On-Demand Sampling: Learning Optimally from Multiple Distributions*

Nika Haghtalab, Michael I. Jordan, and Eric Zhao

University of California, Berkeley
{nika,jordan,eric.zh}@berkeley.edu

Abstract

Social and real-world considerations such as robustness, fairness, social welfare and multi-agent tradeoffs have given rise to multi-distribution learning paradigms, such as *collaborative* [9], *group distributionally robust* [50], and *fair federated* learning [39]. In each of these settings, a learner seeks to uniformly minimize its expected loss over n predefined data distributions, while using as few samples as possible. In this paper, we establish the optimal sample complexity of these learning paradigms and give algorithms that meet this sample complexity. Importantly, our sample complexity bounds exceed that of learning a single distribution by only an additive factor of $\frac{n\log(n)}{\varepsilon^2}$. This improves upon the best known sample complexity bounds for fair federated learning (by Mohri et al. [39]) and collaborative learning (by Nguyen and Zakynthinou [42]) by multiplicative factors of n and $\frac{\log(n)}{\varepsilon^3}$, respectively. We also provide the first sample complexity bounds for the *group DRO* objective of Sagawa et al. [50]. To guarantee these optimal sample complexity bounds, our algorithms learn to sample from data distributions on demand. Our algorithm design and analysis are enabled by our extensions of online learning techniques for solving stochastic zero-sum games. In particular, we contribute stochastic variants of no-regret dynamics that can trade off between players' differing sampling costs.

1 Introduction

Pervasive needs for robustness, fairness, and multi-agent collaboration in learning have given rise to multi-distribution learning paradigms (e.g., [9, 50, 39, 18]). In these settings, we seek to learn a model that performs well on *any distribution* in a predefined set of interest. For fairness considerations, these distributions may represent heterogeneous populations of different protected or socioeconomic attributes; in robustness applications, they may capture a learner's uncertainty regarding the true underlying task; and in multi-agent collaborative or federated applications, they may represent agent-specific learning tasks. In these applications, the performance and optimality of a model is measured by its worst test-time performance on a distribution in the set. We are concerned with this fundamental problem of designing sample-efficient multi-distribution learning algorithms.

The sample complexity of multi-distribution learning differs from that of learning a single distribution in several ways. On one hand, varying numbers of samples are required when learning tasks of varying difficulty. On the other hand, similarity or overlap among learning tasks may obviate the need to sample from some distributions. This makes the use of a fixed per-distribution sample budget highly inefficient and suggests that optimal multi-distribution learning algorithms should *sample on demand*. That is, algorithms should take additional samples *whenever they need them* and *from whichever data distribution* they want them. On-demand sampling is especially appropriate when some population data is scarce (as in fairness mechanisms in which samples are amended [46]); when the designer can actively perturb datasets towards rare or atypical instances (such as in robustness applications [29, 59]); or when sample sets represent agents' contributions to an interactive multi-agent system [39, 10].

*Authors are ordered alphabetically. Correspondence to eric.zh@berkeley.edu.

Problem	Sample Complexity	Thm	Best Previous Result
Collab. Learning UB	$\varepsilon^{-2}(\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta))$	[5.1]	$\varepsilon^{-5}\log(\frac{1}{\varepsilon})\log(n/\delta)(\log(\lvert\mathcal{H}\rvert) + n)$ [42]
Collab. Learning LB	$\varepsilon^{-2}(\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta))$	[5.3]	$\varepsilon^{-1}(\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta))$ [9]
GDRO/AFL UB	$\varepsilon^{-2}(\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta))$	[5.1]	$\varepsilon^{-2}(n\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta))$ [39]
GDRO/AFL UB	$\varepsilon^{-2}(D_{\mathcal{H}} + n\log(n/\delta))$	[6.1]	N/A
(Training error convg.)	$\varepsilon^{-2}(D_{\mathcal{H}} + n\log(n/\delta))$	[6.2]	$\varepsilon^{-2}n(\log(n) + D_{\mathcal{H}})$ (expected convergence only) [50]

Table 1: This table lists upper (UB) and lower bounds (LB) on the sample complexity of learning a model class $\mathcal{H}$ on n distributions. For the collaborative learning and agnostic federated learning (AFL) settings, the sample complexity upper bounds refer to the problem of learning a (potentially randomized) model whose expected loss on each distribution is at most $\mathrm{OPT} + \varepsilon$, where OPT is the best possible such guarantee. For the GDRO setting, sample complexity refers to learning a deterministic model with expected losses of at most $\mathrm{OPT} + \varepsilon$, from a convex compact model space $\mathcal{H}$ with a Bregman radius of $D_{\mathcal{H}}$. Sample complexity bounds for collaborative and agnostic federated learning in existing works extend to VC dimension and Rademacher complexity. Our results also extend to VC dimension under some assumptions.

