Collaborative Learning with Different Labeling Functions

Yuyang Deng* Mingda Qiao†

Abstract

We study a variant of Collaborative PAC Learning, in which we aim to learn an accurate classifier for each of the n data distributions, while minimizing the number of samples drawn from them in total. Unlike in the usual collaborative learning setup, it is not assumed that there exists a single classifier that is simultaneously accurate for all distributions.

We show that, when the data distributions satisfy a weaker realizability assumption, which appeared in [CM12] in the context of multi-task learning, sample-efficient learning is still feasible. We give a learning algorithm based on Empirical Risk Minimization (ERM) on a natural augmentation of the hypothesis class, and the analysis relies on an upper bound on the VC dimension of this augmented class.

In terms of the computational efficiency, we show that ERM on the augmented hypothesis class is NP-hard, which gives evidence against the existence of computationally efficient learners in general. On the positive side, for two special cases, we give learners that are both sample- and computationally-efficient.

1 Introduction

In recent years, the remarkable success of data-driven machine learning has transformed numerous domains using the vast and diverse datasets collected from the real world. An ever-increasing volume of decentralized data is generated on a multitude of distributed devices, such as smartphones and personal computers. To better utilize these distributed data shards, we are faced with a challenge: how to effectively learn from these heterogeneous and noisy data sources?

Collaborative PAC Learning [BHPQ17] is a theoretical framework that abstracts the challenge above. In this model, there are n data distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$, from which we can adaptively sample. We are asked to learn n classifiers $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$, such that each $\hat{f}_i$ has an error at most ϵ on $\mathcal{D}_i$. The goal is to minimize the number of labeled examples that we sample from the n distributions in total.

Note that if we ignore the potential connection among the n learning tasks and solve them separately, the sample complexity is necessarily linear in n. Previously, [BHPQ17] introduced a sample-efficient algorithm when all distributions admit the same labeling function, i.e., some classifier in the hypothesis class has a zero error on every $\mathcal{D}_i$. Their algorithm has an $O((d+n)\log n)$ sample complexity, where d is the VC dimension of hypothesis class.[1] When d is large, the *overhead* of the sample complexity is significantly reduced from n to $\log n$.

*Pennsylvania State University. Email: yzd82@psu.edu.

†University of California, Berkeley. Email: mingda.qiao@berkeley.edu. Part of this work was done while the author was a graduate student at Stanford University.

[1] For brevity, we treat the accuracy and confidence parameters as constants here.

However, in the real world, it is often too strong an assumption that every data distribution is consistent with the *same* ground truth classifier. This is especially true when we are learning for a diverse population consisting of multiple sub-groups, each with different demographics and preferences. In light of this, we study a model of *collaborative learning with different labeling functions*. In particular, we aim to determine the conditions under which sample-efficient learning is viable when the data from different sources are labeled differently, and find the optimal sample complexity.

The contribution of this work is summarized as follows; see Section 1.2 for formal statements of our results.

- We formalize a model of collaborative learning with different labeling functions, and a sufficient condition, termed (k, ϵ)-*realizability*, for sample-efficient collaborative learning. This realizability assumption was used by [CM12] in the context of multi-task learning. Under this assumption, we give a learning algorithm with sample complexity $O(kd \log(n/k) + n \log n)$. This algorithm is based on Empirical Risk Minimizarion (ERM) over an augmentation of the hypothesis class.
- We show that the ERM problem over the augmented hypothesis class is always NP-hard when $k \geq 3$, and NP-hard for a specific hypothesis class when $k = 2$. This rules out efficient learners based on ERM, as well as *strongly proper* learners that always output at most k different classifiers in the hypothesis class.
- Finally, we identify two cases in which computationally efficient learning *is* possible. When all distributions share the same marginal distribution on $\mathcal{X}$, we give a simple polynomial-time algorithm that matches the information-theoretic bound. When the hypothesis class satisfies a "2-refutability" assumption, we give a different algorithm based on approximate graph coloring, which outperforms the naïve approach with an $\Omega(nd)$ sample complexity.

1.1 Problem Setup

We adopt the following standard model of binary classification: The hypothesis class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ is a family of binary functions over the instance space $\mathcal{X}$. A data distribution $\mathcal{D}$ is a distribution over $\mathcal{X} \times \{0,1\}$. The *population error* of a function $f : \mathcal{X} \to \{0,1\}$ on data distribution $\mathcal{D}$ is defined as

$$\mathrm{err}_{\mathcal{D}}(f) \coloneqq \Pr_{(x,y)\sim\mathcal{D}} [f(x) \neq y] .$$

A dataset is a multiset with elements in $\mathcal{X} \times \{0,1\}$. The *training error* of $f : \mathcal{X} \to \{0,1\}$ on dataset $S = \{(x_i, y_i)\}_{i\in[m]}$ is defined as

$$\mathrm{err}_{S}(f) \coloneqq \frac{1}{m} \sum_{i=1}^{m} \mathbb{1} \{f(x_i) \neq y_i\} .$$

The learning algorithm is given sample access to n data distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$. At each step, the algorithm is allowed to choose one of the n distributions (possibly depending on the previous samples) and draw a labeled example from it. The algorithm may terminate at any time and return n functions $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$. The learning algorithm is (ϵ, δ)-PAC if it satisfies

$$\Pr \left[\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \leq \epsilon, \ \forall i \in [n] \right] \geq 1 - \delta.$$

The sample complexity of the algorithm is the expected number of labeled examples sampled in total.

Note that the above is almost the same as the personalized setup (i.e., the algorithm may output different classifiers for different distributions) of the model of [BHPQ17], except that in their model, in addition, it is assumed that there exists a classifier in $\mathcal{F}$ with a zero error on every $\mathcal{D}_i$.

1.2 Our Results

A sufficient condition for sample-efficient learning. We start by stating a sufficient condition for n distributions to be learnable with a sample complexity that is (almost) linear in some parameter k instead of in n.

Definition 1 ((k,ϵ)-Realizability)**.** *Distributions $\mathcal{D}_1$, $\mathcal{D}_2$, …, $\mathcal{D}_n$ are (k,ϵ)-realizable with respect to hypothesis class $\mathcal{F}$, if there exist $f_1^*, f_2^*, \ldots, f_k^* \in \mathcal{F}$ such that $\min_{j\in[k]} \mathrm{err}_{\mathcal{D}_i}(f_j^*) \le \epsilon$ holds for every $i \in [n]$.*

In words, (k,ϵ)-realizability states that we can find k classifiers in $\mathcal{F}$, such that on each of the n distributions, at least one of the classifiers achieves a population error below ϵ.

Our first result is a general algorithm that efficiently learns under the (k,ϵ)-realizability assumption.

Theorem 1. *Suppose that $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ are (k,ϵ)-realizable with respect to hypothesis class $\mathcal{F}$. For any $\delta > 0$, there is an $(8\epsilon,\delta)$-PAC algorithm with sample complexity*

$$O\left(\frac{kd\log(n/k)\log(1/\epsilon)}{\epsilon} + \frac{n\log k\log(1/\epsilon) + n\log(n/\delta)}{\epsilon}\right).$$

Viewing ϵ and δ as constants, the sample complexity reduces to $O(kd\log(n/k) + n\log n)$. When d is large, the overhead in the sample complexity is $k\log(n/k)$, which interpolates between the $O(\log n)$ overhead at $k = 1$ (shown by [BHPQ17]) and the $O(n)$ overhead at $k = n$ (where the n learning tasks are essentially unrelated, and a linear overhead is unavoidable).

The factor 8 in the PAC guarantee can be replaced by any fixed constant that is strictly greater than 1, at the cost of a different hidden constant in the sample complexity. This follows from straightforward modifications to our proof.

While we prove Theorem 1 (as well as the other positive results in the paper) under the assumption that the learning algorithm is given the value of k, this assumption can be removed via a standard doubling trick: We consider a sequence of guesses on the value of k: $1 = k_1 < k_2 < k_3 < \cdots$, where each k_{i+1} is the smallest value of k such that the sample complexity bound is at least twice the bound for k_i. Then, we run the learning algorithm with k set to $k_1, k_2 - 1, k_2, k_3 - 1, k_3, \ldots$ in order, and terminate the algorithm as soon as we are convinced that the actual k is larger than the current guess. This procedure succeeds as soon as the guess exceeds the actual value of k, and the sample complexity only increases by a constant factor.

We prove Theorem 1 using the following natural augmentation of the instance space and the hypothesis class.

Definition 2 ((G,k)-Augmentation)**.** *Let $\mathcal{F}$ be a hypothesis class over instance space $\mathcal{X}$. For finite set G and $k \in [|G|]$, the (G,k)-augmentation of $\mathcal{F}$ is the hypothesis class $\mathcal{F}_{G,k}$ over $\mathcal{X}' := G \times \mathcal{X}$ defined as:*

$$\mathcal{F}_{G,k} := \left\{ g_{f,c} : f \in \mathcal{F}^k, c \in [k]^G \right\},$$

where for $f = (f_1, f_2, \ldots, f_k)$ and $c = (c_i)_{i \in G}$, $g_{f,c}$ is the function that maps $(i, x) \in \mathcal{X}'$ to $f_{c_i}(x)$.

When $G = [n]$ for some integer n, we use the shorthands "$\mathcal{F}_{n,k}$" and "(n, k)-augmentation".

Definition 2 becomes more natural in light of the following observation. For each $i \in [n]$, let $\mathcal{D}'_i$ be the distribution of $((i, x), y)$ when (x, y) is drawn from $\mathcal{D}_i$. Then, for any $f \in \mathcal{F}^k$ and $c \in [k]^n$, we have

$$\mathrm{err}_{\mathcal{D}'_i}(g_{f,c}) = \Pr_{((i,x),y)\sim\mathcal{D}'_i}[g_{f,c}(i, x) \neq y] = \Pr_{(x,y)\sim\mathcal{D}_i}[f_{c_i}(x) \neq y] = \mathrm{err}_{\mathcal{D}_i}(f_{c_i}).$$

In particular, when distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ are (k, ϵ)-realizable w.r.t. $\mathcal{F}$, by definition, there exist k classifiers $f_1, f_2, \ldots, f_k \in \mathcal{F}$ and n numbers $c_1, c_2, \ldots, c_n \in [k]$ such that $\mathrm{err}_{\mathcal{D}_i}(f_{c_i}) \leq \epsilon$. Then, the corresponding $g_{f,c}$ has a population error of at most ϵ on every $\mathcal{D}'_i$. This reduces the problem to an instance of collaborative learning on hypothesis class $\mathcal{F}_{n,k}$ and distributions $\mathcal{D}'_1$ through $\mathcal{D}'_n$, with a *single* unknown classifier that is simultaneously ϵ-accurate for all $\mathcal{D}'_i$ (i.e., the $(1, \epsilon)$-realizability assumption).

Our proof of Theorem 1 first upper bounds the VC dimension of $\mathcal{F}_{n,k}$ by a function of n, k, and the VC dimension of $\mathcal{F}$. Then, we adapt an algorithm of [BHPQ17] to achieve the sample complexity bound.

A sample complexity lower bound. Complementary to Theorem 1, our next result shows that the sample complexity can be lower bounded in terms of the sample complexity for the $(1, 0)$-realizable case.

Theorem 2. *Let $m(n, d, \epsilon, \delta)$ denote the optimal sample complexity of (ϵ, δ)-learning on a hypothesis class of VC dimension d and n distributions that are $(1, 0)$-realizable. Then, under the $(k, 0)$-realizable assumption, the sample complexity is lower bounded by $\Omega(k) \cdot m(\lfloor n/k \rfloor, d, \epsilon, O(\delta/k))$.*

Theorem 3.1 of [BHPQ17] bounds $m(n, d, \epsilon, \delta)$ by

$$O\left(\frac{\log n}{\epsilon}\left((d + n)\log(1/\epsilon) + n\log(n/\delta)\right)\right),$$

which contains a $(d \log n)/\epsilon$ term. Assuming that this term is unavoidable (i.e., $m(n, d, \epsilon, \delta) = \Omega((d \log n)/\epsilon)$), by Theorem 2, we have a lower bound of $\Omega\left(\frac{kd\log(n/k)}{\epsilon}\right)$ for the $(k, 0)$-realizable case. In other words, the leading term of the sample complexity in Theorem 1 is necessary. Proving such an $\Omega(d \log n)$ lower bound, however, is still an open problem, even under the additional restriction that the learner must output the same function for all the n distributions (see Problem 2 in the COLT'23 open problem of [AHZ23]).

Intractability of ERM and proper learning. A downside of Theorem 1 is that the learning algorithm might not be *computationally* efficient, even if there is a computationally efficient learner for $\mathcal{F}$ in the usual PAC learning setup. Concretely, our learning algorithm requires Empirical Risk Minimization (ERM) on $\mathcal{F}_{n,k}$, the (n, k)-augmentation of $\mathcal{F}$. The straightforward approach involves enumerating all partitions of $[n]$ into k sets, which takes exponential time.[2]

Our next result shows that this ERM problem generalizes certain intractable discrete optimization problems, and is unlikely to be efficiently solvable. We first give a formal definition of the an ERM oracle.

[2]There is a faster algorithm via dynamic programming, though its runtime is still $2^{\Omega(n)}$.

