \times \mathcal{D}_\gamma \times \{0,1,*\})^{n'}$ for some $n' \leq 2n\cdot(\lfloor 1/\gamma \rfloor + 1)$, as follows:

$$\tilde{S} = \left\{ ((x_i,\tau),y'_{i,\tau}) \ : \ i \in [n], \tau \in \mathcal{D}_\gamma, y'_{i,\tau} \neq * \right\}. \tag{26}$$

  Call `Adaboost` (Algorithm 5) on the dataset $\tilde{S}$ using the weak learner $\mathscr{A}$ with $T = \lceil 16\log(4n/\delta)/\eta^2 \rceil$, and denote the resulting hypothesis by $J : \mathcal{X} \times \mathcal{D}_\gamma \to \{0,1\}$.

  return $H(x) := \gamma \cdot \sum_{\tau \in \mathcal{D}_\gamma} J(x,\tau)$.

function $\textsc{RegAgnostic}(S,x,\mathscr{A},\eta,\delta,\gamma)$

  Let $\hat{y} \in [0,1]^n$ denote the output of `SampleERM.Real`$(S,\gamma/2,\mathcal{O}^{\mathsf{erm,w}})$. ▷ *Algorithm 3*

  Define $\tilde{S} = \{(x_i,\hat{y}_i) \ : \ i \in [n]\}$.

  return $H(x) :=$ `RegRealizable`$(\tilde{S},x,\mathscr{A},\eta,\delta,\gamma,2\gamma)$.

Proof. Since there is some $h^\star$ so that each $(x,y) \in \bar{S}$ satisfies $|y - h^\star(x)| \leq \beta - \gamma$, the dataset $\tilde{S}$ defined in (26) is $\mathscr{T}_\alpha(\mathcal{H})$-realizable. Then by Lemma E.1 together with the assumed guarantee on $\mathscr{A}$, the output hypothesis $J : \mathcal{X} \times \mathcal{D}_\gamma \to \{0,1\}$ of `Adaboost` satisfies the following with probability $1-\delta$:

$$\forall i \in [n],\ \tau \in \{\tau' \in \mathcal{D}_\gamma \ : \ |y_i - \tau'| \geq \beta\}, \quad J(x_i,\tau) = \mathbb{1}\{\tau > y_i\}. \tag{27}$$

Thus, for each $i \in [n]$, letting $\hat{y}_i := \tau \cdot \lfloor y_i/\tau \rfloor$, we have

$$\begin{aligned} \ell^{\mathsf{abs}}(H(x_i),y_i) = |H(x_i) - y_i| \leq |H(x_i) - \hat{y}_i| + |\hat{y}_i - y_i| &\leq \gamma + \gamma \cdot \sum_{\tau \in \mathcal{D}_\gamma} \mathbb{1}\{J(x_i,\tau) \neq \mathbb{1}\{\hat{y}_i > \tau\}\} \\ &\leq \gamma + \gamma\lceil \beta/\gamma \rceil + \gamma \cdot \sum_{\tau \in \mathcal{D}_\gamma,\ |y_i - \tau| \geq \beta} \mathbb{1}\{J(x_i,\tau) \neq \mathbb{1}\{\hat{y}_i > \tau\}\} \leq 3\beta, \end{aligned}$$

where the final equalitty uses (27). It follows that $\widehat{\mathrm{er}}_{\bar{S}}(H) = \frac{1}{n}\sum_{i=1}^n \ell^{\mathsf{abs}}(H(x_i),y_i) \leq 3\beta$, as desired. □

Proof of Theorem 4.2. The proof closely follows that of Theorem 4.1. Let a mapping $\gamma \mapsto d_{\mathsf{fat},\gamma}$ be given and consider a concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ with $d_{\mathsf{fat},\gamma}(\mathcal{H}) \leq d_{\mathsf{fat},\gamma}$ for all $\gamma \in [0,1]$, together a weak range oracle $\mathcal{O}^{\mathsf{range}}$ and a weak ERM oracle $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$. Let C_1, C_2 be the constants in the statement of Corollary H.2, and fix $\gamma \in (0,1)$. Define $m := C_1 d_{\mathsf{fat},\gamma} \cdot \log(d_{\mathsf{fat},\gamma})$ and $\eta := \frac{1}{C_2 m \log m}$. Write $T = \lceil 16\log(4n/\delta)/\eta^2 \rceil$ as in Algorithm 8. Let $\mathscr{A} : (\mathcal{X} \times \mathcal{D}_\gamma \times \{0,1\})^m \times (\mathcal{X} \times \mathcal{D}_\gamma) \to \{0,1\}$ denote the learner of Corollary H.2 (in particular, $\mathscr{A}(S,(x,\tau))$ calls `WeakRealizable` with inputs $S,(x,\tau)$, and an appropriate choice of the parameters). Then by Corollary H.2 and Lemma D.4, $\mathscr{A}$ is an m-sample weak learner for the partial concept class $\mathscr{T}_\alpha(\mathcal{H})$ with margin η. Moreover, as guaranteed by Corollary H.2, a single call of $\mathscr{A}$ can be completed using $\tilde{O}(m^3)$ calls to either $\mathcal{O}^{\mathsf{range}}$ or $\mathcal{O}^{\mathsf{erm,w}}$.