Blum et al. [9] demonstrated the benefit of on-demand sampling in the *collaborative learning* setting, when all data distributions are realizable with respect to the same target classifier. This line of work established that learning n distributions with on-demand sampling requires a factor of $\widetilde{O}(\log(n))$ times the sample complexity of learning a single realizable distribution [9, 13, 42], whereas relying on batched uniform convergence takes $\widetilde{\Omega}(n)$ times more samples than learning a single distribution [9]. However, beyond the realizable setting, the best known multi-distribution learning results fall short of this promise: existing on-demand sample complexity bounds for agnostic collaborative learning have highly suboptimal dependence on ε, requiring $\widetilde{O}(\log(n)/\varepsilon^3)$ times the sample complexity of agnostically learning a single distribution [42]. On the other hand, agnostic fair federated learning bounds [39] have been studied only for algorithms that sample in one large batch and thus require $\widetilde{\Omega}(n)$ times the sample complexity of a single learning task. Moreover, the test-time performance of some key multi-distribution learning methods, such as group distributionally robust optimization [50], have not been studied from a provable or mathematical perspective before.

In this paper, we give a general framework for obtaining *optimal and on-demand sample complexity* for three multi-distribution learning settings. Table 1 summarizes our results. All three of these settings consider a set $\mathcal{D}$ of n data distributions and a model class $\mathcal{H}$, evaluating the performance of a model h by its worst-case expected loss, $\max_{D\in\mathcal{D}} \mathcal{R}_D(h)$. As a benchmark, they consider the worst-case expected loss of the best model, i.e., $\mathrm{OPT} = \min_{h^*\in\mathcal{H}} \max_{D\in\mathcal{D}} \mathcal{R}_D(h^*)$. Notably, all of our sample complexity upper bounds demonstrate only an *additive increase of* $\varepsilon^{-2}n\log(n/\delta)$ *over the sample complexity of a single learning task*, compared to the multiplicative factor increase required by existing works.

- *Collaborative learning of Blum et al. [9]:* For agnostic collaborative learning, our Theorem 5.1 gives a randomized and a deterministic model that achieves performance guarantees of $\mathrm{OPT} + \varepsilon$ and $2\mathrm{OPT} + \varepsilon$, respectively. Our algorithms have an optimal sample complexity of $O(\frac{1}{\varepsilon^2}(\log(\lvert\mathcal{H}\rvert) + n\log(n/\delta)))$. This improves upon the work of Nguyen and Zakynthinou [42] in two ways. First, it provides risk bounds of $\mathrm{OPT} + \varepsilon$ for randomized classifiers, where only $2\mathrm{OPT} + \varepsilon$ was established previously. Second, it improves the upper bound of Nguyen and Zakynthinou [42] by a multiplicative factor of $\log(n)/\varepsilon^3$. In Theorem 5.3, we give a matching lower bound on this sample complexity, thereby establishing the optimality of our algorithms.

- *Group distributionally robust learning (group DRO) of Sagawa et al. [50]:* For group DRO, we consider a convex and compact model space $\mathcal{H}$. Our Theorem 6.1 studies a model that achieves an $\mathrm{OPT} + \varepsilon$ guarantee on the worst-case test-time performance of the model with an on-demand sample complexity of $O\left(\frac{1}{\varepsilon^2}(D_{\mathcal{H}} + n\log(n/\delta))\right)$. Our results also imply a high-probability bound for the convergence of group

DRO training error that improves upon the (expected) convergence guarantees of Sagawa et al. [50] by a factor of n.

- *Agnostic federated learning of [39]:* For agnostic federated learning, we consider a finite class of hypotheses. Our Theorems 5.1 and 6.1 show that on-demand sampling can accelerate the generalization of agnostic federated learning by a factor of n compared to batch results established by Mohri et al. [39]. Our results also imply matching high-probability bounds with respect to Mohri et al. [39] on the convergence of the training error in the batched setting.