Definition 3 (ERM Oracle)**.** *An ERM oracle for hypothesis class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ is an oracle that, given any dataset S, returns $f^* \in \arg\min_{f \in \mathcal{F}} \mathrm{err}_S(f)$.*

To state the hardness result rigorously, we need to consider a parametrized family of hypothesis classes instead of a fixed one.

Definition 4 (Regular Hypothesis Family)**.** *A regular hypothesis family is $\{(\mathcal{X}_d, \mathcal{F}_d)\}_{d \in \mathbb{N}}$ that satisfies the following for every d:*

- *$\mathcal{F}_d$ is a collection of binary functions over $\mathcal{X}_d$ with VC dimension at least d.*
- *There is an efficient algorithm that, given d, outputs $x_1, x_2, \ldots, x_d \in \mathcal{X}_d$ that are shattered by $\mathcal{F}_d$.*

Remark 5. *The first condition prevents the family from containing only simple classes with bounded VC dimensions. The second condition allows us to efficiently find witnesses for the VC dimension. Note that the second condition holds for natural hypothesis classes such as halfspaces and parity functions, the VC dimension of which can be lower bounded in a constructive way.*

We will show that the following decision version of ERM is already hard for $\mathcal{F}_{n,k}$: instead of finding $f^* \in \arg\min_{f \in \mathcal{F}_{n,k}} \mathrm{err}_S(f)$, we are only required to decide whether $\min_{f \in \mathcal{F}_{n,k}} \mathrm{err}_S(f)$ is 0 or not.

Problem 1 (ERM over Augmented Classes)**.** *For a regular hypothesis family $\{(\mathcal{X}_d, \mathcal{F}_d)\}_{d \in \mathbb{N}}$, an instance of the ERM problem consists of parameters (d, n, k) and n datasets $S_1, S_2, \ldots, S_n \subseteq \mathcal{X}_d \times \{0,1\}$. The goal is to decide whether there exist classifiers $f_1, f_2, \ldots, f_k \in \mathcal{F}_d$ such that for every $i \in [n]$, $\min_{j \in [k]} \mathrm{err}_{S_i}(f_j) = 0$.*

Remark 6. *Problem 1 is equivalent to deciding whether there exists a classifier $f \in \mathcal{F}_{n,k}$ with a zero training error on the dataset $\{((i,x),y) : i \in [n], (x,y) \in S_i\}$. Therefore, if we could efficiently implement the ERM oracle for $\mathcal{F}_{n,k}$, we would be able to solve Problem 1 efficiently as well.*

Now we are ready to state our intractability result.

Theorem 3. *For any regular hypothesis family, ERM over augmented classes (Problem 1) is* NP*-hard for any $k \geq 3$. Furthermore, there exists a regular hypothesis family on which Problem 1 is polynomial-time solvable for $k = 1$ but* NP*-hard for $k = 2$.*

One might argue that Theorem 3 only addresses the worst case, and does not exclude the possibility of efficiently implementing ERM (with high probability) over datasets that are randomly drawn. In Section 5.3, we state and prove a "distributional" analogue of Theorem 3, which shows that it is also unlikely for an efficient (and possibly randomized) algorithm to succeed on randomly drawn samples.

Recall that a proper learner is one that always returns hypotheses in the hypothesis class. In our setup, we say that a learning algorithm is *strongly proper* if, when executed under the (k,ϵ)-realizability assumption, it always outputs n functions $\hat{f}_1, \ldots, \hat{f}_n \in \mathcal{F}$ such that $|\{\hat{f}_1, \ldots, \hat{f}_n\}| \leq k$. Note that the (k,ϵ)-realizability assumption implies that it is always possible to find accurate classifiers that satisfy this constraint. Unfortunately, our proof of Theorem 3 also implies that, unless $\mathsf{RP} = \mathsf{NP}$, no strongly proper learner can be computationally efficient in general.

Efficient algorithms for special cases. Despite the computational hardness in the general case, we identify two special cases in which computationally efficient learners exist, assuming an efficient ERM for $\mathcal{F}$.

The first case is when the n data distributions share the same marginal over $\mathcal{X}$.

Theorem 4. *Suppose that $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ are (k, ϵ)-realizable and have the same marginal distribution on $\mathcal{X}$. Fix constant $\alpha > 0$. For any $\delta > 0$, there is a $((3+\alpha)\epsilon, \delta)$-PAC algorithm that runs in $\mathrm{poly}(n, k, 1/\epsilon, \log(1/\delta))$ time, makes at most k calls to an ERM oracle for $\mathcal{F}$, and has a sample complexity of*

$$O\left(\frac{kd\log(1/\epsilon)}{\epsilon} + \frac{n\log(n/\delta)}{\epsilon}\right).$$

Our algorithm for the theorem above follows a similar approach to the lifelong learning algorithms of [BBV15, PU16].

Our next positive result applies to hypothesis classes that are 2-*refutable* in the sense that whenever a dataset cannot be perfectly fit by $\mathcal{F}$, it contains two labeled examples that explain this inconsistency.

Definition 7 (2-Refutability)**.** *A hypothesis class $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ is 2-refutable if, for any dataset S such that $\min_{f\in\mathcal{F}} \mathrm{err}_S(f) > 0$, there is $S' = \{(x_1, y_1), (x_2, y_2)\} \subseteq S$ such that $\min_{f\in\mathcal{F}} \mathrm{err}_{S'}(f) > 0$.*

The following gives examples of natural hypothesis classes that are 2-refutable, and shows that 2-refutability is preserved under certain operations.

Example 5. *The following hypothesis classes are 2-refutable:*

- *$\mathcal{F} = \{0,1\}^{\mathcal{X}}$. Any dataset that cannot be perfectly fit by $\mathcal{F}$ must contain both $(x, 0)$ and $(x, 1)$ for some $x \in \mathcal{X}$.*
- *$\mathcal{F} = \{f : \mathcal{X} \to \{0,1\} : \sum_{x\in\mathcal{X}} f(x) \leq 1\}$. Any dataset that cannot be perfectly fit by $\mathcal{F}$ must contain $(x_1, 1)$ and $(x_2, 1)$ for different $x_1, x_2 \in \mathcal{X}$.*
- *$\mathcal{F} = \{f' \circ g : f' \in \mathcal{F}'\}$, where $\mathcal{F}' \subseteq \{0,1\}^{\mathcal{X}'}$ is 2-refutable and $g : \mathcal{X} \to \mathcal{X}'$ is fixed.*
- *$\mathcal{F} = \{f' \oplus g : f' \in \mathcal{F}'\}$, where $\mathcal{F}' \subseteq \{0,1\}^{\mathcal{X}}$ is 2-refutable, $g : \mathcal{X} \to \{0,1\}$ is fixed, and $\oplus$ denotes pointwise XOR.*

Assuming that the hypothesis class is 2-refutable and the data distributions are $(k, 0)$-realizable, ERM on the augmented class $\mathcal{F}_{n,k}$ gets reduced to graph coloring, in light of the following definition and simple lemma.

Definition 8 (Conflict Graph)**.** *The conflict graph induced by datasets $S_1, S_2, \ldots, S_n$ and hypothesis class $\mathcal{F}$ is an undirected graph $G = ([n], E)$, where $\{i, j\} \in E$ if and only if $\min_{f\in\mathcal{F}} \mathrm{err}_{S_i \cup S_j}(f) > 0$.*

Lemma 9. *Let $\mathcal{F}$ be a 2-refutable hypothesis class. Datasets $S_1, \ldots, S_n$ satisfy $\min_{f\in\mathcal{F}} \mathrm{err}_{S_i}(f) = 0$ for every $i \in [n]$. Let V be an independent set in the conflict graph induced by $S_1, S_2, \ldots, S_n$ and $\mathcal{F}$. Then, for $S' = \bigcup_{i\in V} S_i$, it holds that $\min_{f\in\mathcal{F}} \mathrm{err}_{S'}(f) = 0$.*

Proof. Suppose for a contradiction that $\min_{f\in\mathcal{F}} \mathrm{err}_{S'}(f)$ is non-zero. Since $\mathcal{F}$ is 2-refutable, there exist $i_1, i_2 \in V$, $(x_1, y_1) \in S_{i_1}$ and $(x_2, y_2) \in S_{i_2}$ such that no classifier in $\mathcal{F}$ correctly labels both examples. If $i_1 = i_2$, this contradicts the assumption $\min_{f\in\mathcal{F}} \mathrm{err}_{S_{i_1}}(f) = 0$. If $i_1 \neq i_2$, i_1 and i_2 must be neighbours in the conflict graph, which contradicts the independence of V. □

Assuming that the datasets are drawn from distributions are $(k, 0)$-realizable, the induced conflict graph must be k-colorable. If we could find a valid k-coloring efficiently, each color corresponds to an independent set of the graph. By Lemma 9, we can call the ERM oracle for $\mathcal{F}$ to find a consistent function. Combining the functions for the k different colors gives a solution to the ERM problem over the augmented class $\mathcal{F}_{n,k}$.

Unfortunately, graph coloring is NP-hard when $k \geq 3$. Nevertheless, there are efficient algorithms for *approximate coloring*, i.e., color a graph using a few colors, when the graph is promised to be k-colorable for some small k. The definition below together with Theorem 6 gives a way of systematically translating an approximate coloring algorithm into an efficient algorithm for collaborative learning.

Definition 10. *For $k \geq 3$, let $c_k^* \in (0, 1]$ denote any constant such that any k-colorable graph with n vertices can be efficiently colored with $O(n^{c_k^*})$ colors.*

A result of Karger, Motwani and Sudan [KMS98] shows that we can take $c_k^* = 1 - \frac{3}{k+1} + \epsilon$ for any $\epsilon > 0$. For $k = 3$, a more recent breakthrough of Kawarabayashi and Thorup [KT17] gives $c_3^* = 0.19996$.

Theorem 6. *Suppose that $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ are $(k, 0)$-realizable with respect to a 2-refutable hypothesis class $\mathcal{F}$. For any $\delta > 0$, there is an (ϵ, δ)-PAC algorithm that runs in $\mathrm{poly}(n, k, 1/\epsilon, \log(1/\delta))$ time, makes $\mathrm{poly}(n)$ calls to an ERM oracle for $\mathcal{F}$, and has a sample complexity of*

$$O\left(\frac{d\log(1/\epsilon) + n}{\epsilon} \cdot \log n + \frac{n\log(1/\delta)}{\epsilon}\right)$$

if $k = 2$, and

$$O\left(\frac{d\log(1/\epsilon) + n}{\epsilon} \cdot n^{c_k^*} + \frac{n\log(1/\delta)}{\epsilon}\right)$$

if $k \geq 3$.

Note that when $k = 2$, the sample complexity is as good as the one in Theorem 1. When $k \geq 3$, the overhead increases from $\log n$ to $\mathrm{poly}(n)$. Nevertheless, this overhead is still sub-linear in n for any fixed k.

1.3 Related Work

Collaborative learning. Most closely related to our work are the previous studies of Collaborative PAC Learning [BHPQ17, NZ18, CZZ18, Qia18] and related fields such as multi-task learning [HK22], multi-distribution learning [HJZ22, AHZ23, Pen23, ZZC+23], federated learning [MMR+17, MSS19, CCD23] and multi-source domain adaptation [MMR08, KL19, MMR+21].

In multi-task learning/multi-source domain adaptation, there are n distributions, each with a fixed number of samples, and our goal is to use these samples to learn a hypothesis that has a small risk on some target distribution. A line of works [BDBC+10, KL19, MMR+21] studied the generalization risk in this scenario, but their bounds all depend on the discrepancy among n distributions and contain non-vanishing residual constants. To avoid this residual constant in the bounds, [HK22] considered a Bernstein condition assumption on the hypothesis class and some transferrability assumptions between the n source distributions and the target distribution. They studied the minimax rate of this learning scenario, and gave a nearly-optimal adaptation algorithm.

Specially, some works studied the multi-task linear regression [YZW+20, HXL+23]. [YZW+20] considered linear regression from multiple distributions, where all tasks share the same input feature covariate $\mathbf{X} \in \mathbb{R}^{m\times d}$, but with different labeling functions. They designed the Hard Parameter Sharing estimator and established an excess risk upper bound of $O(\frac{d\sigma^2}{m} + \text{heterogeneity})$. Recently, [HXL+23] proposed the *s-sparse heterogeneity* assumption among the labeling functions in multi-task linear regression, and designed an algorithm which achieves an $O(\frac{s\sigma^2}{m_i} + \frac{d\sigma^2}{\sum_{i=1}^n m_i})$ excess risk on the i-th task. Notice that when s is smaller than d, this bound is strictly sharper than individual learning bound $O(\frac{d\sigma^2}{m_i})$. The difference between our scenario and multi-task learning is that we allow the learner to draw an arbitrary number of samples from each distribution, instead of assuming that each distribution only has a fixed number of samples.

Federated learning is another relevant key learning scenario, where n players with their own underlying distributions, and fixed number of samples drawn from them, aim at learning model(s) that can have small risk on everyone's distribution. A line of works aimed at studying its statistical properties [CZLS23, CCD23, MSS19]. [CZLS23] studied the minimax risk of federated learning in logistic regression setting, and showed that the minimax risk is controlled by the heterogeneity among n distributions and their labeling functions. [CCD23] studied the risk bound of federated learning in the linear regression setting, and in an asymptotic fashion when the dimension of the model goes to infinity. Similar to multi-task learning, federated learning also assumes each player only has fixed number of samples, and the analysis does not give a PAC learning bound.