Realizable case. Given $n,\delta \in (0,1)$ and an $\mathcal{H}$-realizable distribution $P \in \Delta(\mathcal{X}\times[K])$, consider a sample $S \sim P^n$. We will show that the algorithm which, given S and some $x \in \mathcal{X}$ as input, returns the output $H(x)$ of `RegRealizable`$(S,x,\mathscr{A},\eta,\delta,\gamma,\gamma)$ satisfies the requirements of the theorem. Let $H:\mathcal{X}\to[0,1]$ be defined by $H(\cdot)=\gamma\cdot\sum_{\tau\in\mathcal{D}_\gamma} J(\cdot,\tau)$, where J is the hypothesis defined in Line 4 of `RegRealizable`. We want to show that with probability $1-\delta$ over S and the random bits of `RegRealizable`, $\mathrm{er}_P(H)\leq\epsilon$ for an appropriate choice of ϵ. To do so, we define a distribution Q over sample compression schemes (ρ,κ) on datasets of size at most $2n\cdot(\lfloor 1/\gamma\rfloor+1)$, so that the distribution of $\rho(\kappa(S))$ is the same as the distribution of H. The call to `Adaboost` (Algorithm 5) in Line 4 of Algorithm 8 generates a sequence of i.i.d. datasets $\tilde{S}_1,\dots,\tilde{S}_T \in (\mathcal{X}\times\mathcal{D}_\gamma\times\{0,1\})^m$, each consisting only of examples in $\tilde{S}$ (defined in (26)), together with a sequence of parameters $\alpha_1,\dots,\alpha_T\in\mathbb{R}$ and a sequence of random bitstrings $R_1,\dots,R_T$. The values of $\tilde{S}_t,\alpha_t,R_t$ satisfy the following: the output hypothesis $J:\mathcal{X}\times\mathcal{D}_\gamma\to\{0,1\}$ of `Adaboost` is given by $J(x,\tau):=\frac{1}{2}+\frac{1}{2}\mathrm{sign}\left(\sum_{t=1}^T\alpha_t\cdot(2\mathscr{A}_{R_t}(\tilde{S}_t,x,\tau)-1)\right)$. For $j\in[m]$, the jth example in $\tilde{S}_t$ may be written as $(x_{i_{t,j}},\tau_{t,j},b_{t,j})$, for some random variables $i_{t,j}\in[n],\tau_{t,j}\in\mathcal{D}_\gamma,b_{t,j}\in\{0,1\}$. We then define $S_t:=\{(x_{i_{t,j}},y_{i_{t,j}})_{j\in[m]}$.

We define $(\rho,\kappa)\sim Q$ to be distributed as follows:

- κ maps the dataset $S\in(\mathcal{X}\times[0,1])^n$ to $\kappa(S)=(S_1,\dots,S_T),(\alpha_t,(\tau_{t,j},b_{t,j})_{j\in[m]})_{t\in[T]}$, where $S_t,\alpha_t,\tau_{t,j},b_{t,j}$ are defined as a (deterministic) function of $(R_t)_{t\in[T]}$ and S as described above.
- ρ proceeds as follows, given an input of the form $(S_1',\dots,S_T'),\left(\alpha_t',(\tau_{t,j}',b_{t,j}')_{j\in[m]}\right)_{t\in[T]}$. Denoting $S_t'=\{(x_{t,j}',y_{t,j}')\}_{j\in[m]}$, let us define $\tilde{S}_t':=\{(x_{t_j}',\tau_{t,j}',b_{t,j}')\}_{j\in[m]}$; then ρ outputs the hypothesis $x\mapsto\gamma\cdot\sum_{\tau\in\mathcal{D}_\gamma}\left(\frac{1}{2}+\frac{1}{2}\mathrm{sign}\left(\sum_{t=1}^T\alpha_t'\cdot(2\mathscr{A}_{R_t}(\tilde{S}_t',x,\tau)-1)\right)\right)$.

Since the values α_t defined in `Adaboost` can be encoded with $O(\log n)$ bits and the list $(\tau_{t,j},b_{t,j})_{j\in[m]}$ can be encoded with $O(m\log 1/\gamma)$ bits, with probability 1 over $(\kappa,\rho)\sim Q$, the size of κ for input samples S of size n is bounded by $|\kappa|\leq O(T\cdot(m\log 1/\gamma+\log n))$. Lemma H.3 gives that there is a subset $\mathcal{E}$ of compression schemes for which $Q(\mathcal{E})\geq 1-\delta$ so that for all $(\kappa,\rho)\in\mathcal{E}$, $\widehat{\mathrm{er}}_S(\rho(\kappa(S)))\leq 3\beta$. By Lemma A.1 applied to the absolute loss function, it follows that with probability $1-2\delta$ over the draw of $S\sim P^n$ and the draw of $(\rho,\kappa)\sim Q$, we have that, for some constant C,

$$\begin{aligned}\mathrm{er}_P(\rho(\kappa(S)))\leq C\cdot\sqrt{\gamma\Delta}+\Delta\leq\gamma+2C^2\Delta \\ \text{for } \Delta:=\frac{\log(n)\log(n/\delta)\cdot\frac{1}{\eta^2}\cdot(m\log(1/\gamma)+\log n)}{n},\end{aligned} \tag{28}$$

where the second inequality in (28) uses the AM-GM inequality. Given S, the distribution of $\rho(\kappa(S))$ is the same as the distribution of the output hypothesis H of `RegRealizable`. Thus, by making an optimal choice of the discretization parameter γ, our choices of m,η and by rescaling δ, we see that $\mathrm{er}_P(H)\leq\epsilon$ with probability $1-\delta$ for

$$\epsilon:=\inf_{\gamma\in[0,1]}\left\{O(\gamma)+\frac{d_{\mathsf{fat},\gamma}^3\cdot\log(1/\delta)}{n}\cdot\mathrm{polylog}(d_{\mathsf{fat},\gamma},n,\log\log 1/\delta)\right\}. \tag{29}$$

Note that the dataset $\tilde{S}$ passed to `Adaboost` in `RegRealizable` is of size $|\tilde{S}| = O(n/\gamma) \leq O(n^2)$, since the optimal choice of γ in (29) is always bounded below by $1/n$. Since the number of rounds of `Adaboost` is $T \leq \mathrm{poly}(n)$ and the weak learner $\mathscr{A}$ makes $\mathrm{poly}(m)$ calls to $\mathcal{O}^{\mathsf{range}}$ (Corollary H.2), the same argument as in the proof of Theorem 3.1 establishes that the cumulative query cost of `RegRealizable` for the oracle $\mathcal{O}^{\mathsf{range}}$ is $\mathrm{poly}(n)$.

Agnostic case. Fix a distribution $P \in \Delta(\mathcal{X} \times [0,1])$. Given $n \in \mathbb{N}$, we will show that the algorithm which takes as input an i.i.d. sample $S \sim P^n$ and a point $x \in \mathcal{X}$, and returns the output $H(x)$ of $\texttt{RegAgnostic}(S, x, \mathscr{A}, \eta, \delta, \gamma, 2\gamma)$ (Algorithm 8) satisfies the requirements of the theorem. Let $H : \mathcal{X} \to [K]$ denote the (random) hypothesis defined by $x \mapsto \texttt{RegAgnostic}(S, x, \mathscr{A}, \eta, \delta, \gamma)$ (in particular, this hypothesis is exactly the one given by $x \mapsto \gamma \sum_{\tau \in \mathcal{D}_\gamma} J(x,\tau)$, where J is the hypothesis defined on Line 4 in the call to `RegRealizable` from `RegAgnostic`). We want to show that, with probability $1-\delta$ over S and the random bits of `RegAgnostic`, $\mathrm{er}_P(H) \leq \inf_{h \in \mathcal{H}} \mathrm{er}_P(h) + \epsilon$, for an appropriate value of ϵ. To do so, we consider the distribution Q' on compression schemes which is defined similarly to Q as above, with the exception that $(\rho,\kappa) \sim Q'$ are defined so as to simulate the execution of $\texttt{RegRealizable}(S,\mathscr{A},\eta,\delta,\gamma,2\gamma)$, as opposed to $\texttt{RegRealizable}(S,\mathscr{A},\eta,\delta,\gamma,\gamma)$. (In particular, this changes the definition of the i.i.d. datasets $\tilde{S}_t$ as defined in (26).)