To achieve these results, we frame multi-distribution learning as a stochastic zero-sum game: a maximizing player chooses a weight vector over data distributions $\mathcal{D}$ and a minimizing player chooses a weight vector over hypotheses $\mathcal{H}$. These two players require different numbers of datapoints in order to estimate their respective payoff vectors. We therefore solve the game using no-regret dynamics, utilizing stochastic mirror descent to optimally trade off the players' asymmetric needs for datapoints. In Section 3, we give an overview of this approach and its technical challenges and contributions. Our results also extend directly to settings with not only multiple data distributions but also multiple loss functions.

1.1 Related Work

There are many lines of work that study multi-distribution learning but which have evolved independently in separate communities.

Collaborative and agnostic federated learning. Blum, Haghtalab, Procaccia, and Qiao [9] posed the first fully general description of multi-distribution learning, motivated by the application of *collaborative PAC learning*. The field of *collaborative learning* is concerned with the learning of a shared machine learning model by multiple *stakeholders* that each desire a model with low error on their own data distribution. The line of work studies on-demand sample complexity bounds for the setting where stakeholders collect data so as to minimize the error of the worst-off stakeholder [9, 42, 13, 11]. This setting, stated in its full generality, yields the multi-distribution learning problem as presented in this paper. Blum et al. [9] established a $\log(n)$ factor blowup for the realizable case. For the general agnostic setting the best existing sample complexity requires a factor $\log(n)/\varepsilon^3$ blowup [42]. In comparison, our work establishes a tight additive increase in the sample complexity (which is comparable to $\log(n)$ multiplicative factor blowup with no dependence on ε). A related line of work concerns the strategic considerations of collaborative learning and seeks incentive-aware mechanisms for collecting data in the collaborative learning setting [10].

The field of *federated learning* focuses on a related motivating application where the goal is to learn a model from data dispersed across multiple devices but where querying data from each device is expensive [38]. The *agnostic* federated learning framework of Mohri, Sivek, and Suresh [39] poses (a variant of) the multi-distribution learning objective as a target for federated learning algorithms, and studies it in the offline setting with a data-dependent analysis. Their results involve a blowup by a factor n for the sample complexity.

Group distributionally robust optimization (Group DRO). Multi-distribution learning also arises in distributionally robust optimization [8] under the name of Group DRO, a class of DRO problems where the distributional uncertainty set is finite [24]. The group DRO literature is motivated by applications where the distributions correspond to deployment domains or protected demographics that a machine learning model should avoid spuriously linking to labels [24, 50, 51]. Although Group DRO—like collaborative learning—is mathematically an instance of multi-distribution learning, prior work on Group DRO focuses on the convergence of training error in offline settings, with a particular focus on deep learning applications. As we discuss later, theoretical aspects of on-demand multi-distribution learning can translate into actionable insights for Group DRO applications.

Multi-group fairness. Multi-distribution learning is also related to the fields of multi-group learning [49, 53] and multi-group fairness [19, 27]. These works study offline learning settings with a single distribution D and implicitly consider distribution D_i to be the conditional distribution on a subset of the support representing group i. In these settings, the learner does not have explicit access to oracles that sample from distributions

$D_1, \ldots, D_n$ and instead uses rejection sampling to collect data from $D_1, \ldots, D_n$. As a result, they experience a sub-optimal sample complexity blowup by a factor n. This blowup may not be obvious upon first glance, as these works provide theoretical guarantees for each group in terms of the number of datapoints from that group. Multi-group learning [49, 53] considers a similar problem to multi-distribution learning; by assuming that there exists a hypothesis that is simultaneously ε-optimal on every distribution (an assumption not made in our setting) they compare their learned hypothesis against the best hypothesis for each individual distribution.

Multi-source domain adaptation. Multi-source domain adaptation, or multi-task learning, is another related line of work that is concerned with using data from multiple different training distributions to learn some target distribution, under the assumption that the training and target distributions share some task relatedness [7, 36]. Multi-distribution learning can be framed similarly as using a finite set of training distributions to simultaneously learn the convex hull of the training distributions. Interestingly, the requirement in the multi-distribution setting of learning the entire convex hull obviates the need for the task-relatedness assumptions of multi-source learning.