Multi-distribution learning was recently proposed by [HJZ22], where n players try to learn a single model $\hat{f}$, that can have an ϵ excess error on the worst case distribution among n players, i.e., $\max_{i\in[n]} \text{err}_{\mathcal{D}_i}(\hat{f}) \le \epsilon + \text{OPT}$. They gave an algorithm with sample complexity $O(\frac{d\log n}{\epsilon^2} + \frac{nd\log(d/\epsilon)}{\epsilon})$, and proved a lower bound of $\tilde{\Omega}(\frac{d+k}{\epsilon^2})$. Note that this is a more *pessimistic* learning guarantee than ours, since the value OPT can be very large. Very recently, two concurrent papers [Pen23, ZZC+23] gave algorithms that match this lower bound, resolving some of the open problems formulated in [AHZ23].

Mixture learning from batches. Another recent line of work [KSS+20, KSKO20, DJKS23, JSK+23] studied learning mixtures of linear regressions from data batches. In these setups, there are k unknown linear regression models. Each *data batch* consists of labeled examples produced by one of the linear models chosen randomly. This line of work gave trade-offs between the number of batches and the batch size in order for the parameters of the k linear models to be efficiently learnable.

In comparison, our model allows the learner to adaptively sample from the data distributions, whereas the batch sizes are fixed in the model of learning with batches. We also note that the results for learning with batches require assumptions on the marginal distribution, such as Gaussianity or certain hypercontractivity and condition number properties. Also, except the recent work of [JSK+23], they all required the marginal distribution of the instance to be the same across all batches.

Computational hardness of learning. There is a huge body of work on the computational hardness of learning. Early work along this line showed that, under standard complexity-theoretic assumptions, it is hard to properly and agnostically learn halfspaces [FGKP06, GR09] and boolean disjunctions [KSS92, Fel06]. More recent work have obtained a finer-grained understanding of this computational hardness. It is now known that many natural hypothesis classes are hard to

learn even under additional assumptions, e.g., learning halfspaces under Massart noise [DKMR22], agnostically learning halfspaces under Gaussian Marginals [DKR23], and properly learning decision trees using membership queries [KST23a, KST23b].

Approximate coloring. Approximate coloring is the problem of finding a valid coloring of a given graph with as few colors as possible. This line of work was initiated by Wigderson [Wig83], who gave an efficient algorithm that colors a 3-colorable graph with n vertices using $O(\sqrt{n})$ colors. This result was later improved by a series of work [BR90, Blu94, KMS98, BK97, ACC06, Chl07, KT17]. The best known upper bound of $O(n^{0.19996})$ is due to Kawarabayashi and Thorup [KT17]. The analogous problem for k-colorable graphs (where $k \geq 4$) has also been studied.

2 Discussion on Open Problems

Tighter sample complexity bounds. The most obvious open problem is to either improve the $kd\log(n/k)/\epsilon$ term in the sample complexity bound in Theorem 1 or prove a matching lower bound. In light of Theorem 2, it is sufficient to prove a lower bound of $\Omega((d\log n)/\epsilon)$ for the personalized setup of collaborative learning (i.e., the $(1,0)$-realizable case). Conversely, any improvement on this term implies a better algorithm for the $(1,0)$-realizable case.

A stronger hardness result. The NP-hardness is proved either for the ERM problem (in Theorem 3), or against learners that are strongly proper in the sense that they always return at most k different classifiers (recall the discussion in Section 1.2). Our result does not rule out efficient learners that are neither ERM-based nor strongly proper.[3] Can we prove the intractability of sample-efficient learning directly, at least for specific hypothesis classes?

Conflict graphs with bounded degrees. Our Theorem 6 gives a computationally efficient learner based on approximate coloring. It is also known that k-colorable graphs with the maximum degree bounded by Δ can be colored with a smaller number of colors (e.g., $\tilde{O}(\Delta^{1/3})$ colors when $k = 3$ [KMS98]). It is interesting to identify natural assumptions on the data distributions that ensure this small-degree property in the conflict graph, and explore whether that leads to a lower sample complexity.

Efficient learner for concrete hypothesis classes. Even when $\mathcal{F}$ is simply the class of all binary functions on an instance space of size d and the data distributions are $(k,0)$-realizable for $k = 3$, we do not have a computationally efficient learner that achieves the information-theoretic bound in Theorem 1. For this setup, since $\mathcal{F}$ is 2-refutable, Theorem 6 gives an algorithm with sample complexity of roughly $d{\cdot}n^{0.19996}/\epsilon$. Can we improve the overhead from $\mathrm{poly}(n)$ to $\mathrm{polylog}(n)$ via an efficient learner?

[3] In fact, the distributions that we constructed in the proof of Theorem 2 can be easily learned by an improper algorithm.

3 Sample Complexity Upper Bound

In this section, we prove Theorem 1, which upper bounds the sample complexity of collaborative learning under (k, ϵ)-realizability. The key step in the proof is the following upper bound on the VC dimension of $\mathcal{F}_{n,k}$.

Lemma 11. *For any $n \geq k \geq 1$ and hypothesis class $\mathcal{F}$ of VC dimension d, the VC dimension of $\mathcal{F}_{n,k}$ is at most $O(kd + n \log k)$.*

Lemma 11 implies that in general, $\mathcal{F}_{G,k}$ has a VC dimension of $O(kd + |G| \log k)$. To gain some intuition behind the bound in Lemma 11, suppose that $\mathcal{F}$ is a finite class of size 2^d. By Definition 2, the size of $\mathcal{F}_{n,k}$ is at most $|\mathcal{F}|^k \cdot k^n = 2^{kd + n \log_2 k}$, and the bound in the lemma immediately follows. The actual proof, of course, needs to deal with the case that $\mathcal{F}$ is larger or even infinite.

The lemma improves a previous result of [CM12, Theorem 1], which upper bounds the VC dimension of $\mathcal{F}_{n,k}$ (called the class of "hard k-shared task classifiers") by $O(kd \log(nkd) + n \log k)$. Note that this bound can be looser than the one in Lemma 11 by a logarithmic factor.

Proof of Lemma 11. We will show that, for some integer $m \geq kd$ to be chosen later, no m instances in $\mathcal{X}' = [n] \times \mathcal{X}$ can be shattered by $\mathcal{F}_{n,k}$. This upper bounds the VC dimension of $\mathcal{F}_{n,k}$ by $m - 1$.

Fix a set S of m elements in $\mathcal{X}'$. To bound the number of ways in which S can be labeled by $\mathcal{F}_{n,k}$, we also fix $c^* \in [k]^n$ and focus on the classifiers in $\mathcal{F}_{n,k}$ associated with c^*. Note that c^* naturally partitions S into $S_1 \cup S_2 \cup \cdots \cup S_k$, where $S_j \coloneqq \{(i, x) \in S : c^*_i = j\}$. Furthermore, let $X_j \coloneqq \{x \in \mathcal{X} : (i, x) \in S_j,\ \exists i \in [n]\}$ be the projection of S_j to $\mathcal{X}$. Note that we have

$$\sum_{j=1}^{k} |X_j| \leq \sum_{j=1}^{k} |S_j| = |S| = m.$$

Let $\Phi(\cdot)$ be the growth function of hypothesis class $\mathcal{F}$. Then, for each fixed $c^* \in [k]^n$, the number of ways in which S can be labeled by classifiers in $\{g_{f,c^*} : f \in \mathcal{F}^k\} \subseteq \mathcal{F}_{n,k}$ is at most

$$N_{c^*} \leq \prod_{j=1}^{k} \Phi(|X_j|).$$

By the Sauer-Shelah-Perles lemma, we have the following upper bound:

$$\Phi(m) \leq \overline{\Phi}(m) \coloneqq \begin{cases} e^m, & m \leq d, \\ \left(\frac{em}{d}\right)^d, & m > d. \end{cases}$$

It can be verified that the function $m \mapsto \ln \overline{\Phi}(m)$ is monotone increasing and concave on $[0, +\infty)$.

It then follows from $\sum_{j=1}^{k} |X_j| \le m$ that

$$\begin{aligned}
\ln N_{c^*} &\le k \cdot \frac{1}{k} \sum_{j=1}^{k} \ln \overline{\Phi}(|X_j|) \\
&\le k \cdot \ln \overline{\Phi}\left(\frac{1}{k} \sum_{j=1}^{k} |X_j| \right) && \text{(concavity)} \\
&\le k \cdot \ln \overline{\Phi}\left(\frac{m}{k} \right) && \text{(monotonicity)} \\
&= kd \ln \frac{em}{kd}. && (m \ge kd)
\end{aligned}$$

Then, summing over the k^n different choices of c^*, the logarithm of the growth function of $\mathcal{F}_{n,k}$ at m is at most

$$\ln \left(\sum_{c \in [k]^n} N_c \right) \le n \ln k + kd \ln \frac{em}{kd}.$$

For some sufficiently large $m = O(n \log k + kd)$, the above is strictly smaller than $\ln(2^m)$, which means that the m points in S cannot be shattered by $\mathcal{F}_{n,k}$. □

Given Lemma 11, Theorem 1 essentially follows from the learning algorithm of [BHPQ17] for the personalized setup, with slight modifications. For completeness, we state the algorithm and prove its correctness in Appendix A.

4 Sample Complexity Lower Bound

Our proof of Theorem 2 is based on a simple observation: learning n distributions under $(k, 0)$-realizability is at least as hard as learning k unrelated instances, where each instance consists of learning n/k distributions under $(1, 0)$-realizability.

Our actual proof is essentially the same as the lower bound proof of [BHPQ17], and formalizes the intuition above. Formally, assuming that a learning algorithm $\mathcal{A}$ (ϵ, δ)-PAC learns n distributions under $(k, 0)$-realizability, we use $\mathcal{A}$ to construct another learner $\mathcal{A}'$, which is $(\epsilon, O(\delta/k))$-PAC for n/k distributions that are $(1, 0)$-realizable. Furthermore, the sample complexity of $\mathcal{A}'$ is an $O(1/k)$ fraction of that of $\mathcal{A}$. For completeness, we state the reduction in the following.

Proof of Theorem 2. Fix parameters n, k, d, ϵ, and δ. Let $n' := \lfloor n/k \rfloor$ and $\delta' := \frac{10\delta}{9k}$. Let $\mathcal{F}$ be a hypothesis class with VC dimension d, and $\mathcal{D}^{\mathsf{hard}}$ be a distribution over hard instances for collaborative learning on n' distributions. Formally, $\mathcal{D}^{\mathsf{hard}}$ is a distribution over n' data distributions $(\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{n'})$ such that:

- Every $(\mathcal{D}_1, \ldots, \mathcal{D}_{n'})$ in the support of $\mathcal{D}^{\mathsf{hard}}$ is $(1, 0)$-realizable with respect to $\mathcal{F}$.
- If a learning algorithm achieves an (ϵ, δ')-PAC guarantee when learning $\mathcal{F}$ on $\mathcal{D}_1, \ldots, \mathcal{D}'_{n'}$ drawn from $\mathcal{D}^{\mathsf{hard}}$, it must take $m(n', d, \epsilon, \delta')$ samples in expectation.

Now, let $\mathcal{A}$ be an (ϵ, δ)-PAC learning algorithm over $k \cdot n'$ distributions under $(k, 0)$-realizability. For brevity, we relabel the $k \cdot n'$ distributions as $\mathcal{D}_{i,j}$ where $i \in [k]$ and $j \in [n']$. In the following, we construct another learning algorithm $\mathcal{A}'$ that learns n' distributions (denoted by $\mathcal{D}_1^{\mathsf{actual}}, \ldots, \mathcal{D}_{n'}^{\mathsf{actual}}$) drawn from $\mathcal{D}^{\mathsf{hard}}$ by simulating $\mathcal{A}$:

1. For each $i \in [k]$, independently draw $(\mathcal{D}_{i,1}, \ldots, \mathcal{D}_{i,n'})$ from $\mathcal{D}^{\mathsf{hard}}$.
2. Sample i^* from $[k]$ uniformly at random.
3. We simulate algorithm $\mathcal{A}$ on distributions $(\mathcal{D}_{i,j})_{i \in [k], j \in [n']}$, except that $\mathcal{D}_{i^*,1}$ through $\mathcal{D}_{i^*,n'}$ are replaced by the n' actual distributions $\mathcal{D}_1^{\mathsf{actual}}, \ldots, \mathcal{D}_{n'}^{\mathsf{actual}}$. In other words, whenever $\mathcal{A}$ requires a sample from $\mathcal{D}_{i,j}$, we truly sample from $\mathcal{D}_{i,j}$ if $i \neq i^*$; otherwise, we sample from $\mathcal{D}_j^{\mathsf{actual}}$, and forward the sample to $\mathcal{A}$.
4. When $\mathcal{A}$ terminates and outputs $(\hat{f}_{i,j})_{i \in [k], j \in [n']}$, we test if $\mathrm{err}_{\mathcal{D}_{i,j}}(\hat{f}_{i,j}) \leq \epsilon$ holds for all $i \neq i^*$ and $j \in [n']$. If so, we output $\hat{f}_{i^*,1}$ through $\hat{f}_{i^*,n'}$ as the answer; otherwise, we repeat the procedure above.