Let Σ denote the mapping which, takes as input $S \in (\mathcal{X} \times [0,1])^n$ and outputs the dataset $\tilde{S} \in (\mathcal{X} \times [0,1])^n$ as defined in Line 8 of `RegAgnostic`. For any compression scheme $(\kappa,\rho) \in \mathrm{supp}(Q')$, $(\kappa \circ \Sigma, \rho)$ is a compression scheme of size $|\kappa \circ \Sigma| \leq |\kappa| \leq O(T \cdot (m \log(1/\gamma) + \log n))$. Thus, by Lemma A.1, there is a constant $C > 0$ so that, for any fixed $(\kappa,\rho) \in \mathrm{supp}(Q')$, with probability $1-\delta$ over the draw of $S \sim P^n$,

$$\mathrm{er}_P(\rho(\kappa(\Sigma(S)))) \leq \widehat{\mathrm{er}}_S(\rho(\kappa(\Sigma(S)))) + C\sqrt{\frac{1}{n}(T \log(n) \cdot (m \log(1/\gamma) + \log n) + \log(1/\delta))}. \tag{30}$$

By Lemma C.2 and the definition of Σ, for any $S = \{(x_i,y_i)\}_{i \in [n]}$, the dataset $\tilde{S} = \Sigma(S)$, which can be written as $\tilde{S} = \{(x_i,\hat{y}_i)\}_{i \in [n]}$, satisfies the following: there is some $h^\star \in \mathcal{H}$ so that $\widehat{\mathrm{er}}_S(h^\star) = \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h)$ and $\ell^{\mathsf{abs}}(h^\star(x_i),\hat{y}_i) \leq \gamma$ for all $i \in [n]$. Thus, Lemma H.3 with $\beta = 2\gamma$ yields that, for any S, the output hypothesis H of $\texttt{RegRealizable}(\tilde{S},\mathscr{A},\eta,\delta,\gamma,\beta)$ satisfies $\widehat{\mathrm{er}}_{\tilde{S}}(H) \leq 6\gamma$ with probability $1-\delta$. Since the hypothesis H has the same distribution as $\rho(\kappa(\Sigma(S)))$ for $(\rho,\kappa) \sim Q'$, we see that for any S, there is a set $\mathcal{E}'$ of compression schemes satisfying $Q'(\mathcal{E}') \geq 1-\delta$ so that, for all $(\kappa,\rho) \in \mathcal{E}'$, $\widehat{\mathrm{er}}_{\tilde{S}}(\rho(\kappa(\Sigma(S)))) \leq 6\gamma$. Moreover, note that, for any $H : \mathcal{X} \to [0,1]$,

$$\begin{aligned}\widehat{\mathrm{er}}_S(H) = \frac{1}{n}\sum_{i=1}^n \ell^{\mathsf{abs}}(y_i, H(x_i)) &\leq \frac{1}{n}\sum_{i=1}^N \ell^{\mathsf{abs}}(H(x_i),\hat{y}_i) + \frac{1}{n}\sum_{i=1}^n \ell^{\mathsf{abs}}(\hat{y}_i, h^\star(x_i)) + \frac{1}{n}\sum_{i=1}^n \ell^{\mathsf{abs}}(h^\star(x_i), y_i)\\ &\leq \widehat{\mathrm{er}}_{\tilde{S}}(H) + \gamma + \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h).\end{aligned} \tag{31}$$

Combining (31) with $H = \rho(\kappa(\Sigma(S)))$ and (30) gives that with probability $1-\delta$ over the draw of $S \sim P^n$ and $(\rho,\kappa) \sim Q'$,

$$\begin{aligned}\mathrm{er}_P(\rho(\kappa(\Sigma(S)))) &\leq \widehat{\mathrm{er}}_{\tilde{S}}(\rho(\kappa(\Sigma(S)))) + \gamma + \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h) + C\sqrt{\frac{1}{n}(T \log(n) \cdot (m \log(1/\gamma) + \log n) + \log(1/\delta))}\\ &\leq 7\gamma + \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h) + C\sqrt{\frac{1}{n}\left(\frac{1}{\eta^2} \cdot \log(n/\delta)\log(n) \cdot (m \log(1/\gamma) + \log n) + \log(1/\delta)\right)}.\end{aligned}$$

McDiarmid's inequality yields that with probability $1-\delta$ over the choice of $S \sim P^n$, $\inf_{h\in\mathcal{H}} \widehat{\mathrm{er}}_S(h) \leq \inf_{h\in\mathcal{H}} \mathrm{er}_P(h) + C\sqrt{\log(1/\delta)/n}$, and combining this fact with the above display and the choice of Q' gives that, with probability $1-\delta$ over the choice of $S \sim P^n$ and the execution of `RegAgnostic`, its output hypothesis H satisfies

$$\mathrm{er}_P(H) \leq \inf_{h\in\mathcal{H}} \mathrm{er}_P(h) + 7\gamma + C\sqrt{\frac{d_{\mathsf{fat},\gamma}^3 \cdot \log(1/\delta)}{n}} \cdot \mathrm{polylog}(d_{\mathsf{fat},\gamma}, n, \log\log 1/\delta).$$

Infimizing over $\gamma \in (0,1)$ gives the claimed result.