Stochastic game equilibria. Our approach relates to a line of research on using online algorithms to find min-max equilibria by playing no-regret algorithms against one another [48, 21, 45, 14, 15]. Online mirror descent (OMD) is a well-studied family of methods that can find approximate minima of convex functions, and also find approximate min-max equilibria of convex-concave games, with high probability, using noisy first-order information [47, 40, 23, 6]. We bring these online learning tools to bear on the problem of finding saddle points in robust optimization formulations. The primary technical difference between multi-distribution learning and traditional saddle-point optimization problems is that we have sample access to data distributions instead of noisy local gradients.

Other paradigms. Several other machine learning paradigms also consider learning from multiple distributions. Notably, *distributed learning* (e.g., [44, 12, 5, 16, 52]) and *federated learning* (e.g., [32, 31, 38]) consider learning from data that is spread across multiple sources or devices. Classically, both of these settings have focused on minimizing the training or testing error *averaged* over these devices. The literature in these fields has primarily focused on methods for minimizing the average loss using communication-efficient, private, and robust-to-dropout training methods. However, optimizing average performance produces models that can significantly underperform on some data sources, especially when the data is heterogeneously spread across the sources. In comparison, multi-distribution learning paradigms such as collaborative learning [9], agnostic federated learning [39], and Group DRO [50] learn models that perform well across any one of the data sources.

Subsequent work. Haghtalab et al. [22] formalized multicalibration as a type of multi-distribution learning, building on the framework presented in this manuscript. Their work improves upon state-of-art multicalibration algorithms by implementing multi-distribution learning game dynamics using online learning algorithms that leverage the structure of calibration losses. Zhang et al. [61] extended the discussion on the sample complexity of Group DRO to settings with data budgets. They also noted an erroneous bandit-to-full-information reduction in an earlier version of this manuscript, which we corrected in a previous version (arXiv V2) with a minor change that employs Exp3 [41] or ELP [1] in place of our earlier reduction. Awasthi et al. [4] presented steps towards answering the sample complexity of multi-distribution learning with VC classes. This open problem was recently settled up to log factors by Zhang et al. [62], Peng [43].

2 Preliminaries

Throughout this manuscript, we use the shorthands $x^{(1:T)} \coloneqq x^{(1)}, \ldots, x^{(T)}$ and $f(\cdot, b) \coloneqq a \mapsto f(a, b)$. We write $\Delta(\mathcal{A})$ to denote the set of probability distributions supported on a set A and Δ_d to denote the probability simplex in $\mathbb{R}^{d-1}$. We use $\|\cdot\|_*$ to denote the dual of the norm $\|\cdot\|$ and $e_i \in \mathbb{R}^n$ to denote the ith standard basis vector. Given a data distribution D supported on the space of datapoints $\mathcal{Z}$, hypothesis

class $\mathcal{H}$, and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$, we denote the expected loss (risk) of a hypothesis $h \in \mathcal{H}$ by $\mathcal{R}_{D,\ell}(h) \coloneqq \mathbb{E}_{z\sim D}[\ell(h,z)]$, writing $\mathcal{R}_D(h)$ if ℓ is clear from context.

2.1 Multi-Distribution Learning

The goal of multi-distribution learning is finding a hypothesis that uniformly minimizes expected loss across multiple data distributions and loss functions. Importantly, we make no assumptions on the relationships between the data distributions; for example, we do not assume the existence of a hypothesis that is simultaneously optimal for every distribution. Formally, given a set of data distributions $\mathcal{D} = \{D_i\}_{i=1}^n$, losses $\mathcal{L} = \{\ell_j\}_{j=1}^m$, and a hypothesis class $\mathcal{H}$, we say a hypothesis h is ε-optimal for the multi-distribution learning problem $(\mathcal{D}, \mathcal{L}, \mathcal{H})$ if

$$\max_{D\in\mathcal{D}} \max_{\ell\in\mathcal{L}} \mathcal{R}_{D,\ell}(h) \leq \mathrm{OPT} + \varepsilon, \text{ where } \mathrm{OPT} \coloneqq \min_{h\in\mathcal{H}} \max_{D\in\mathcal{D}} \max_{\ell\in\mathcal{L}} \mathcal{R}_{D,\ell}(h). \tag{1}$$