In each repetition of the above procedure, from the perspective of algorithm $\mathcal{A}$, it runs on $k \cdot n'$ distributions divided into k groups, where each group consists of n' distributions drawn from $\mathcal{D}^{\mathsf{hard}}$. Clearly, the $k \cdot n'$ distributions together satisfy $(k, 0)$-realizability. Intuitively, it is impossible for $\mathcal{A}$ to tell the index i^* that corresponds to the actual instance, so the actual instance only suffers from an $O(1/k)$ fraction of the error probability as well as the sample complexity.

Let M denote the expected number of samples that $\mathcal{A}$ draws on such a random instance. Analogous to Claims 4.3 and 4.4 from [BHPQ17][4], we have the following guarantees of the constructed learner $\mathcal{A}'$:

Claim 1. *Assuming $\delta \leq 0.1$, on a random instance drawn from $\mathcal{D}^{\mathsf{hard}}$, $\mathcal{A}'$ achieves an $(\epsilon, \frac{10\delta}{9k})$-PAC guarantee and draws at most $\frac{10M}{9k}$ samples in expectation.*

By our assumption on $\mathcal{D}^{\mathsf{hard}}$, we have $\frac{10M}{9k} \geq m(n', d, \epsilon, \delta')$. Hence, we conclude that $\mathcal{A}$ takes at least

$$\frac{9k}{10} \cdot m(n', d, \epsilon, \delta') = \Omega(k) \cdot m(\lfloor n/k \rfloor, d, \epsilon, O(\delta/k))$$

samples in expectation. □

5 Evidence of Intractability

In this section, we prove Theorem 3 as well as a distributional version of it.

5.1 Reduction from Graph Coloring

We prove the first part (the $k \geq 3$ case) by a reduction from graph coloring, which is well-known to be NP-hard.

[4]While the two claims in [BHPQ17] were proved for a concrete construction of hard instances, the proof only relies on the symmetry and independence among the instances, and thus can be applied to our case without modification.

Proof of Theorem 3 (the first part). Fix an arbitrary regular hypothesis family $\{(\mathcal{X}_d, \mathcal{F}_d)\}_{d\in\mathbb{N}}$. We will show that if, for any fixed $k \geq 3$, there is a polynomial-time algorithm that solves Problem 1, the same algorithm can be used to solve graph k-coloring efficiently. This implies the first part of the theorem.

Given an instance $G = (V, E)$ of the k-coloring problem, we construct an instance of Problem 1 with parameters k and $d = n = |V|$. Without loss of generality, we assume $V = [n]$, since we can always relabel the vertices. By Definition 4, the VC dimension of $\mathcal{F}_d$ is at least $d = n$, and we can efficiently find n instances $x_1, x_2, \ldots, x_n \in \mathcal{X}_d$ that are shattered by $\mathcal{F}_d$.

For each $v \in [n]$, we define the v-th dataset as

$$S_v \coloneqq \{(x_v, 1)\} \cup \{(x_u, 0) : \{u, v\} \in E\}.$$

We will show in the following that G has a k-coloring if and only if the ERM instance is feasible, i.e., the n datasets can be perfectly fit by k classifiers from $\mathcal{F}_d$.

From coloring to classifiers. Suppose that $c : V \to [k]$ is a valid k-coloring of G. Since $\mathcal{F}_d$ shatters $x_1, x_2, \ldots, x_n$, we can find $f_1, f_2, \ldots, f_k \in \mathcal{F}_d$ such that $f_i(x_v) \coloneqq \mathbb{1}\{c(v) = i\}$ holds for every $i \in [k]$ and $v \in [n]$. Then, for every $v \in [n]$, the dataset S_v is perfectly fit by classifier $f_{c(v)}$, since $f_{c(v)}(x_v) = \mathbb{1}\{c(v) = c(v)\} = 1$ and $f_{c(v)}(x_u) = \mathbb{1}\{c(u) = c(v)\} = 0$ for every neighbor u of v.

From classifiers to coloring. Conversely, let $f_1, f_2, \ldots, f_k \in \mathcal{F}_d$ be k classifiers such that each dataset S_v is consistent with one of the classifiers. We can then choose a labeling $c : V \to [k]$ such that S_v is perfectly fit by $f_{c(v)}$. Now we show that c is a valid k-coloring. Indeed, suppose that $\{u, v\} \in E$ is an edge and $c(u) = c(v) = i$. Then, f_i must correctly label both $(x_u, 1) \in S_u$ and $(x_u, 0) \in S_v$, which is impossible.

Finally, note that the above reduction works for any $k \geq 3$ and any family of hypothesis classes, while k-coloring is NP-hard for any $k \geq 3$. This proves the first part of Theorem 3. □

5.2 Hardness under Two Labeling Functions

Next, we deal with the $k = 2$ case. Since 2-coloring can be efficiently solved, we must reduce from a different NP-hard problem. Intuitively, we want the problem to correspond to partitioning a set into $k = 2$ parts. This motivates our reduction from (a variant of) subset sum.

Proof of Theorem 3, the second part. We start by defining the regular hypothesis family. For each integer $d \geq 1$, we consider the instance space $\mathcal{X}_d \coloneqq [d] \times \{0, 1, \ldots, 2^d\}$ and the following hypothesis class:

$$\mathcal{F}_d \coloneqq \left\{ f_\theta : \theta \in \{0, 1, \ldots, 2^d\}^d, \sum_{i=1}^{d} \theta_i \leq 2^d \right\},$$

where f_θ is defined as

$$f_\theta(i, j) = \mathbb{1}\{j \leq \theta_i\}.$$

In other words, each $f_\theta \in \mathcal{F}_d$ can be viewed as a direct product of d threshold functions, subject to that the thresholds sum up to at most 2^d. It can be easily verified that $\{(\mathcal{X}_d, \mathcal{F}_d)\}_{d\in\mathbb{N}}$ satisfies Definition 4, since for every d, the instances $(1, 1), (2, 1), \ldots, (d, 1)$ are shattered by $\mathcal{F}_d$, witnessed by $\{f_\theta : \theta \in \{0, 1\}^d\} \subseteq \mathcal{F}_d$.

Vanilla ERM is easy. We first show that the $k = 1$ case of Problem 1 is easy. Indeed, when $k = 1$, Problem 1 reduces to deciding whether $S_1 \cup S_2 \cup \cdots \cup S_n \subseteq \mathcal{X}_d \times \{0, 1\}$ can be perfectly fit by a hypothesis in $\mathcal{F}_d$. By construction of $\mathcal{F}_d$, this can be easily done via the following two steps:

- First, check whether there exist $i \in [d]$ and $0 \le j_1 < j_2 \le 2^d$ such that both $((i, j_1), 0)$ and $((i, j_2), 1)$ are in the dataset. Also check whether the dataset contains $((i, 0), 0)$ for any $i \in [d]$. If either condition holds, report "no solution".
- Then, for each $i \in [d]$, let θ_i denote the largest value $j \in \{0, 1, \ldots, 2^d\}$ such that the dataset contains $((i, j), 1)$; let $\theta_i = 0$ if no such labeled example exists. If $\sum_{i=1}^{d} \theta_i \le 2^d$, the function f_θ is a valid solution; otherwise, report "no solution".

The correctness of this procedure is immediate given the definition of $\mathcal{F}_d$.

Construction of datasets. We consider a specific choice of the datasets: for each $i \in [n]$, the i-th dataset contains exactly one data point of form (i, a_i) with label 1. In the rest of the proof, the key observation is that the datasets with indices in $T \subseteq [n]$ can be simultaneously satisfied by a hypothesis in $\mathcal{F}_d$ if and only if $\sum_{i \in T} a_i \le 2^d$. The ERM problem for $k = 2$ is then equivalent to deciding whether $\{a_i : i \in [n]\}$ can be partitioned into two sets, each of which sums up to $\le 2^d$. This problem can be easily shown to be NP-hard via a standard reduction from the subset sum problem.

From subset sum to a special case. We first reduce the general subset sum problem to a special case, in which the n numbers sum up to 2^{n+1} and the target value is exactly 2^n. Let $(\{a_i\}_{i \in [m]}, t)$ be an instance of subset sum (i.e., deciding whether there exists $T \subseteq [m]$ such that $\sum_{i \in T} a_i = t$). Let $s = \sum_{i=1}^{m} a_i$ and pick $n = \max\{m + 2, \lfloor \log_2 s \rfloor + 1\}$ such that $n - m \ge 2$ and $2^n > s$. We pad $n - m$ numbers to the instance, such that $a_{m+1} = a_{m+2} = \cdots = a_{n-2} = 0$, $a_{n-1} = 2^n - t$, $a_n = 2^n - (s - t)$. Note that the numbers now sum up to

$$\sum_{i=1}^{n} a_i = s + (2^n - t) + [2^n - (s - t)] = 2^{n+1}.$$

Also note that the size of the instance increases at most polynomially after the padding: A natural representation of the original subset sum instance $(\{a_i\}_{i \in [m]}, t)$ takes at least $m + \log_2 s$ bits. In the new instance, there are $n = O(m + \log s)$ numbers that sum up to 2^{n+1}, so its representation takes at most $O(n^2)$ bits, which is at most quadratic in the size of the original instance.

We claim that $(\{a_i\}_{i \in [n]}, 2^n)$ has the same answer as $(\{a_i\}_{i \in [m]}, t)$. Indeed, if $\sum_{i \in T} a_i = t$ for some $T \subseteq [m]$, $T \cup \{n - 1\}$ would be a feasible solution to the new instance. Conversely, suppose that $T \subseteq [n]$ is a subset such that $\sum_{i \in T} a_i = 2^n$. Since $a_{n-1} + a_n = 2^{n+1} - s > 2^n$, T must contain exactly one of $n - 1$ and n. Without loss of generality, we have $n - 1 \in T$, and $T \setminus \{n - 1\}$ would then give $\sum_{i \in T \setminus \{n-1\}} a_i = 2^n - (2^n - t) = t$.

From the special case to ERM. Then, we construct a collaborative learning instance with n distributions and set $d = n$ in the definition of $\mathcal{X}_d$ and $\mathcal{F}_d$. The i-th dataset only contains $((i, a_i), 1)$. We claim that the datasets can be fit by $k = 2$ functions in $\mathcal{F}_d$ if and only if the subset sum instance

$(\{a_i\}_{i\in[n]}, 2^n)$ is feasible. For the "if" direction, suppose that $T \subseteq [n]$ satisfies $\sum_{i\in T} a_i = 2^n$. Then, we define $\theta^{(1)}$ and $\theta^{(2)}$ as:

$$\theta_i^{(1)} = \begin{cases} a_i, & i \in T, \\ 0, & i \notin T, \end{cases} \quad \text{and} \quad \theta_i^{(2)} = \begin{cases} a_i, & i \notin T, \\ 0, & i \in T. \end{cases}$$

Clearly, both $f_{\theta^{(1)}}$ and $f_{\theta^{(2)}}$ are in $\mathcal{F}_d$. The i-th dataset is consistent with the first function if $i \in T$ and the second otherwise.

Conversely, suppose the datasets can be perfectly fit by two hypotheses $f_{\theta^{(1)}}, f_{\theta^{(2)}} \in \mathcal{F}_d$. Let $T \coloneqq \{i \in [n] : f_{\theta^{(1)}}(i, a_i) = 1\}$ be the indices of the data that are consistent with the former. We then have $\theta_i^{(1)} \geq a_i$ for every $i \in T$ and thus $\sum_{i\in T} a_i \leq \sum_{i\in T} \theta_i^{(1)} \leq 2^n$. The same argument, when applied to $[n] \setminus T$ and $\theta^{(2)}$, implies $\sum_{i\in[n]\setminus T} a_i \leq \sum_{i\in[n]\setminus T} \theta_i^{(2)} \leq 2^n$. Since the a_i's sum up to 2^{n+1}, we conclude that each summation must be equal to 2^n, i.e., T is a feasible solution to the subset sum instance. □

5.3 Hardness of the Distributional Version of ERM

As we mentioned earlier, Theorem 3 and its proof are arguably of a worst-case nature, and does not exclude the possibility of efficiently performing ERM (with high probability) over datasets that are randomly drawn. Indeed, for the first part of the proof (via a reduction from graph coloring), when the graph $G = (V, E)$ is dense, each dataset constructed in the proof would be of size $\Omega(|V|) = \Omega(d)$, whereas in the context of sample-efficient collaborative learning, the datasets tend to be much smaller.

Unfortunately, we can also prove the hardness of this distributional version, which we formally define below.

Problem 2 (ERM over Randomly Drawn Datasets)**.** *This is a variant of Problem 1, in which we specify n distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ over $\mathcal{X}_d \times \{0, 1\}$ and a parameter m. Each dataset S_i consists of m samples independently drawn from $\mathcal{D}_i$.*

We first note that the $k = 2$ part of Theorem 3 still holds for Problem 2. This is because our proof shows that Problem 1 is hard (for a specific hypothesis family) even if all the datasets are of size 1. Then, the same construction can be used to reduce subset sum to Problem 2, in which each distribution is a degenerate distribution.

To prove the hardness of Problem 2 when $k \geq 3$ and the hypothesis family is arbitrary, we will reduce from the following variant of the graph k-coloring problem, in which the graph is promised to have degrees bounded by $O(k)$.