To analyze the oracle complexity of `RegAgnostic`, we first note that the call to `SampleERM.Real` makes $O(n/\gamma) \leq O(n^2)$ calls to $\mathcal{O}^{\mathsf{erm,w}}$ (Lemma C.2). The remainder of the analysis of oracle complexity follows the realizable case exactly. $\square$

Appendix I. Lower bounds

In this section, we present lower bounds for oracle-efficient PAC learning with a (strong) ERM oracle. Appendix I.1 treats the setting of multiclass classification with bounded DS dimension, and Appendix I.2 treats the setting of realizable regression with bounded one-inclusion graph dimension. Notice that in both of these settings, uniform convergence does not hold (otherwise, a single ERM call on the i.i.d. sample would suffice).

I.1. Lower bound for multiclass classification

A recent breakthrough result (Brukhim et al., 2022) established that the *DS dimension* of a multiclass concept class $\mathcal{H} \subset [K]^{\mathcal{X}}$ characterizes the sample complexity of PAC learning up to a polynomial factor, in both the realizable and agnostic settings. Our main result of this section, Theorem I.2, shows that even when the DS dimension is 1 a strong ERM oracle is insufficient for PAC learnability.

We begin by introducing the DS dimension. For simplicity, we restrict to the setting that the number of classes K is finite (as such will be the case in our lower bound). Given $n \in \mathbb{N}$ and a nonempty subset $\mathcal{S} \subset [K]^n$, $\mathcal{S}$ is called a *n-dimensional pseudocube* if for each $y \in \mathcal{S}$, there is some $y' \in \mathcal{S}$ so that $y'_i \neq y_i$ and $y_j = y'_j$ for all $j \neq i$. The *DS dimension* of $\mathcal{H} \subset [K]^{\mathcal{X}}$, denoted $d_{\mathsf{DS}}(\mathcal{H})$ (Daniely and Shalev-Shwartz, 2014), is defined to be the largest positive integer d so that, for some $X = (x_1, \ldots, x_d) \in \mathcal{X}^d$, $\mathcal{H}|_X$ contains a d-dimensional pseudocube.

Next, we introducing the concept class which will be used to prove Theorem I.2. Given positive integers $N, K \in \mathbb{N}$, we set $\mathcal{X}_N := [N]$ and let the label space be $\mathcal{Y}_K := [K]$. For simplicity, we will often write $\mathcal{X} = \mathcal{X}_N = [N]$ and $\mathcal{Y} = \mathcal{Y}_K$ when the values of N, K are clear.

Let $\mathcal{X}^{\leq q} = [N]^{\leq q}$ denote the set of subsequences of $[N]$ of length at most q (including the empty sequence). Note that $|\mathcal{X}^{\leq q}| \leq 2qN^q$.

For mappings $h^\star : \mathcal{X} \to [K]$ and $\phi^\star : \mathcal{X}^{\leq q} \to [K]$ and an element $z \in \mathcal{X}^{\leq q}$, we define the *merged hypothesis* $\mathsf{merge}(h^\star, \phi^\star, z) \in \mathcal{Y}^{\mathcal{X}}$, as follows:

$$\mathsf{merge}(h^\star, \phi^\star, z)(x) := \begin{cases} \phi^\star(z) & : x \notin z \\ h^\star(x) & : x \in z, \end{cases}$$

where $x \in z$ denotes the event that x is one of the elements of z.

Given mappings $h^\star : \mathcal{X} \to [K]$ and $\phi^\star : \mathcal{X}^{\leq q} \to [K]$, we define the class $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star, \phi^\star) \subset \mathcal{Y}^{\mathcal{X}}$ as follows:

$$\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star, \phi^\star) = \{h^\star\} \cup \{\mathsf{merge}(h^\star, \phi^\star, z) \,:\, z \in \mathcal{X}^{\leq q}\}.$$

Lemma I.1. *For any $h^\star : \mathcal{X} \to [K]$ and $\phi^\star : \mathcal{X}^{\le q} \to [K]$ for which $\phi^\star$ is injective, the DS dimension of $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$ is bounded as $d_{\mathsf{DS}}(\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)) \le 1$.*

Proof. Suppose for the purpose of contradiction that there were distinct $x_1,x_2 \in \mathcal{X}$ so that $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)|_{\{x_1,x_2\}}$ contains a 2-dimensional pseudo-cube. Certainly this pseudo-cube must have some element $(h(x_1),h(x_2))$ (indexed by $h \in \mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$) so that $h(x_1) \ne h^\star(x_1)$ and $h(x_2) \ne h^\star(x_2)$. (Indeed, take some element in this pseudo-cube, and for each $i \in \{1,2\}$ for which the ith coordinate is $h^\star(x_i)$, move to an i-neighbor.) But by definition of $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$, we must have $h(x_1) = h(x_2) = \phi^\star(z) \notin \{h^\star(x_1),h^\star(x_2)\}$ for some $z \in \mathcal{X}^{\le q}$, and thus, since $\phi^\star$ is injective, $h = \mathsf{merge}(h^\star,\phi^\star,z)$, meaning that $x_1,x_2 \notin z$. But it is also clear that $(h(x_1),h(x_2))$ cannot have any neighbor in $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)|_{\{x_1,x_2\}}$: any neighbor $(h'(x_1),h'(x_2))$ must satisfy $h'(x_1) = h^\star(x_1)$ or $h'(x_2) = h^\star(x_2)$, but there is no function $h' \in \mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star)$ with $h'(x_i) = h^\star(x_i)$ and $h'(x_{3-i}) = \phi^\star(z)$ for some $i \in \{1,2\}$ (since $x_1,x_2 \notin z$). $\square$

Theorem I.2. *For any $q \in \mathbb{N}$, there are domains $\mathcal{X},\mathcal{Y}$ satisfying $|\mathcal{Y}| \le q^{O(q)}$ so that the following holds. There is no algorithm Alg which satisfies the following guarantee: for any class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ with $d_{\mathsf{DS}}(\mathcal{H}) = 1$ together with a strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ for $\mathcal{H}$, Alg is a $(\mathcal{O}^{\mathsf{erm,s}};1/4,1/4)$-PAC learner for $\mathcal{H}$ with sample complexity and oracle complexity at most q.*

As a consequence of the bound $|\mathcal{Y}| \le q^{O(q)}$ in Theorem I.2 and the fact that $d_{\mathsf{N}}(\mathcal{H}) \le d_{\mathsf{DS}}(\mathcal{H})$, we observe the following: even with a strong ERM oracle, there is no realizable PAC learning algorithm with sample complexity and ERM query cost $o(\log(K)/\log\log(K))$, even for classes with Natarajan dimension 1. This lower bound is nearly tight, as simply returning an ERM on the sample yields error at most ϵ with $q = O(d_{\mathsf{N}}(\mathcal{H}) \cdot \log(K)/\epsilon)$ samples and overall query cost. (Daniely and Shalev-Shwartz, 2014) We additionally note that in the hard instance used to prove Theorem I.2, the distribution P has the additional property that its marginal over $\mathcal{X}$ is uniform.