Throughout this manuscript, we will often assume we are working with smooth and convex loss functions. Formally, we say a multi-distribution learning problem $(\mathcal{D}, \mathcal{L}, \mathcal{H})$ has smooth convex losses if two conditions are met. First, $\mathcal{H}$ is parameterized by a convex compact Euclidean parameter space Θ such that $\mathcal{H} = \{h_\theta\}_{\theta\in\Theta}$. Second, for the same parameter space Θ, for every loss function $\ell \in \mathcal{L}$ and datapoint $z \in \mathcal{Z}$, the mapping $f : \Theta \to [0,1]$ defined as $f(\theta) = \ell(h_\theta, z)$ is convex and 1-smooth; i.e., $\|\nabla_\theta f(\theta)\| \leq 1$ for all $\theta \in \Theta$. We remark that the assumption of smooth convex losses is a weak assumption. In fact, we will observe that our results on smooth convex losses easily extend to bounded non-smooth non-convex losses when the hypothesis class $\mathcal{H}$ is finite or combinatorially bounded, such as when $\mathcal{H}$ has finite VC dimension or Littlestone dimension [33].

Sample complexity. We are interested in the design of multi-distribution learning algorithms that have sample access to the distributions $D_1, \ldots, D_n$ and only take a small number of samples from these distributions overall. We formalize this access by defining a set of *example oracles*, $\mathrm{EX}(D_1), \ldots, \mathrm{EX}(D_n)$, where each $\mathrm{EX}(D_i)$ returns i.i.d. samples from D_i. We can then define the sample complexity of a multi-distribution learning algorithm by the cumulative number of calls it makes to these example oracles in order to find a solution.

We note that a multi-distribution learning algorithm may make these example oracle calls in an adaptive fashion; i.e., choosing which example oracle to call based on the datapoints it received from previous oracle calls. As first noted by Blum et al. [9], this ability to query for samples *on-demand* is critical for achieving efficient multi-distribution learning sample complexities. We also note that multi-distribution algorithms can use a set of example oracles to sample from any mixture distribution $q \in \Delta\mathcal{D}$; e.g., by first sampling a supporting distribution D_i from the mixture distribution and then calling its example oracle $\mathrm{EX}(D_i)$.

2.2 Instances of Multi-Distribution Learning

Multi-distribution learning unifies the problem formulations of collaborative learning [9], agnostic federated learning [39], and group distributionally robust optimization (group DRO) [50]. These problems have each spawned a line of highly influential works but were previously not recognized to be equivalent. We emphasize our view that multi-distribution learning is a particularly useful level of generality at which to study these problems, as it allows for their unified treatment both conceptually and algorithmically.

Collaborative learning. In the *collaborative PAC learning model* of Blum et al. [9], and its agnostic extensions by Nguyen and Zakynthinou [42], the goal is to learn a hypothesis that guarantees small risk for every distribution in a collection of distributions. These data distributions are usually interpreted as the heterogeneous problem domains faced by multiple participants that are collaborating on data collection; the goal of collaborative learning is to learn a machine learning model that all participants are satisfied with.

Collaborative learning is usually studied in a supervised learning setting where datapoints consist of a feature-label pair, i.e., $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, and where hypothesis classes $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ are either finite or combinatorially bounded. Importantly, loss functions are assumed to be bounded in $[0,1]$, but may be non-smooth and non-convex. Formally, given a set of data distributions, $\mathcal{D} \coloneqq \{D_1, \ldots, D_n\}$, supported on $\mathcal{X} \times \mathcal{Y}$, a loss

function $\ell : \mathcal{Y}^{\mathcal{X}} \times \mathcal{Z} \to [0,1]$, and a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, a collaborative learning instance, $(\mathcal{H}, \mathcal{D})$, is formulated as the problem of finding a solution $h \in \mathcal{Y}^{\mathcal{X}}$ such that

$$\max_{D \in \mathcal{D}} \mathcal{R}_D(h) \leq \mathrm{OPT} + \varepsilon, \text{ where } \mathrm{OPT} \coloneqq \min_{h \in \mathcal{H}} \max_{D \in \mathcal{D}} \mathcal{R}_D(h). \tag{2}$$

We say a solution h is *proper* if it is in class, i.e., $h \in \mathcal{H}$, and *randomized* if h is a probability distribution supported on the class, i.e., $h \in \Delta(\mathcal{H})$. In the latter case, we define the expected loss for a randomized hypothesis as $\mathcal{R}_D(h) \coloneqq \mathbb{E}_{f \sim h}[\mathcal{R}_D(f)]$.