Problem 3. *Given a graph $G = (V, E)$ with maximum degree at most $2k - 1$, decide whether G can be k-colored.*

Lemma 12. *Problem 3 is* NP*-hard.*

Proof. We reduce from the usual k-coloring. Let $G = (V, E)$ be an instance of k-coloring. We will give an efficient algorithm that transforms G into a graph G' that is a valid instance for Problem 3. Furthermore, G is k-colorable if and only if G' is k-colorable. This immediately implies the NP-hardness of Problem 3.

For each node $v \in V$, we split it into $|V|-1$ copies $v^{(1)}, v^{(2)}, \ldots, v^{(|V|-1)}$. We also add $|V|-2$ cliques of size $k-1$, denoted by $C_{v,1}, C_{v,2}, \ldots, C_{v,|V|-2}$. Every vertex in clique $C_{v,i}$ is also linked to $v^{(i)}$ and $v^{(i+1)}$. Note that this forces $v^{(1)}, v^{(2)}, \ldots, v^{(|V|-1)}$ to take the same color in a valid k-coloring. Then, for every edge $\{u, v\} \in E$, we link some copy $u^{(i)}$ of u to another copy $v^{(j)}$ of v, so that no copy of any vertex is used twice. The resulting graph $G' = (V', E')$ satisfies

$$|V'| = |V| \cdot [|V| - 1 + (|V| - 2) \cdot (k-1)] = O(k|V|^2),$$

and the maximum degree is $1 + 2(k-1) = 2k-1$. The equivalence between the k-colorability of G and G' is immediate from our construction. □

Now we prove a strengthening of Theorem 3.

Theorem 7. *Unless* NP = RP*, no polynomial-time (possibly randomized) algorithm for Problem 2 achieves the following guarantee for $k \geq 3$ and $m = \Omega(k \log n)$: With probability at least $1/\mathrm{poly}(d, n, m)$ over the randomness in $S_1, \ldots, S_n$, if $S_1, \ldots, S_n$ admits a feasible solution, the algorithm outputs a feasible solution with probability at least $1/\mathrm{poly}(d, n, m)$.*

Indeed, the guarantee required in Theorem 7 seems minimal for a useful ERM oracle: It only needs to succeed on a non-negligible fraction of instances, and the definition of "success" is merely to be able to output a feasible solution (if one exists) with a non-negligible probability. Still, it is unlikely to achieve such a guarantee efficiently under the standard computational hardness assumption of NP ≠ RP.

Proof. We prove the contrapositive: the existence of such algorithms implies NP = RP. Suppose that $\mathcal{A}$ is an efficient algorithm with the desired guarantees. We derive an efficient algorithm for Problem 3.

Given an instance of Problem 3, the reduction from the proof of Theorem 3 produces $n = |V|$ datasets $\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_n$, each of size at most $2k$. We define the i-th data distribution as the uniform distribution over $\hat{S}_i$. When m samples are drawn from each $\mathcal{D}_i$, every element in the support of every $\mathcal{D}_i$ appears at least once, except with probability at most

$$n \cdot 2k \cdot \left(1 - \frac{1}{2k}\right)^m \leq 2kn \cdot \exp\left(-\frac{m}{2k}\right),$$

which can be made much smaller than the $1/\mathrm{poly}(d, n, m)$ term in the theorem statement for some appropriate $m = \Omega(k \log n)$. In other words, except with a negligibly small probability, the randomly drawn datasets $S_1, \ldots, S_n$ coincide with the intended datasets $\hat{S}_1, \ldots, \hat{S}_n$ (when both are viewed as sets rather than multisets).

Then, the hypothetical algorithm $\mathcal{A}$ for Problem 2 must output the correct answer with probability larger than $1/\mathrm{poly}(d, n, m)$, when the sampled datasets are $\hat{S}_1, \ldots, \hat{S}_n$. We repeat $\mathcal{A}$ on $\hat{S}_1, \ldots, \hat{S}_n$ for $O(\mathrm{poly}(d, n, m))$ times and check whether it ever outputs a feasible solution. If so, we output "Yes"; we output "No" otherwise. This gives an efficient randomized algorithm for Problem 3 that: (1) when the input graph is k-colorable, outputs "Yes" with probability $\geq 1/2$; (2) when the graph is not k-colorable, always outputs "No". This implies NP = RP in light of Lemma 12. □

6 An Efficient Algorithm for Identical Marginals

In this section, we prove Theorem 4, which addresses the special case that $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ share the same marginal distribution over $\mathcal{X}$. In this case, we show that there is a simple algorithm that efficiently clusters the distributions and learn accurate classifiers using $\ll nd$ samples. The algorithm is formally defined as Algorithm 1, and follows a similar approach to the lifelong learning algorithms of [BBV15, PU16].

Algorithm 1 Efficient Clustering under Identical Marginals

Input: Hypothesis class $\mathcal{F}$. Sample access to $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Parameters $k, \epsilon, \delta, \alpha, c$.
Output: Hypotheses $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$.
$F \leftarrow \emptyset$;
for $i \in [n]$ **do**
  Draw $c \cdot \frac{\ln(n|F|/\delta)}{\epsilon}$ samples from $\mathcal{D}_i$ to form dataset S;
  $\hat{f} \leftarrow \arg\min_{f \in F} \mathrm{err}_S(f)$;
  if $\mathrm{err}_S(\hat{f}) \le (3 + \frac{2}{3}\alpha)\epsilon$ **then**
    $\hat{f}_i \leftarrow \hat{f}$;
  else
    Draw $c \cdot \frac{d\ln(1/\epsilon) + \ln[(|F|+1)/\delta]}{\epsilon}$ samples from $\mathcal{D}_i$ to form dataset S;
    $\hat{f}_i \leftarrow \arg\min_{f \in \mathcal{F}} \mathrm{err}_S(f)$;
    $F \leftarrow F \cup \{\hat{f}_i\}$;
  end if
end for
Return: $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$;

The algorithm maintains a list F of classifiers from class $\mathcal{F}$. For each distribution $\mathcal{D}_i$, we first test whether any classifier in F is accurate enough on it. If so, we set $\hat{f}_i$ as the best classifier in F; otherwise, we draw fresh samples to learn an accurate classifier for $\mathcal{D}_i$ and add it to $\mathcal{F}$.

Now we analyze Algorithm 1. We first define a good event that implies the accuracy and sample efficiency of the algorithm.

Definition 13. *Let $\mathcal{E}^{\mathsf{good}}$ denote the event that the following happen simultaneously when Algorithm 1 is executed:*

- *Whenever Line 5 is reached, it holds for every $f \in F$ that: (1) $\mathrm{err}_{\mathcal{D}_i}(f) \le (3 + \alpha/3)\epsilon$ implies $\mathrm{err}_S(f) \le \left(3 + \frac{2}{3}\alpha\right)\epsilon$; (2) $\mathrm{err}_S(f) > (3 + \alpha)\epsilon$ implies $\mathrm{err}_{\mathcal{D}_i}(f) > \left(3 + \frac{2}{3}\alpha\right)\epsilon$.*
- *Whenever Line 11 is reached, it holds that $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (1 + \alpha/3)\epsilon$.*

We show that $\mathcal{E}^{\mathsf{good}}$ happens with high probability, and implies that Algorithm 1 is $((3+\alpha)\epsilon, \delta)$-PAC and has a small sample complexity.

Lemma 14. *For any fixed $\alpha > 0$, there exists a sufficiently large $c > 0$ such that when Algorithm 1 is executed with parameters α and c, $\Pr\left[\mathcal{E}^{\mathsf{good}}\right] \ge 1 - \delta$.*

Proof. We upper bound the probability for each of the two conditions in $\mathcal{E}^{\mathsf{good}}$ to be violated.

For the first condition, we fix $i \in [n]$ and $f \in F$. We note that $\mathrm{err}_S(f)$ is the average of $c \cdot \frac{\ln(n|F|/\delta)}{\epsilon}$ independent Bernoulli random variables, each with expectation $\mathrm{err}_{\mathcal{D}_i}(f)$. By a Chernoff

bound, for sufficiently large constant c (that depends on α), it holds with probability $1 - \frac{\delta}{2n|F|}$ that: (1) $\mathrm{err}_{\mathcal{D}_i}(f) \le (3+\alpha/3)\epsilon$ implies $\mathrm{err}_S(f) \le \left(3+\frac{2}{3}\alpha\right)\epsilon$; (2) $\mathrm{err}_{\mathcal{D}_i}(f) > (3+\alpha)\epsilon$ implies $\mathrm{err}_S(f) > \left(3+\frac{2}{3}\alpha\right)\epsilon$. By a union bound over all $f \in F$, the first condition of $\mathcal{E}^{\mathsf{good}}$ holds for a specific $i \in [n]$ with probability at least $1 - \delta/(2n)$. By another union bound, the first condition holds for all $i \in [n]$ with probability at least $1 - \delta/2$.

For the second condition, suppose that we reach Line 11 at the i-th iteration of the for loop, and $|F| = r - 1$. Recall that the (k,ϵ)-realizability of $\mathcal{D}_1$ through $\mathcal{D}_n$ implies that there exists $f \in \mathcal{F}$ such that $\mathrm{err}_{\mathcal{D}_i}(f) \le \epsilon$. Then, by Theorem 5.7 of [AB99], for some sufficiently large c, we have $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (1+\alpha/3)\epsilon$ with probability at least $1 - \delta/(4r^2)$. Since $|F|$ is incremented whenever Line 11 is reached, we only need a union bound over all $r = 1, 2, \ldots, n$, and the probability for the second condition to be violated is upper bounded by

$$\sum_{r=1}^{n} \frac{\delta}{4r^2} \le \frac{\delta}{4} \sum_{r=1}^{+\infty} \frac{1}{r^2} = \frac{\delta}{4} \cdot \frac{\pi^2}{6} < \frac{\delta}{2}.$$

Finally, yet another union bound gives $\Pr\left[\mathcal{E}^{\mathsf{good}}\right] \ge 1 - \delta/2 - \delta/2 = 1 - \delta$. □

Lemma 15. *When $\mathcal{E}^{\mathsf{good}}$ happens, the outputs of Algorithm 1 satisfy $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (3+\alpha)\epsilon$ for every $i \in [n]$.*

Proof. Fix $i \in [n]$, and consider the i-th iteration of the for-loop in Algorithm 1. If the condition $\mathrm{err}_S(\hat{f}) \le \left(3+\frac{2}{3}\alpha\right)\epsilon$ holds, by the first condition in the definition of $\mathcal{E}^{\mathsf{good}}$, we must have $\mathrm{err}_{\mathcal{D}_i}(\hat{f}) \le (3+\alpha)\epsilon$. Then, by setting $\hat{f}_i$ to $\hat{f}$, we guarantee that $\hat{f}_i$ is $(3+\alpha)\epsilon$-accurate for $\mathcal{D}_i$. Otherwise, we pick $\hat{f}_i$ in Line 11, in which case the second condition of $\mathcal{E}^{\mathsf{good}}$ gives $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (1+\alpha)\epsilon \le (3+\alpha)\epsilon$. □

Lemma 16. *When $\mathcal{E}^{\mathsf{good}}$ happens, Algorithm 1 runs in $\mathrm{poly}(n, d, 1/\epsilon, \log(1/\delta))$ time, makes at most k calls to the ERM oracle, and takes*

$$O\left(\frac{kd\log(1/\epsilon)}{\epsilon} + \frac{n\log(n/\delta)}{\epsilon}\right)$$

samples.

Proof. The key of the proof is to show that when $\mathcal{E}^{\mathsf{good}}$ happens, $|F|$ is always at most k throughout the execution of Algorithm 1.

Upper bound $|F|$. Suppose towards a contradiction that $|F| > k$ at the end of Algorithm 1, while $\mathcal{E}^{\mathsf{good}}$ happens. By definition of (k,ϵ)-realizability from Definition 1, there exist $f_1^*, \ldots, f_k^* \in \mathcal{F}$ and $c \in [k]^n$ such that $\mathrm{err}_{\mathcal{D}_i}(f_{c_i}^*) \le \epsilon$ holds for every $i \in [n]$. By the pigeonhole principle, there exist $i < j$ such that $c_i = c_j$, and Algorithm 1 increments $|F|$ on both the i-th and the j-th iterations of the for-loop.