Proof of Theorem I.2. Fix $q \in \mathbb{N}$, and set $N = 16q, K = 96q^2N^q$. We take $\mathcal{X} = \mathcal{X}_N = [N]$ and $\mathcal{Y} = \mathcal{Y}_K = [K]$, so that $|\mathcal{Y}| \le q^{O(q)}$ holds. Let $\mathcal{H}_0 := \mathcal{Y}^{\mathcal{X}} = [K]^{[N]}$ denote the space of all mappings $h : \mathcal{X} \to \mathcal{Y}$ and $\Phi := [K]^{[N]^{\le q}}$ denote the space of all mappings $\phi : \mathcal{X}^{\le q} \to [K]$. Let $\mathscr{U} := \mathrm{Unif}(\mathcal{H}_0 \times \Phi)$: in particular, for a tuple $(h^\star,\phi^\star) \sim \mathscr{U}$, the values $h^\star(x) \in \mathcal{Y}$ and $\phi^\star(z) \in \mathcal{Y}$ are all independent and uniform for all $x \in \mathcal{X}, z \in [N]^{\le q}$. Given $h^\star : \mathcal{X} \to \mathcal{Y}$, $\phi^\star : \mathcal{X}^{\le q} \to \mathcal{Y}$, and a subset $\mathcal{X}' \subset \mathcal{X}$, let $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$ denote an arbitrary strong ERM oracle for the class $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$ satisfying the following condition: if there is an empirical risk minimizer of the sample passed to $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$ of the form $\mathsf{merge}(h^\star,\phi^\star,z)$ for some $z \subset \mathcal{X}'$, then the oracle returns $\mathsf{merge}(h^\star,\phi^\star,z)$. Moreover, let $P_{h^\star} \in \Delta(\mathcal{X} \times \mathcal{Y})$ denote the uniform distribution over tuples $(x,h^\star(x))$, for $x \in \mathcal{X}$. Note that the distribution $P_{h^\star}$ is realizable with respect to the class $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$, for any $\phi^\star \in \Phi$.

Let us consider the execution of Alg with the class $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$ for a choice of $(h^\star,\phi^\star) \sim \mathscr{U}$. Let $(x_1,y_1),\ldots,(x_q,y_q) \in \mathcal{X} \times \mathcal{Y}$ denote the i.i.d. realizable sequence sampled with respect to $P_{h^\star}$. Let $\mathcal{X}' = \{x_1,\ldots,x_q\}$ – we will consider the interaction of Alg with the oracle $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$. For $1 \le t \le q$, let $(\hat{x}_t,\hat{y}_t) \in \mathcal{X} \times \mathcal{Y}$ denote the tth tuple in $\mathcal{X} \times \mathcal{Y}$ queried in the course of the oracle calls of Alg. (In particular, we concatenate the tuples of each oracle call and let $(\hat{x}_t,\hat{y}_t)$ denote the tth element in this concatenated list. As such, the ERM oracle will, in general, only return a hypothesis after receiving certain examples $(\hat{x}_t,\hat{y}_t)$ corresponding to the last datapoint in each dataset passed to it.) Let $\mathscr{F}_t$ denote the sigma-algebra generated by $(x_1,h^\star(x_1)),\ldots,(x_q,h^\star(x_q))$, $(\hat{x}_1,\hat{y}_1),\ldots,(\hat{x}_t,\hat{y}_t)$, the values of $\phi^\star(z)$ for $z \subset \{x_1,\ldots,x_q\}$, the internal randomness of Alg, and the results of all oracle calls (to $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$) which terminated at some step $s \le t$. Let $\mathcal{E}_t$ denote the event that all oracle calls terminating at some step $s \le t$ return an element of $\mathsf{merge}(h^\star,\phi^\star,z)$

for some $z \subset \{x_1,...,x_q\}$. Since $\mathsf{merge}(h^\star,\phi^\star,z)$ is $\mathscr{F}_t$-measurable for each $z \subset \{x_1,...,x_q\}$, the event $\mathcal{E}_t$ is $\mathscr{F}_t$-measurable.

Note that, conditioned on $\mathscr{F}_t$, $(\hat{x}_{t+1},\hat{y}_{t+1})$ is independent of $\mathbb{1}\{\mathcal{E}_t\}\cdot\mathbb{1}\{\hat{x}_{t+1}\not\in\{x_1,...,x_q\}\}\cdot h^\star(\hat{x}_{t+1})$ and $\mathbb{1}\{\mathcal{E}_t\}\cdot\phi^\star(z)$ for all $z \not\subset \{x_1,\dots,x_q\}$: this holds because $(\hat{x}_{t+1},\hat{y}_{t+1})$ is measurable with respect to $\mathscr{F}_t$, and conditioned on any instantiation of the random variables generating $\mathscr{F}_t$ for which $\mathcal{E}_t$ occurs, the values of $\mathbb{1}\{\mathcal{E}_t\}\cdot\phi^\star(z)$ for $z \not\subset \{x_1,\dots,x_q\}$ and of $\mathbb{1}\{\mathcal{E}_t\}\cdot h^\star(x)$ for $x \not\in \{x_1,\dots,x_q\}$ are all independently and uniformly distributed in $[K]$. Thus, conditioned on $\mathscr{F}_t$, with probability at least $1-(1+2qN^q)/K \geq 1-3qN^q/K$ (over $\mathscr{U}$ and the execution of Alg), we have that

$$\hat{y}_{t+1}\not\in(\{\mathbb{1}\{\mathcal{E}_t\}\cdot\mathbb{1}\{\hat{x}_{t+1}\not\in\{x_1,...,x_q\}\}\cdot h^\star(\hat{x}_{t+1})\}\cup\{\mathbb{1}\{\mathcal{E}_t\}\cdot\phi^\star(z)\,:\,z\not\subset\{x_1,...,x_q\}\}). \tag{32}$$