Multi-distribution learning with smooth convex losses and collaborative learning seem to differ significantly in terms of their formally definition. However, we can reduce any collaborative learning problem to multi-distribution learning with smooth convex loss functions—as long as we allow for improper or randomized solutions to our collaborative learning problem. Allowing for improper or randomized solutions is not unreasonable and is in fact necessary to achieve non-trivial sample complexities in collaborative learning [9].

The first step to reducing collaborative learning to multi-distribution learning is to relax the optimization problem on the hypothesis class $\mathcal{H}$ onto the class of randomized hypotheses $\Delta(\mathcal{H})$.

Fact 2.1. *Consider a collaborative learning problem $(\mathcal{H}, \mathcal{D})$. Define the relaxed loss function $\widetilde{\ell} : \Delta(\mathcal{H}) \to [0,1]$ as $\widetilde{\ell}(h, z) = \mathbb{E}_{f \sim h}[\ell(f, z)]$. The induced losses in the multi-distribution learning problem, $(\mathcal{D}, \left\{\widetilde{\ell}\right\}, \Delta(\mathcal{H}))$, are smooth and convex, and any ε-optimal solution $h \in \Delta(\mathcal{H})$ is also an ε-optimal randomized solution to the collaborative learning problem $(\mathcal{H}, \mathcal{D})$; i.e., it satisfies Equation 2.*

This fact implies that multi-distribution learning can solve for deterministic but improper solutions to collaborative learning problems. This is because we can always extract a deterministic solution from a non-deterministic solution $h \in \Delta(\mathcal{H})$ by taking a majority vote, where we denote the majority vote hypothesis as h_{Maj}. The expected loss guarantee of this deterministic hypothesis is approximately bounded by that of the randomized h. We state this formally below for the setting where $\mathcal{H}$ is a set of binary classifiers; that is, where the label space is binary, $\mathcal{Y} = \{0,1\}$, and the loss function ℓ can be written as $\ell(h, (x, y)) = g(h(x), y)$ for some choice of $g : \mathcal{Y}^2 \to [0,1]$.

Fact 2.2. *Consider a collaborative learning problem $(\mathcal{H}, \mathcal{D})$ on a set of binary classifiers. For any randomized solution $h \in \Delta(\mathcal{H})$, define the deterministic hypothesis h_{Maj} as $h_{Maj}(x) = 1[\Pr_{f \sim h}(f(x) = 1) > \frac{1}{2}]$. The expected loss of h^{Maj} is bounded by $\max_{D \in \mathcal{D}} \mathcal{R}_D(h_{Maj}) \leq 2 \max_{D \in \mathcal{D}} \mathcal{R}_D(h)$.*

Group distributionally robust optimization. In the closely related setting of *group distributionally robust optimization (group DRO)* of Sagawa et al. [50], the goal is similarly to learn some hypothesis that guarantees small risk for every data distribution in a collection of distributions. In group DRO, the various data distributions are usually interpreted to either represent heterogeneous user populations and protected groups (for algorithmic fairness applications) or potential domains in which a model may be deployed (for robustness applications).

In contrast to the collaborative learning problem, group DRO problems are typically studied in a convex optimization setting where the hypothesis class is parameterized by some convex set and the loss function is smooth and convex. That is, the usual definition of the group DRO problem setting coincides with the definition of multi-distribution learning with *a single* smooth convex loss, i.e., $|\mathcal{L}| = 1$. Unlike in collaborative learning where we are interested in potentially improper or randomized solutions, the goal of group DRO is to learn a proper model $h_\theta \in \mathcal{H}$ where

$$\max_{D \in \mathcal{D}} \mathcal{R}_D(h_\theta) \leq \mathrm{OPT} + \varepsilon, \text{ where } \mathrm{OPT} \coloneqq \min_{\theta^* \in \Theta} \max_{D \in \mathcal{D}} \mathcal{R}_D(h_{\theta^*}). \tag{3}$$

It is the convexity of the group DRO problem setting that allows for the efficient learning of proper solutions and avoid relaxation to randomized solutions.

Agnostic federated learning. The agnostic federated learning framework of Mohri et al. [39] also coincides with multi-distribution learning with a single loss function. Like group distributionally robust optimization, agnostic federated learning is usually studied in a convex optimization setting with convex parameter spaces and smooth convex losses. As the general agnostic federated learning setting does not differ from group distributionally robust optimization in its formulation, we provide an identical treatment of both settings in Section 6.