During the i-th iteration, we add $\hat{f}_i$ to F. By the second condition in the definition of $\mathcal{E}^{\mathsf{good}}$, we have $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (1+\alpha/3)\epsilon$. Now, we use the fact that $\mathcal{D}_i$ and $\mathcal{D}_j$ share the same marginal over $\mathcal{X}$, which we denote by $\mathcal{D}_x$. Define function $p_i : \mathcal{X} \to [0,1]$ as $p_i(x') \coloneqq \Pr_{(x,y)\sim\mathcal{D}_i}[y = 1 | x = x']$, i.e., the expectation of $y|x$ according to $\mathcal{D}_i$. Define $p_j(x') \coloneqq \Pr_{(x,y)\sim\mathcal{D}_j}[y = 1 | x = x']$ analogously. Note that we have the following relation for every $f : \mathcal{X} \to \{0,1\}$ and $i' \in \{i, j\}$:

$$\mathrm{err}_{\mathcal{D}_{i'}}(f) = \Pr_{(x,y)\sim\mathcal{D}_{i'}}[f(x) \ne y] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[\Pr_{y\sim\mathsf{Bernoulli}(p_{i'}(x))}[f(x) \ne y]\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[|f(x) - p_{i'}(x)|\right].$$

Since $c_i = c_j$, for every $x \in \mathcal{X}$ we have

$$\hat{f}_i(x) - p_j(x) = [\hat{f}_i(x) - p_i(x)] + [p_i(x) - f^*_{c_i}(x)] + [f^*_{c_j}(x) - p_j(x)].$$

It then follows from the triangle inequality that

$$\begin{aligned}
\mathrm{err}_{\mathcal{D}_j}(\hat{f}_i) &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[\left|\hat{f}_i(x) - p_j(x)\right|\right] \\
&\le \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[\left|\hat{f}_i(x) - p_i(x)\right|\right] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[\left|p_i(x) - f^*_{c_i}(x)\right|\right] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}\left[\left|f^*_{c_j}(x) - p_j(x)\right|\right] \\
&= \mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) + \mathrm{err}_{\mathcal{D}_i}(f^*_{c_i}) + \mathrm{err}_{\mathcal{D}_j}(f^*_{c_j}) \\
&\le (3 + \alpha/3)\epsilon.
\end{aligned}$$

The last step above applies $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le (1 + \alpha/3)\epsilon$, $\mathrm{err}_{\mathcal{D}_i}(f^*_{c_i}) \le \epsilon$, and $\mathrm{err}_{\mathcal{D}_j}(f^*_{c_j}) \le \epsilon$. The first inequality was proved earlier. The second and the third inequalities follow from our choice of $f^*_1, \ldots, f^*_k$ and $c \in [k]^n$.

Finally, by the first condition in the definition of $\mathcal{E}^{\mathsf{good}}$, $\mathrm{err}_{\mathcal{D}_j}(\hat{f}_i) \le (3 + \alpha/3)\epsilon$ implies that, during the j-th iteration of the for-loop, we have $\mathrm{err}_S(\hat{f}_i) \le \left(3 + \frac{2}{3}\alpha\right)\epsilon$ on Line 5. Then, we will not increment $|F|$ during the j-th iteration, which leads to a contradiction.

Oracle calls, runtime, and sample complexity. We first note that $|F|$ is incremented each time we call the ERM oracle on Line 11. Therefore, we make at most k calls to the ERM oracle. Other than this step, the remainder of Algorithm 1 can clearly be implemented in polynomial time. Finally, to bound the sample complexity, we note that in every iteration of the for-loop, $c \cdot \frac{\ln(n|F|/\delta)}{\epsilon} = O\left(\frac{\log(n/\delta)}{\epsilon}\right)$ samples are drawn. In addition, before each time $|F|$ is incremented, $O\left(\frac{d\log(1/\epsilon)+\log(k/\delta)}{\epsilon}\right)$ samples are drawn. Therefore, the total sample complexity is upper bounded by:

$$n \cdot O\left(\frac{\log(n/\delta)}{\epsilon}\right) + k \cdot O\left(\frac{d\log(1/\epsilon) + \log(k/\delta)}{\epsilon}\right) = O\left(\frac{kd\log(1/\epsilon) + n\log(n/\delta)}{\epsilon}\right).$$

□

Finally, we put everything together to prove Theorem 4.

Proof of Theorem 4. By Lemmas 14, 15 and 16, conditioning on an event that happens with probability at least $1 - \delta$, Algorithm 1 returns $(3 + \alpha)\epsilon$-accurate classifiers for all the n distributions, and the runtime, number of ERM oracle calls, and the number of samples are bounded accordingly.

In order to control the (unconditional) sample complexity, runtime, and number of oracle calls, we simply terminate the algorithm when any of these quantities exceeds the corresponding bound. The resulting algorithm is still $((3 + \alpha)\epsilon, \delta)$-PAC, and satisfies the desired upper bounds on the sample complexity, runtime, and number of oracle calls. □

7 Efficient Learning via Approximate Coloring

In this section, we prove Theorem 6, which gives efficient learning algorithms when the hypothesis class is 2-refutable.

7.1 The Learning Algorithm

The two cases are proved via a common strategy. In each iteration, we carefully choose a parameter m and draw m samples from each distribution. We use an approximate coloring algorithm to color the conflict graph (Definition 8) induced by the datasets. For each color that is used by considerably many vertices, we combine the corresponding datasets and fit a classifier to this joint dataset. The key is to argue that this classifier must be accurate for many distributions. Finally, we repeat the above on the distributions that have not received an accurate classifier.

We formally define a meta-algorithm in Algorithm 2. In the r-th iteration of the while-loop, we draw $m^{(r)}$ samples from each of the remaining distributions in $G^{(r)}$. We then build the conflict graph based on these datasets, and compute a $\gamma^{(r)}$-coloring of the graph. The vertices that receive color i are denoted by G_i, and $\hat{g}_i$ is chosen as an arbitrary classifier in $\mathcal{F}$ that is consistent with S_v for every $v \in G_i$. This choice is always possible by Lemma 9.

The algorithm is under-specified in three aspects: the number of samples $m^{(r)}$, the number of colors $\gamma^{(r)}$, as well as the algorithm for computing a $\gamma^{(r)}$-coloring. We will specify these choices when we prove Theorem 6 later.

Algorithm 2 Collaborative Learning via Approximate Coloring

Input: 2-refutable hypothesis class $\mathcal{F}$. Sample access to $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Parameters k, ϵ, δ, c.
Output: Hypotheses $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$.
$r \leftarrow 1$; $G^{(1)} \leftarrow [n]$;
while $G^{(r)} \neq \emptyset$ **do**
  $\delta^{(r)} \leftarrow \delta / r^2$;
  Set parameters $m^{(r)}$ and $\gamma^{(r)}$ according to $|G^{(r)}|$, d, ϵ, $\delta^{(r)}$;
  for $i \in G^{(r)}$ **do**
    Draw $m^{(r)}$ samples from $\mathcal{D}_i$ to form S_i;
  end for
  $(G^{(r)}, E) \leftarrow$ conflict graph of $\{S_i : i \in G^{(r)}\}$;
  Compute a $\gamma^{(r)}$-coloring of $(G^{(r)}, E)$. Let $G_i \subseteq G^{(r)}$ denote the set of vertices with color i;
  $G^{(r+1)} \leftarrow G^{(r)}$;
  for $i \in [\gamma^{(r)}]$ such that $|G_i| \geq |G^{(r)}|/(2\gamma^{(r)})$ **do**
    Find $\hat{g}_i \in \mathcal{F}$ such that $\mathrm{err}_{S_v}(\hat{g}_i) = 0$, $\forall v \in G_i$;
    for $v \in G_i$ **do**
      Draw $c \cdot \frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon}$ samples from $\mathcal{D}_v$ to form S_v;
      if $\mathrm{err}_{S_v}(\hat{g}_i) \leq \epsilon/2$ **then**
        $\hat{f}_v \leftarrow \hat{g}_i$;
        $G^{(r+1)} \leftarrow G^{(r+1)} \setminus \{v\}$;
      end if
    end for
  end for
  $r \leftarrow r + 1$;
end while
Return $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$;

7.2 Technical Lemmas

We state and prove a few technical lemmas for the analysis of Algorithm 2. The key of the analysis is the following lemma, which states that, as long as a sufficiently many vertices are colored with color i, the learned classifier $\hat{g}_i$ is good on average for the vertices with color i.

Lemma 17. *There is a universal constant $c > 0$ such that the following holds. In the r-th iteration of the while-loop, if $m^{(r)}$ is at least*

$$c \cdot \max\left\{\frac{\gamma^{(r)}}{|G^{(r)}|} \cdot \frac{d\ln(1/\epsilon) + |G^{(r)}| + \ln(1/\delta^{(r)})}{\epsilon}, \ln\frac{|G^{(r)}|}{\delta^{(r)}}\right\},$$

it holds with probability $1 - \delta^{(r)}/6$ that, for every $i \in [\gamma^{(r)}]$ such that $|G_i| \geq |G^{(r)}|/(2\gamma^{(r)})$,

$$\frac{1}{|G_i|}\sum_{v \in G_i} \mathrm{err}_{\mathcal{D}_v}(\hat{g}_i) \leq \epsilon/8.$$

The proof is based on similar techniques to the analysis of [Qia18] for a different variant of collaborative learning, in which a small fraction of the data sources are adversarial.

Proof. For brevity, we omit the superscripts in $m^{(r)}$, $\delta^{(r)}$ and $\gamma^{(r)}$. Let $M \coloneqq m \cdot \frac{|G^{(r)}|}{4\gamma}$. Consider the following thought experiment: We draw M independent samples $z_1^{(i)}, z_2^{(i)}, \ldots, z_M^{(i)}$ from each $\mathcal{D}_i$. (Recall that in Algorithm 2, only $m \ll M$ data points are actually drawn to form the dataset S_i.) Independently, for each non-empty $U \subseteq G^{(r)}$, we choose a sequence $A^{(U)} \in U^M$ uniformly at random. We may then consider the *fictitious dataset* $S^{(U)} = \left\{S_1^{(U)}, \ldots, S_M^{(U)}\right\}$ defined as:

$$S_i^{(U)} \coloneqq z_k^{(j)}, \text{ where } j = A_i^{(U)}, k = \sum_{l=1}^{i} \mathbb{1}\left\{A_l^{(U)} = j\right\}.$$

In words, for each $i \in U$, if entry i appears t times in sequence $A^{(U)}$, $S^{(U)}$ contains the first t data points collected from $\mathcal{D}_i$ (namely, $z_1^{(i)}, \ldots, z_t^{(i)}$). It can be easily verified that, over the randomness in all $z^{(i)}$ and $A^{(U)}$, each fictitious dataset $S^{(U)}$ is identically distributed as M samples from the uniform mixture $\mathcal{D}_U \coloneqq \frac{1}{|U|}\sum_{i \in U} \mathcal{D}_i$.

The rest of the proof consists of two parts: First, we show that with high probability, every $S^{(U)}$ is "representative" for distribution $\mathcal{D}_U$. Formally, any classifier in $\mathcal{F}$ with a zero training error on $S^{(U)}$ must have an $O(\epsilon)$ population error on $\mathcal{D}_U$. Then, we show that for each color $i \in [\gamma]$, the actual datasets (each of size m) collected from the distributions with color i *can* simulate the fictitious dataset $S^{(G_i)}$.

Step 1: Fictitious datasets are representative. For each fixed non-empty $U \subseteq G^{(r)}$, Theorems 28.3 and 28.4 of [SSBD14] imply that for some universal constant $c' > 0$,

$$\Pr\left[\forall f \in \mathcal{F}, \mathrm{err}_{S^{(U)}}(f) = 0 \implies \mathrm{err}_{\mathcal{D}_U}(f) \leq \epsilon/8\right] \geq 1 - (8/\epsilon)^d \cdot e^{-\epsilon M/c'}.$$

For $M \geq c' \cdot \frac{d\ln(8/\epsilon) + |G^{(r)}|\ln 2 + \ln(12/\delta)}{\epsilon}$, the right-hand side above is at least $1 - \frac{\delta}{12 \cdot 2^{|G^{(r)}|}}$. Then, a union bound over the $2^{|G^{(r)}|} - 1$ choices of U shows that with probability at least $1 - \delta/12$, for all non-empty $U \subseteq G^{(r)}$, any classifier in $\mathcal{F}$ that is consistent with $S^{(U)}$ has an error $\leq \epsilon/8$ on distribution $\mathcal{D}_U$.

Step 2: Fictitious datasets can be simulated. Fix $i \in [\gamma]$ such that $|G_i| \geq |G^{(r)}|/(2\gamma)$. Recall that the classifier $\hat{g}_i$ has a zero training error on $T_i := \bigcup_{v \in G_i} S_v$. In the first step, we showed that any $f \in \mathcal{F}$ that achieves a zero training error on $S^{(G_i)}$ must have a small population error on $\mathcal{D}_{G_i}$. Thus, it suffices to argue that $T_i \supseteq S^{(G_i)}$ with high probability.

Recall that we computed the coloring solely based on the datasets, which are independent of the indices $A^{(U)}$. Therefore, conditioning on the realization of $G_1, G_2, \ldots, G_\gamma$, each $A^{(G_i)}$ still uniformly distributed among G_i^M. In particular, for every $i \in [\gamma]$ and $v \in G_i$, the number of times v appears in $A^{(G_i)}$, denoted by $n_{i,v}$, follows the binomial distribution $\mathsf{Binomial}(M, 1/|G_i|)$. As long as $n_{i,v} \leq m$ for every (i, v) pair, each T_i (which contains the first m data points from $\mathcal{D}_v$) will be a superset of $S^{(G_i)}$ (which contains the first $n_{i,v}$ data points from $\mathcal{D}_v$).

There are at most $|G^{(r)}|$ such (i, v) pairs. The probability for each pair to violate the condition is at most

$$\begin{aligned}
\Pr_{X \sim \mathsf{Binomial}(M, 1/|G_i|)}[X \geq m] &\leq \Pr_{X \sim \mathsf{Binomial}(M, 1/|G_i|)}\left[X \geq \frac{4M\gamma}{|G^{(r)}|}\right] && (M = m|G^{(r)}|/(4\gamma)) \\
&\leq \Pr_{X \sim \mathsf{Binomial}(M, 2\gamma/|G^{(r)}|)}\left[X \geq \frac{4M\gamma}{|G^{(r)}|}\right]. && (|G_i| \geq |G^{(r)}|/(2\gamma))
\end{aligned}$$

By a Chernoff bound, the last expression is at most $\exp\left(-\frac{2M\gamma}{3|G^{(r)}|}\right)$, which can be made smaller than $\frac{\delta}{12|G^{(r)}|}$ since $M \geq c \cdot \frac{|G^{(r)}|}{4\gamma} \ln \frac{|G^{(r)}|}{\delta}$ for sufficiently large c. By a union bound, the aforementioned condition holds for all (i, v) pairs with probability at least $1 - \delta/12$.