Let the event that (32) holds be denoted $\mathcal{G}_{t+1}$. Thus, under $\mathcal{E}_t \cap \bigcap_{s \leq t+1}\mathcal{G}_s$, for all $s \leq t+1$, all pairs $(\hat{x}_s,\hat{y}_s)$ for which $\hat{y}_s = h(\hat{x}_s)$ for some $h \in \mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$ must satisfy either $\hat{y}_s \in \{\phi^\star(z)\,:\,z \subset \{x_1,...,x_q\}\}$ or $\hat{x}_s \in \{x_1,...,x_q\}$. In particular, for any subset of the pairs $(\hat{x}_s,\hat{y}_s)$, there must be some empirical risk minimizer for this subset which belongs to $\{\mathsf{merge}(h^\star,\phi^\star,z)\,:\,z \subset \{x_1,...,x_q\}\}$. Thus, by definition of $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$, under $\mathcal{E}_t \cap \bigcap_{s\leq t+1}\mathcal{G}_s$, the empirical risk minimizer returned by $\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'}$ at step $t+1$ (if any) must belong to $\{\mathsf{merge}(h^\star,\phi^\star,z)\,:\,z \subset \{x_1,...,x_q\}\}$, i.e., $\mathcal{E}_{t+1}$ holds.

Since (32) gives that $\Pr_{\mathscr{U},\mathsf{Alg}}(\mathcal{G}_t) \geq 1 - 3qN^q/K$, a union bound gives that $\Pr\left(\bigcap_{t\leq q}\mathcal{G}_t\right) \geq 1-3q^2N^q/K$. Since $\mathcal{E}_0$ holds with probability 1 and $\mathcal{E}_{t+1}$ holds under $\mathcal{E}_t \cap \bigcap_{s\leq t+1}\mathcal{G}_s$, we have that $\mathcal{E}_q$ holds under the event $\bigcap_{t\leq q}\mathcal{G}_t$, i.e., with probability at least $1-3q^2N^q/K$. Write $\mathcal{E}^\star := \mathcal{E}_q \cap \bigcap_{s\leq t+1}\mathcal{G}_s$.

Let $H : \mathcal{X} \to \mathcal{Y}$ denote the output hypothesis of Alg; note that H is $\mathscr{F}_q$-measurable. We argued above that for any $x \not\in \{x_1,...,x_q\}$, conditioned on any instantiation of the random variables generating $\mathscr{F}_q$ for which $\mathcal{E}^\star$ (and thus $\mathcal{E}_q$) occurs, $h^\star(x)$ is uniformly distributed in $[K]$. Thus, $\Pr_{\mathscr{U},\mathsf{Alg}}(H(x) = \mathbb{1}\{\mathcal{E}^\star\}\cdot h^\star(x)) \leq 1/K$. Taking expectation over $(x,h^\star(x)) \sim P_{h^\star}$, we have that $\mathbb{E}_{\mathscr{U},\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[\mathbb{1}\{H(x) = \mathbb{1}\{\mathcal{E}^\star\}\cdot y\}] \leq 1/K + q/N$. Thus, by Markov's inequality, there is some subset $\mathcal{J} \subset \mathcal{H}_0 \times \Phi$ of measure $\mathscr{U}(\mathcal{J}) \geq 3/4$ so that for all $(h^\star,\phi^\star) \in \mathcal{J}$, we have $\mathbb{E}_{\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[\mathbb{1}\{H(x) = \mathbb{1}\{\mathcal{E}^\star\}\cdot y\}] \leq 2/K + 2q/N$. Since also $\mathcal{E}^\star$ occurs with probability at least $1-3q^2N^q/K \geq 31/32$ (over the choice of $(h^\star,\phi^\star) \sim \mathscr{U}$ and the execution of Alg), by Markov's inequality there is a subset $\mathcal{J}' \subset \mathcal{H}_0 \times \Phi$ of measure $\mathscr{U}(\mathcal{J}') \geq 3/4$ so that, for all $(h^\star,\phi^\star) \in \mathcal{J}'$, we have $\Pr_{\mathsf{Alg}}(\mathcal{E}^\star) \geq 7/8$.

Thus, for any $(h^\star,\phi^\star) \in \mathcal{J} \cap \mathcal{J}'$, we have $\Pr_{\mathsf{Alg}}(\mathcal{E}^\star) \geq 7/8$ and $\mathbb{E}_{\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[\mathbb{1}\{H(x) = \mathbb{1}\{\mathcal{E}^\star\}\cdot y\}] \leq 1/K + 2q/N$. Then

$$\mathbb{E}_{\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[\mathbb{1}\{H(x)=y\}] \leq \mathbb{E}_{\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[(1-\mathbb{1}\{\mathcal{E}^\star\}) + \mathbb{1}\{H(x) = \mathbb{1}\{\mathcal{E}^\star\}\cdot y\}] \leq 1/8 + 2/K + 2q/N < 1/2.$$

In particular, $\mathbb{E}_{\mathsf{Alg}}\mathbb{E}_{(x,y)\sim P_{h^\star}}[\ell^{\mathsf{mc}}(H(x),y)] > 1/2$, which implies that Alg cannot be a $(\mathcal{O}^{\mathsf{erm,s}}_{h^\star,\phi^\star,\mathcal{X}'};1/4,1/4)$-PAC learner for for $\mathcal{H}^{\mathsf{mc}}_{N,q}(h^\star,\phi^\star)$. $\square$

I.2. Lower bounds for realizable regression

In this section, we prove a lower bound for oracle-efficient learning in the setting of realizable regression. A recent paper (Attias et al., 2023) identified a combinatorial dimension depending on a scale parameter γ, called the *one-inclusion graph dimension at scale* γ of a class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ (denoted $d_{\mathsf{OIG},\gamma}(\mathcal{H})$), so that finiteness of $d_{\mathsf{OIG},\gamma}(\mathcal{H})$ at all scales γ characterizes real-valued learnability in the realizable setting. (In contrast, in the agnostic setting, recall that fat-shattering dimension is known to characterize learnability (Alon et al., 1997).) Moreover, (Attias et al., 2023) showed, roughly speaking, that the optimal sample complexity of PAC learning $\mathcal{H}$ scales nearly linearly with $d_{\mathsf{OIG},\gamma}(\mathcal{H})$ for an appropriate choice of γ. In

Theorem I.3 below, we will show that classes whose one-inclusion graph dimension is constant at all scales may not be learnable with respect to a strong ERM oracle. To do so, we need to consider a slightly different notion of strong ERM oracle, since a minimizer of empirical risk may not exist if the class $\mathcal{H}$ is infinite. For simplicity, we assume that $\mathcal{X}$ is finite (as it will be in our lower bound).