Finally, the lemma follows from the two steps above and another union bound. □

As in the analysis in the previous section, we define a "good event" that implies the success of Algorithm 2.

Definition 18. *Let $\mathcal{E}^{\mathsf{good}}$ denote the event that the following happen simultaneously when Algorithm 2 is executed:*

- *The condition in Lemma 17 holds at every iteration r.*
- *Whenever Line 16 is reached, $\mathrm{err}_{\mathcal{D}_v}(\hat{g}_i) \leq \epsilon/4$ implies $\mathrm{err}_{S_v}(\hat{g}_i) \leq \epsilon/2$ and $\mathrm{err}_{\mathcal{D}_v}(\hat{g}_i) > \epsilon$ implies $\mathrm{err}_{S_v}(\hat{g}_i) > \epsilon/2$.*

Lemma 19. *When Algorithm 2 is executed with some sufficiently large constant c, it holds that $\Pr\left[\mathcal{E}^{\mathsf{good}}\right] \geq 1 - \delta$.*

Proof. By Lemma 17, the probability for the condition in Lemma 17 to be violated in the r-th iteration is at most $\delta^{(r)}/6$. Summing over all r gives $\sum_{r=1}^{+\infty} \frac{\delta^{(r)}}{6} = \frac{\delta}{6} \cdot \frac{\pi^2}{6} < \delta/3$. By the same argument as in the proof of Lemma 14, the probability for the second condition to be violated is also at most $\delta/3$. By a union bound, $\Pr\left[\mathcal{E}^{\mathsf{good}}\right] \geq 1 - \delta/3 - \delta/3 \geq 1 - \delta$. □

Analogous to Lemma 15, we have the following lemma, which states that event $\mathcal{E}^{\mathsf{good}}$ guarantees that the classifiers returned by the algorithm are accurate.

Lemma 20. *When $\mathcal{E}^{\mathsf{good}}$ happens, the output of Algorithm 2 satisfies $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \leq \epsilon$ for every $i \in [n]$.*

Finally, we prove that the number of active distributions, $|G^{(r)}|$, decreases at an exponential rate, so the while-loop is executed at most $O(\log n)$ times. This will be useful for upper bounding the sample complexity.

Lemma 21. *When event $\mathcal{E}^{\mathsf{good}}$ happens, $|G^{(r+1)}| \le \frac{3}{4}|G^{(r)}|$ holds at the end of the r-th iteration of the while-loop.*

Proof. Consider the r-th iteration of the while-loop. For brevity, we drop the superscript in $\gamma^{(r)}$. Since $\sum_{i=1}^{\gamma} |G_i| = |G^{(r)}|$, we have

$$\begin{aligned}
&\sum_{i=1}^{\gamma} |G_i| \cdot \mathbb{1}\left\{|G_i| \ge |G^{(r)}|/(2\gamma)\right\} \\
=& \sum_{i=1}^{\gamma} |G_i| - \sum_{i=1}^{\gamma} |G_i| \cdot \mathbb{1}\left\{|G_i| < |G^{(r)}|/(2\gamma)\right\} \\
\ge& |G^{(r)}| - \gamma \cdot \frac{|G^{(r)}|}{2\gamma} = |G^{(r)}|/2.
\end{aligned}$$

Fix $i \in [\gamma]$ that satisfies $|G_i| \ge |G^{(r)}|/(2\gamma)$. By the first condition in the definition of $\mathcal{E}^{\mathsf{good}}$, event $\mathcal{E}^{\mathsf{good}}$ implies that

$$\frac{1}{|G_i|} \sum_{v \in G_i} \mathrm{err}_{\mathcal{D}_v}(\hat{g}_i) \le \epsilon/8.$$

By Markov's inequality, there are at least $|G_i|/2$ elements $v \in G_i$ such that $\mathrm{err}_{\mathcal{D}_v}(\hat{g}_i) \le \epsilon/4$. Then, by the second condition in the definition of $\mathcal{E}^{\mathsf{good}}$, every such element v will not appear in $G^{(r+1)}$. Therefore, we conclude that

$$|G^{(r+1)}| \le |G^{(r)}| - \sum_{i=1}^{\gamma} \frac{|G_i|}{2} \cdot \mathbb{1}\left\{|G_i| \ge |G^{(r)}|/(2\gamma)\right\} \le |G^{(r)}| - \frac{|G^{(r)}|}{4} = \frac{3}{4}|G^{(r)}|.$$

□

7.3 The Bipartite Case

We start with the simpler case that $k = 2$. In this case, we set $m^{(r)}$ according to Lemma 17 and set $\gamma^{(r)} = 2$ in Algorithm 2. Furthermore, the coloring algorithm is simply the efficient algorithm for 2-coloring.

Proof of Theorem 6, the $k = 2$ case. In light of Lemmas 19 and 20, it remains to upper bound the sample complexity of Algorithm 2 under event $\mathcal{E}^{\mathsf{good}}$. As a simple corollary of Lemma 21, we have $|G^{(r)}| \le (3/4)^{r-1} \cdot n$ at the r-th iteration of the while-loop.

The sample complexity of the r-th iteration of the while-loop is upper bounded by

$$\begin{aligned}
& m^{(r)} \cdot |G^{(r)}| + |G^{(r)}| \cdot c \cdot \frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon} \\
\preceq\ & \frac{d\log(1/\epsilon) + |G^{(r)}| + \log(1/\delta^{(r)})}{\epsilon} + |G^{(r)}| \cdot \frac{\log(|G^{(r)}|/\delta^{(r)})}{\epsilon} \\
\preceq\ & \frac{d\log(1/\epsilon) + (3/4)^r \cdot n + \log(1/\delta) + \log r}{\epsilon} + (3/4)^r \cdot n \cdot \frac{\log[(3/4)^r \cdot n] + \log(1/\delta) + \log r}{\epsilon}.
\end{aligned}$$

Since the while-loop terminates when $G^{(r)}$ is empty, there are at most $O(\log n)$ iterations, and summing the above over all rounds gives

$$\begin{aligned}&\frac{d\log(1/\epsilon)\log n + n + \log(1/\delta)\log n + \log^2 n}{\epsilon} + \frac{n\log n + n\log(1/\delta)}{\epsilon}\\ \preceq\ &\frac{d\log(1/\epsilon)+n}{\epsilon}\cdot\log n + \frac{n\log(1/\delta)}{\epsilon}.\end{aligned}$$

Therefore, we have the desired sample complexity upper bound. □

7.4 The General Case

When $k \geq 3$, we can no longer find a k-coloring efficiently. Instead, we compute an approximate coloring with $O(n^{c_k^*})$ colors, where $n = |G^{(r)}|$ is the number of vertices in the graph. The hope is that as long as $c_k^* < 1$, we can still combine the datasets from vertices that share the same color, and use the data more efficiently.

Formally, let α be a constant such that there is an efficient algorithm that colors every k-colorable graph with n vertices using at most $\alpha \cdot n^{c_k^*}$ colors. We set $\gamma^{(r)} = \alpha \cdot |G^{(r)}|^{c_k^*}$ and set $m^{(r)}$ according to Lemma 17.

Proof of Theorem 6, the $k \geq 3$ case. Again, we focus on upper bounding the sample complexity. The number of samples drawn in the r-th round is at most

$$\begin{aligned}&m^{(r)} \cdot |G^{(r)}| + |G^{(r)}| \cdot c \cdot \frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon}\\ \preceq\ &|G^{(r)}|^{c_k^*} \cdot \frac{d\log(1/\epsilon) + |G^{(r)}| + \log(1/\delta^{(r)})}{\epsilon} + |G^{(r)}| \cdot \frac{\log(|G^{(r)}|/\delta^{(r)})}{\epsilon}.\end{aligned}$$

Plugging $|G^{(r)}| \leq (3/4)^{r-1} \cdot n$ into the above and summing over $r = 1, 2, \ldots$ gives

$$\begin{aligned}&\frac{dn^{c_k^*}\log(1/\epsilon) + n^{1+c_k^*} + n^{c_k^*}\log(1/\delta)}{\epsilon} + \frac{n\log n + n\log(1/\delta)}{\epsilon}\\ \preceq\ &\frac{d\log(1/\epsilon)+n}{\epsilon}\cdot n^{c_k^*} + \frac{n\log(1/\delta)}{\epsilon}.\end{aligned}$$

□

References

[AB99] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. 6, A

[ACC06] Sanjeev Arora, Eden Chlamtac, and Moses Charikar. New approximation guarantee for chromatic number. In *Symposium on Theory of Computing (STOC)*, pages 215–224, 2006. 1.3

[AHZ23] Pranjal Awasthi, Nika Haghtalab, and Eric Zhao. Open problem: The sample complexity of multi-distribution learning for vc classes. In *Conference on Learning Theory (COLT)*, pages 5943–5949, 2023. 1.2, 1.3

[BBV15] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory (COLT)*, pages 191–210, 2015. 1.2, 6

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010. 1.3

[BHPQ17] Avrim Blum, Nika Haghtalab, Ariel D Procaccia, and Mingda Qiao. Collaborative PAC learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2389–2398, 2017. 1, 1.1, 1.2, 1.2, 1.2, 1.3, 3, 4, 4, 4, A

[BK97] Avrim Blum and David Karger. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Information Processing Letters*, 61(1):49–53, 1997. 1.3

[Blu94] Avrim Blum. New approximation algorithms for graph coloring. *Journal of the ACM (JACM)*, 41(3):470–516, 1994. 1.3

[BR90] Bonnie Berger and John Rompel. A better performance guarantee for approximate graph coloring. *Algorithmica*, 5(1-4):459–466, 1990. 1.3

[CCD23] Gary Cheng, Karan Chadha, and John Duchi. Federated asymptotics: a model to compare federated learning algorithms. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 10650–10689, 2023. 1.3

[Chl07] Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Foundations of Computer Science (FOCS)*, pages 691–701, 2007. 1.3

[CM12] Koby Crammer and Yishay Mansour. Learning multiple tasks using shared hypotheses. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1475–1483, 2012. (document), 1, 3

[CZLS23] Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J. Su. Minimax estimation for personalized federated learning: An alternative between fedavg and local training? *Journal of Machine Learning Research*, 24(262):1–59, 2023. 1.3

[CZZ18] Jiecao Chen, Qin Zhang, and Yuan Zhou. Tight bounds for collaborative pac learning via multiplicative weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3602–3611, 2018. 1.3

[DJKS23] Abhimanyu Das, Ayush Jain, Weihao Kong, and Rajat Sen. Efficient list-decodable regression using batches. In *International Conference on Machine Learning (ICML)*, pages 7025–7065, 2023. 1.3

[DKMR22] Ilias Diakonikolas, Daniel Kane, Pasin Manurangsi, and Lisheng Ren. Cryptographic hardness of learning halfspaces with massart noise. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:3624–3636, 2022. 1.3

[DKR23] Ilias Diakonikolas, Daniel Kane, and Lisheng Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals. In *International Conference on Machine Learning (ICML)*, pages 7922–7938, 2023. 1.3

[Fel06] Vitaly Feldman. Optimal hardness results for maximizing agreements with monomials. In *Conference on Computational Complexity (CCC)*, pages 226–236, 2006. 1.3

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Foundations of Computer Science (FOCS)*, pages 563–574, 2006. 1.3

[GR09] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009. 1.3

[HJZ22] Nika Haghtalab, Michael Jordan, and Eric Zhao. On-demand sampling: Learning optimally from multiple distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 406–419, 2022. 1.3

[HK22] Steve Hanneke and Samory Kpotufe. A no-free-lunch theorem for multitask learning. *The Annals of Statistics*, 50(6):3119–3143, 2022. 1.3

[HXL+23] Xinmeng Huang, Kan Xu, Donghwan Lee, Hamed Hassani, Hamsa Bastani, and Edgar Dobriban. Optimal heterogeneous collaborative linear regression and contextual bandits. *arXiv preprint arXiv:2306.06291*, 2023. 1.3

[JSK+23] Ayush Jain, Rajat Sen, Weihao Kong, Abhimanyu Das, and Alon Orlitsky. Linear regression using heterogeneous data batches. *arXiv preprint arXiv:2309.01973*, 2023. 1.3

[KL19] Nikola Konstantinov and Christoph Lampert. Robust learning from untrusted sources. In *International Conference on Machine Learning (ICML)*, pages 3488–3498, 2019. 1.3

[KMS98] David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM (JACM)*, 45(2):246–265, 1998. 1.2, 1.3, 2

[KSKO20] Weihao Kong, Raghav Somani, Sham Kakade, and Sewoong Oh. Robust meta-learning for mixed linear regression with small batches. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4683–4696, 2020. 1.3

[KSS92] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. In *Annual Workshop on Computational Learning Theory (COLT)*, pages 341–352, 1992. 1.3

[KSS+20] Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. In *International Conference on Machine Learning (ICML)*, pages 5394–5404, 2020. 1.3