Definition I.1 (Strong limiting ERM oracle)**.** Consider a real-valued concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$. A *limiting strong ERM oracle* $\mathcal{O}^{\mathsf{erm,w}}$ for $\mathcal{H}$ is a mapping which takes as input a dataset $S \in (\mathcal{X} \times [0,1])^n$ and outputs a concept $h : \mathcal{X} \to [0,1]$ so that, for some sequence of hypotheses $h_n \in \mathcal{H}$, $\|h_n - h\|_\infty \to 0$ as $n \to \infty$ and $\lim_{n\to\infty} \widehat{\mathrm{er}}_{S,\ell^{\mathsf{abs}}}(h_n) = \inf_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{S,\ell^{\mathsf{abs}}}(h)$.

Note that an h as required in Definition I.1 always exists since $\mathcal{X}$ is finite, and thus $[0,1]^{\mathcal{X}}$ is compact.

Theorem I.3. *Consider any $q \in \mathbb{N}$ and algorithm* Alg *with access to a limiting strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ for a real-valued concept class. Suppose that* Alg *has sample complexity and oracle complexity at most $q/200$. Then there is a set $\mathcal{X}$ and a concept class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ which satisfies the following two conditions:*

1. Alg *is not a $(\mathcal{O}^{\mathsf{erm,s}}; 1/400, 1/400)$-PAC learner for $\mathcal{H}$.*
2. *For any $\mathcal{H}$-realizable distribution $P \in \Delta(\mathcal{X} \times [0,1])$, there is a PAC learning algorithm that achieves 0 error with 1 sample with probability 1.*

To avoid needing to formally define the one-inclusion graph dimension $d_{\mathsf{OIG},\gamma}(\cdot)$, we have not explicitly stated an upper bound on $d_{\mathsf{OIG},\gamma}(\mathcal{H})$ in the statement of Theorem I.3, instead directly stating its PAC learnability. Using the second item of the above theorem, the results of (Attias et al., 2023) imply that the class $\mathcal{H}$ satisfies $d_{\mathsf{OIG},\gamma}(\mathcal{H}) = O(1)$ for all $\gamma \in (0,1)$; alternatively, it can be verified directly that $d_{\mathsf{OIG},\gamma}(\mathcal{H}) = 1$ for all γ.

Proof of Theorem I.3. For $n \in \mathbb{N}$ and $q \in \mathbb{N}$, write $N_{n,q} := n^q$, and let p_n denote a prime number which is greater than $4n \cdot N_{n,q}$ and distinct from $p_{n-1}, \ldots, p_1$. We set $\mathcal{X} = [q]$. For any sequence $\bar{\sigma} = (\sigma_1, \sigma_2, \ldots)$, where $\sigma_n : [N_{n,q}] \to [N_{n,q}]$ is a permutation, we define a function class $\mathcal{H}_{\bar{\sigma}} \subset [0,1]^{\mathcal{X}}$, as follows. For $n \in \mathbb{N}$ and $1 \le i \le N_{n,q}$, let $g_{n,i} : \mathcal{X} \to \{0, 1/n, \ldots, (n-1)/n\}$ be the ith function (lexicographically) in $\{0, 1/n, \ldots, (n-1)/n\}^{\mathcal{X}}$. Now define

$$h_{n,i,\bar{\sigma}}(x) := \frac{\lceil g_{n,i}(x) \cdot p_n \rceil}{p_n} + \frac{\sigma_n(i)}{p_n}.$$

By our choice of p_n, we have that $\|h_{n,i}(x) - g_{n,i}\|_\infty \le (\sigma(i)+1)/p_n \le 2N_{n,q}/(4n \cdot N_{n,q}) < 1/n$. We now define $\mathcal{H}_{\bar{\sigma}} := \bigcup_{n \in \mathbb{N}} \{h_{n,i,\bar{\sigma}} : n \in \mathbb{N}, i \in [N_{n,q}]\}$. Note that for any function $f : \mathcal{X} \to [0,1]$ and $n \in \mathbb{N}$, there is some $i \in [N_{n,q}]$ so that $\|f - g_{n,i}\|_\infty \le 1/n$, meaning that $\|f - h_{n,i,\bar{\sigma}}\|_\infty \le 2/n$. Hence, for any $\bar{\sigma}$, $\mathcal{H}_{\bar{\sigma}}$ is dense in the space of functions $[0,1]^{\mathcal{X}}$ (with respect to $\|\cdot\|_\infty$). Thus, we may choose the following limiting strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ for the class $\mathcal{H}_{\bar{\sigma}}$: given a sample $(x_1,y_1), \ldots, (x_n,y_n)$ $\mathcal{O}^{\mathsf{erm,s}}$ returns the function which is 0 on all points $x \notin \{x_1, \ldots, x_n\}$, and which maps each x_i to $\mathrm{median}(\{y_j : x_j = x_i\})$. Note that $\mathcal{O}^{\mathsf{erm,s}}$ does not depend on $\bar{\sigma}$.

We next prove that for any $\bar{\sigma}$, the class $\mathcal{H}_{\bar{\sigma}}$ is learnable with a single sample: indeed, note that for any $x \in \mathcal{X}$, $n \in \mathbb{N}$, and $i \in [N_{n,q}]$, there is no hypothesis $h' \in \mathcal{H}_{\bar{\sigma}}$, $h' \neq h_{n,i,\bar{\sigma}}$, with $h'(x) = h_{n,i,\bar{\sigma}}(x)$, since p_n is prime for each n. Thus, the learning algorithm which sees a sample $(x, h^\star(x))$, for any $x \in \mathcal{X}$, can determine $h^\star$ from the value of $h^\star(x)$, and return $h^\star$.

Finally, we lower bound the performance of Alg, for some choice of $\bar{\sigma}$. Fix $n = 100$, and choose $\sigma_{n'}$ for $n' \neq n$ arbitrarily. Moreover, we let $\sigma_n : [N_{n,q}] \to [N_{n,q}]$ be a uniformly random permutation, $i^\star \sim \mathrm{Unif}([N_{n,q}])$, and $h^\star = h_{n,i^\star,\bar{\sigma}}$. We let P be the distribution $\mathrm{Unif}(\{(x,h^\star(x)) : x \in \mathcal{X}\})$.