[KST23a] Caleb Koch, Carmen Strassle, and Li-Yang Tan. Properly learning decision trees with queries is np-hard. In *Foundations of Computer Science (FOCS)*, pages 2383–2407, 2023. 1.3

[KST23b] Caleb Koch, Carmen Strassle, and Li-Yang Tan. Superpolynomial lower bounds for decision tree learning and testing. In *Symposium on Discrete Algorithms (SODA)*, pages 1962–1994, 2023. 1.3

[KT17] Ken-Ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with less than n1/5 colors. *Journal of the ACM (JACM)*, 64(1):1–23, 2017. 1.2, 1.3

[MMR08] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1041–1048, 2008. 1.3

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017. 1.3

[MMR+21] Yishay Mansour, Mehryar Mohri, Jae Ro, Ananda Theertha Suresh, and Ke Wu. A theory of multiple-source adaptation with limited target labeled data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2332–2340, 2021. 1.3

[MSS19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning (ICML)*, pages 4615–4625, 2019. 1.3

[NZ18] Huy Nguyen and Lydia Zakynthinou. Improved algorithms for collaborative pac learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7631–7639, 2018. 1.3

[Pen23] Binghui Peng. The sample complexity of multi-distribution learning. *arXiv preprint arXiv:2312.04027*, 2023. 1.3

[PU16] Anastasia Pentina and Ruth Urner. Lifelong learning with weighted majority votes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3619–3627, 2016. 1.2, 6

[Qia18] Mingda Qiao. Do outliers ruin collaboration? In *International Conference on Machine Learning (ICML)*, pages 4180–4187, 2018. 1.3, 7.2

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 7.2

[Wig83] Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM (JACM)*, 30(4):729–735, 1983. 1.3

[YZW+20] Fan Yang, Hongyang R. Zhang, Sen Wu, Christopher Ré, and Weijie J. Su. Precise high-dimensional asymptotics for quantifying heterogeneous transfers. *arXiv preprint arXiv:2010.11750*, 2020. 1.3

[ZZC+23] Zihan Zhang, Wenhao Zhan, Yuxin Chen, Simon S Du, and Jason D Lee. Optimal multi-distribution learning. *arXiv preprint arXiv:2312.05134*, 2023. 1.3

A A Sample-Efficient Learning Algorithm

The algorithm is formally defined in Algorithm 3, and follows the same strategy as Algorithm 1 of [BHPQ17].

Algorithm 3 Collaborative Learning for (k, ϵ)-Realizable Distributions

1: **Input:** Hypothesis class $\mathcal{F}$. Sample access to $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Parameters k, ϵ, δ, c.
2: **Output:** Hypotheses $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n$.
3: $r \leftarrow 1$; $G^{(1)} \leftarrow [n]$;
4: **while** $|G^{(r)}| > k$ **do**
5: $\quad \delta^{(r)} \leftarrow \delta / r^2$;
6: $\quad d^{(r)} \leftarrow c \cdot (kd + |G^{(r)}| \log k)$;
7: $\quad$ Draw $c \cdot \frac{d^{(r)} \ln(1/\epsilon) + \ln(1/\delta^{(r)})}{\epsilon}$ samples from $\overline{D} := \frac{1}{|G^{(r)}|} \sum_{i \in G^{(r)}} \mathcal{D}'_i$ to form S;
8: $\quad g_{f,c} \leftarrow \arg\min_{g \in \mathcal{F}_{G^{(r)},k}} \mathrm{err}_S(g)$;
9: $\quad G^{(r+1)} \leftarrow \emptyset$;
10: $\quad$ **for** $i \in G^{(r)}$ **do**
11: $\quad\quad$ Draw $c \cdot \frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon}$ samples from $\mathcal{D}_i$ to form S_i;
12: $\quad\quad$ **if** $\mathrm{err}_{S_i}(f_{c_i}) \le 6\epsilon$ **then**
13: $\quad\quad\quad \hat{f}_i \leftarrow f_{c_i}$;
14: $\quad\quad$ **else**
15: $\quad\quad\quad G^{(r+1)} \leftarrow G^{(r+1)} \cup \{i\}$;
16: $\quad\quad$ **end if**
17: $\quad$ **end for**
18: $\quad r \leftarrow r + 1$;
19: **end while**
20: **for** $i \in G^{(r)}$ **do**
21: $\quad$ Draw $c \cdot \frac{d \ln(1/\epsilon) + \ln(k/\delta)}{\epsilon}$ samples from $\mathcal{D}_i$ to form S_i;
22: $\quad \hat{f}_i \leftarrow \arg\min_{f \in \mathcal{F}} \mathrm{err}_{S_i}(f)$;
23: **end for**
24: **Return:** $\hat{f}_1, \ldots, \hat{f}_n$;

Recall that for each data distribution $\mathcal{D}_i$, $\mathcal{D}'_i$ denotes the distribution of $((i, x), y)$ when $(x, y) \sim \mathcal{D}_i$. Therefore, on Line 7, to sample from the mixture distribution $\frac{1}{|G^{(r)}|} \sum_{i \in G^{(r)}} \mathcal{D}'_i$, it suffices to sample i from $G^{(r)}$ uniformly at random, draw $(x, y) \sim \mathcal{D}_i$, and then use the labeled example $((i, x), y)$.

The algorithm maintains $G^{(r)}$ as the set of active distributions at the beginning of the r-th round. The algorithm samples from the uniform mixture of the active distributions, learns a classifier $g_{f,c} \in \mathcal{F}_{G^{(r)},k}$ via ERM, and then tests whether the learned classifier is good enough for each distribution in $G^{(r)}$. If the learned classifier achieves an $O(\epsilon)$ empirical error on $\mathcal{D}_i$, we use it as the answer $\hat{f}_i$; otherwise, $\mathcal{D}_i$ stays active for the next round. Finally, the iteration terminates whenever the number of active distributions drops below k, at which point we naïvely learn on the $\le k$ remaining distributions separately.

The analysis of Algorithm 3 is straightforward, and relies on the following definition of a "good event".

Definition 22. *Let $\mathcal{E}^{\mathsf{good}}$ denote the event that the following happen simultaneously when Algorithm 3 is executed:*

- *Whenever Line 8 is reached, it holds that* $\mathrm{err}_{\overline{\mathcal{D}}}(g_{f,c}) \le 2\epsilon$.
- *Whenever Line 11 is reached, it holds that: (1)* $\mathrm{err}_{\mathcal{D}_i}(f_{c_i}) \le 4\epsilon$ *implies* $\mathrm{err}_{S_i}(f_{c_i}) \le 6\epsilon$*; (2)* $\mathrm{err}_{\mathcal{D}_i}(f_{c_i}) > 8\epsilon$ *implies* $\mathrm{err}_{S_i}(f_{c_i}) > 6\epsilon$.
- *Whenever Line 22 is reached, it holds that* $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le 2\epsilon$.

Then, Theorem 1 is a consequence of the following three lemmas.

Lemma 23. *For some universal constant c, when Algorithm 3 is executed with parameter c,* $\Pr\left[\mathcal{E}^{\mathsf{good}}\right] \ge 1 - \delta$.

Proof. Whenever Line 8 is reached, by Lemma 11, for some sufficiently large constant $c > 0$, the VC dimension of $\mathcal{F}_{G^{(r)},k}$ is upper bounded by $d^{(r)} = c \cdot (kd + |G^{(r)}| \log k)$. Then, it follows from Theorem 5.7 of [AB99] that, for some universal constant $c > 0$, the first condition is violated at the r-th round with probability at most $\delta^{(r)}/6$. By a union bound over all possible r, the first condition holds with probability at least $1 - \sum_{r=1}^{+\infty} \delta^{(r)}/6 \ge 1 - \delta/3$.

Again, by Theorem 5.7 of [AB99], the probability for the third condition to be violated for a specific i is at most $\delta/(3k)$. Since $|G^{(r)}| \le k$, by a union bound, the third condition holds with probability at least $1 - k \cdot \frac{\delta}{3k} = 1 - \delta/3$.

Finally, a Chernoff bound shows that the second condition holds for a specific r and $i \in G^{(r)}$ with probability at least

$$1 - 2\exp\left(-\Omega\left(c \cdot \frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon} \cdot \epsilon\right)\right),$$

which can be made greater than $1 - \frac{\delta^{(r)}}{6|G^{(r)}|}$ for sufficiently large c. By a union bound over all r and $i \in G^{(r)}$, the second condition holds with probability at least

$$1 - \sum_{r=1}^{+\infty} |G^{(r)}| \cdot \frac{\delta^{(r)}}{6|G^{(r)}|} \ge 1 - \delta/3.$$

The lemma follows from the three claims above and yet another union bound. □

Lemma 24. *When event $\mathcal{E}^{\mathsf{good}}$ happens, the output of Algorithm 3 satisfies* $\mathrm{err}_{\mathcal{D}_i}(\hat{f}_i) \le 8\epsilon$ *for every* $i \in [n]$.

Proof. We assign a classifier as $\hat{f}_i$ for some $i \in [n]$ either right after Line 11 or on Line 22. In either case, event $\mathcal{E}^{\mathsf{good}}$ guarantees that $\hat{f}_i$ is 8ϵ-accurate on $\mathcal{D}_i$. □

Lemma 25. *When event $\mathcal{E}^{\mathsf{good}}$ happens, Algorithm 3 terminates with a sample complexity of*

$$O\left(\frac{kd\log(n/k)\log(1/\epsilon)}{\epsilon} + \frac{n\log k\log(1/\epsilon) + n\log(n/\delta)}{\epsilon}\right).$$

Proof. We first control the size of $G^{(r)}$ in each round r. We claim that when event $\mathcal{E}^{\mathsf{good}}$ happens, if $G^{(r+1)}$ is defined during the execution of Algorithm 3, it holds that $|G^{(r+1)}| \le |G^{(r)}|/2$. Indeed, the first condition of $\mathcal{E}^{\mathsf{good}}$ guarantees that for $\overline{\mathcal{D}} = \frac{1}{|G^{(r)}|}\sum_{i\in G^{(r)}} \mathcal{D}'_i$,

$$\frac{1}{|G^{(r)}|}\sum_{i\in G^{(r)}} \mathrm{err}_{\mathcal{D}_i}\left(f_{c_i}\right) = \frac{1}{|G^{(r)}|}\sum_{i\in G^{(r)}} \mathrm{err}_{\mathcal{D}'_i}\left(g_{f,c}\right) = \mathrm{err}_{\overline{\mathcal{D}}}\left(g_{f,c}\right) \le 2\epsilon.$$

By Markov's inequality, it holds for at least half of the values $i \in G^{(r)}$ that $\mathrm{err}_{\mathcal{D}_i}\left(f_{c_i}\right) \le 4\epsilon$. Then, the second condition of $\mathcal{E}^{\mathsf{good}}$ guarantees that $|G^{(r+1)}| \le |G^{(r)}|/2$. It follows immediately that $|G^{(r)}| \le 2^{1-r}\cdot n$.

Then, the sample complexity of the r-th iteration of the while-loop is upper bounded by

$$\begin{aligned}
&c\cdot\frac{d^{(r)}\ln(1/\epsilon)+\ln(1/\delta^{(r)})}{\epsilon} + |G^{(r)}|\cdot c\cdot\frac{\ln(|G^{(r)}|/\delta^{(r)})}{\epsilon}\\
\preceq\ &\frac{(kd+2^{-r}\cdot n\log k)\log(1/\epsilon)+\log(1/\delta)+\log r}{\epsilon} + 2^{-r}\cdot n\cdot\frac{\log(2^{-r}\cdot n)+\log(1/\delta)+\log r}{\epsilon}.
\end{aligned}$$

Since the while-loop terminates whenever $|G^{(r)}| \le k$, there are at most $O(\log(n/k))$ iterations, and summing the above over $r = 1, 2, \ldots, O(\log(n/k))$ gives

$$\begin{aligned}
&\frac{kd\log(1/\epsilon)+\log(1/\delta)}{\epsilon}\cdot\log(n/k) + \frac{n\log k\log(1/\epsilon)}{\epsilon} + \frac{\log^2(n/k)}{\epsilon} + \frac{n\log n + n\log(1/\delta)}{\epsilon}\\
\preceq\ &\frac{kd\log(n/k)\log(1/\epsilon)}{\epsilon} + \frac{n\log k\log(1/\epsilon)+n\log(n/\delta)}{\epsilon}.
\end{aligned}$$

Finally, the last for-loop of the algorithm takes $O\left(\frac{kd\log(1/\epsilon)+k\log(k/\delta)}{\epsilon}\right)$ samples in total, which is always dominated by the above. Therefore, we have the desired upper bound on the sample complexity. □

Finally, we put all the pieces together and prove Theorem 1.

Proof of Theorem 1. Lemmas 23, 24, and 25 together imply that, conditioning on an event that happens with probability at least $1-\delta$, Algorithm 3 returns 8ϵ-accurate classifiers for each of the n distributions, while the number of samples is upper bounded by some

$$M = O\left(\frac{kd\log(n/k)\log(1/\epsilon)}{\epsilon} + \frac{n\log k\log(1/\epsilon)+n\log(n/\delta)}{\epsilon}\right).$$

To control the sample complexity—the (unconditional) expectation of the number of samples, we simply terminate the algorithm whenever the number of samples exceeds M. The resulting algorithm is still $(8\epsilon,\delta)$-PAC guarantee, and satisfies the desired upper bound on the sample complexity. □