We consider the performance of Alg in expectation over the distribution of $\bar{\sigma}$ and $h^\star$ that we have defined. Let $S = \{(x_1,y_1),...,(x_{q/200},y_{q/200})\}$ denote the i.i.d. sample from P that Alg receives. Note that, conditioned on S, the value of $i^\star$ is uniformly random over the set $\mathcal{I}_S := \{i \in [N_{n,q}] : g_{n,i,\bar{\sigma}}(x_j) = g_{n,i^\star}(x_j)\ \forall j \in [q/200]\}$. This uses the fact that the value of $\sigma_n^{-1}(i^\star) \in [N_{n,q}]$ is uniform and independent of $i^\star$. Since the responses to the queries to $\mathcal{O}^{\mathsf{erm,s}}$ do not depend on $\bar{\sigma}$, the output hypothesis H of Alg and $i^\star$ are conditionally independent, conditioned on S. Thus, conditioned on H, $i^\star$ is uniformly random over the set $\mathcal{I}_S$, which in particular means that the function $g_{n,i^\star}$ is distributed uniformly among all functions $g \in \{0,1/n,...,(n-1)/n\}^{\mathcal{X}}$ satisfying $g(x_j) = g_{n,i^\star}(x_j)$ for $j \in [q/200]$. Hence for any $x \notin \{x_1,...,x_{q/200}\}$, $\mathbb{E}_{i^\star,\bar{\sigma}}[|g_{n,i^\star}(x) - H(x)| \mid H] \geq 1/10$, which yields $\mathbb{E}_{i^\star,\bar{\sigma}}[|h_{n,i^\star,\sigma}(x) - H(x)| \mid H] \geq 1/10 - 2/n > 1/100$. Averaging over the distribution of H and of $x \sim \mathrm{Unif}(\mathcal{X})$ and using that $h^\star = h_{n,i^\star,\bar{\sigma}}$, we see that

$$\mathbb{E}_{i^\star,\bar{\sigma}} \mathbb{E}_{\mathsf{Alg}} \mathbb{E}_{(x,y)\sim P_{h^\star}}[|h^\star(x) - H(x)|] > 1/100 - (q/200)/q = 1/200.$$

Thus, there is some $\bar{\sigma}$ and $i^\star \in [N_{n,q}]$ so that, letting $h^\star = h_{n,i^\star}$, $\mathbb{E}_{\mathsf{Alg}} \mathbb{E}_{(x,y)\sim P_{h^\star}}[\ell^{\mathsf{abs}}(H(x),y)] > 1/200$. Hence Alg cannot be a $(\mathcal{O}^{\mathsf{erm,s}};1/400,1/400)$-PAC learner for $\mathcal{H}_{\bar{\sigma}}$, for some choice of $\bar{\sigma}$. □

Appendix J. Computational separation between weak and strong consistency oracle

We let FACTORING denote the following computational problem: given a positive integer n, represented in binary, in the event that $n = pq$ for primes p,q, then output p and q; otherwise, the output can be arbitrary.[9] It is widely believed that there is no polynomial-time algorithm for FACTORING (Lenstra, 2011).

Below we define a hypothesis class $\mathcal{H}^{\mathsf{primes}}$ for which a weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ (Definition 2.3) can be implemented in polynomial time, yet there is no polynomial-time algorithm which can implement a strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ (Definition 2.6), assuming that there is no polynomial-time algorithm for FACTORING. In fact, our proof shows that it is computationally hard to implement a strong ERM oracle even if it must only succeed on realizable datasets (such an oracle corresponds to the standard notion of consistency oracle, which returns a hypothesis consistent with the input dataset, if one exists).

We set $\mathcal{X} := \mathbb{N}$ and define the class $\mathcal{H}^{\mathsf{primes}} \subset \{0,1\}^{\mathcal{X}}$, as follows. For each pair of primes p,q, we define

$$h_{p,q}(x) := \begin{cases} 1 & : x \in \{p,q,pq\} \\ 0 & : \text{otherwise.} \end{cases}$$

Also write, for each $n \in \mathbb{N}$, $g_n(x) := \mathbb{1}\{x = n\}$. We then set $\mathcal{H}^{\mathsf{primes}} := \{h_{p,q} : p,q \text{ are prime}\} \cup \{g_n : n \in \mathbb{N} \text{ is not a product of two primes}\}$. Note that $d_{\mathsf{VC}}(\mathcal{H}^{\mathsf{primes}}) \leq 3$.

Proposition J.1. *The class $\mathcal{H}^{\mathsf{primes}}$ satisfies the following:*

1. *A weak consistency oracle $\mathcal{O}^{\mathsf{con,w}}$ for $\mathcal{H}^{\mathsf{primes}}$ can be implemented in polynomial time.*
2. *FACTORING reduces to the problem of implementing a strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ for $\mathcal{H}^{\mathsf{primes}}$.*

Proof. We first describe how a weak consistency oracle can be efficiently implemented. Given a dataset $S := \{(x_i,y_i)\}_{i\in[n]}$, we perform the following steps:

9. If one wishes to define a total problem, one can require that the output be 0 if n is not the product of 2 primes.

- Set $T := \{x_i : y_i = 1\}$.
- If $|T| = 0$, return True.
- If $|T| = 1$, let n denote the unique element of T. For each point in S of the form $(x_i, 0)$, check if n is a multiple of x_i. If so, and the ratio n/x_i is prime, then return False. At the end of the loop, if we have not yet returned, then return True.
- If at least two values in T are composite, return False.
- If at least three values in T are prime, return False.
- If T consists of two prime values whose product is m, then output True if and only if the point $(m, 0)$ does not appear in S.
- Otherwise, T consists of a composite value m and at least one prime value p; then output True if and only the point $(m/p, 0)$ does not appear in S.

The above steps may be efficiently implemented since there is a polynomial-time algorithm for determining whether a given natural number is prime (Agrawal et al., 2004).

Now consider a strong ERM oracle $\mathcal{O}^{\mathsf{erm,s}}$ for $\mathcal{H}^{\mathsf{primes}}$. Given a positive integer n, consider the hypothesis $\hat{h}$ returned by $\mathcal{O}^{\mathsf{erm,s}}(\{(n, 1)\})$. If n is a product of two primes p, q then the unique empirical risk minimizer is the hypothesis $h_{p,q}$, meaning that $\mathcal{O}^{\mathsf{erm,s}}$ must return this hypothesis and thereby yield the prime factors p, q. $\square$