2.3 Technical Background

We will use tools and definitions from the literature on zero-sum games and no-regret learning throughout the paper. This section provides a brief overview of these concepts.

Zero-sum games. A two-player zero-sum game is described by the tuple $(\mathcal{A}_-, \mathcal{A}_+, \phi)$ where $\mathcal{A}_-, \mathcal{A}_+$ are convex compact action sets and $\phi : \mathcal{A}_- \times \mathcal{A}_+ \to [0,1]$ is the game payoff. The player who chooses from $\mathcal{A}_-$ is called the minimizing player and tries to minimize the game payoff ϕ, while the player who chooses from $\mathcal{A}_+$ is called the maximizing player. A pair of actions (p, q) is called an *ε-min-max* equilibrium if neither player can unilaterally improve their objective by more than ε; that is, $\phi(p,q) - \min_{p^* \in \mathcal{A}_-} \phi(p^*, q) \leq \varepsilon$ and $\max_{q^* \in \mathcal{A}_+} \phi(p, q^*) - \phi(p,q) \leq \varepsilon$. If ϕ is convex-concave—i.e., $\phi(\cdot, q)$ is convex for every $q \in \mathcal{A}_+$ and $\phi(p, \cdot)$ is concave for every $p \in \mathcal{A}_-$—then an ε-min-max equilibrium always exists for every $\varepsilon \geq 0$. In the next section, we will describe methods that find ε-min-max equilibria by playing online learning algorithms against each other, a technique known as *no-regret game dynamics* [21].

No-regret learning. A no-regret (or *online*) learning algorithm $\mathcal{Q}_A$ maps from a sequence of costs $c^{(1:t-1)}$ to an action $a^{(t)} \in A$, where $a^{(t)} = \mathcal{Q}_A(c^{(1:t-1)})$. Notationally, we use the subscript A when writing an online learning algorithm $\mathcal{Q}_A$ to denote the action set that the algorithm $\mathcal{Q}_A$ is defined to act on. Regret is defined for a sequence of actions $a^{(1)}, \ldots, a^{(T)} \in A$ and costs $c^{(1)}, \ldots, c^{(T)} : A \to [0,1]$ as follows:

$$\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) \coloneqq \sum_{t=1}^{T} c^{(t)}(a^{(t)}) - \min_{a^* \in A} \sum_{t=1}^{T} c^{(t)}(a^*).$$

We say that a no-regret learning algorithm $\mathcal{Q}_A$ has a regret guarantee of $\gamma_T(\mathcal{Q}_A)$ if, for any sequence of linear cost functions $c^{(1:T)}$ of bounded norm, i.e., $\max_{t \in [T]} \left\| c^{(t)} \right\| \leq 1$, the algorithm $\mathcal{Q}_A$ chooses an action sequence $a^{(1:T)}$ with the regret bound $\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) \leq \sqrt{\gamma_T(\mathcal{Q}_A) T}$.

Examples of no-regret algorithms on probability simplexes. A well-studied online learning setting is that in which the action set is a probability simplex,$A = \Delta_n$, and all costs are linear functions of bounded norm. In this setting, we can interpret online learning algorithms as choosing mixed strategies $a^{(t)} \in \Delta_n$ over a set of *meta-actions*, $\{1, \ldots, n\}$, and the adversary as assigning a cost $\left\{c^{(t)}(e_1), \ldots, c^{(t)}(e_n)\right\}$ to each meta-action, so that the algorithm incurs the cost $\mathbb{E}_{i \sim a^{(t)}}\left[c^{(t)}(e_i)\right]$. An example of a no-regret algorithm in this setting is Exponential Gradient Descent (Hedge), defined as

$$\mathrm{Hedge}_A(c^{(1:t-1)}) \coloneqq \widetilde{a}^{(t)} / \left\| \widetilde{a}^{(t)} \right\|_1 \text{ where } \widetilde{a} \in \mathbb{R}^n \text{ and } \widetilde{a}_i \coloneqq \exp\left(-\eta \sum_{\tau=1}^{t-1} c^{(\tau)}(e_i) \right), \tag{4}$$

where $\eta \in (0,1)$ is a learning rate. The following lemma states a classical result for exponential gradient descent (Hedge), showing a regret guarantee of $O(\log(n))$.

Lemma 2.1 ([55])**.** *Let $c^{(1:T)}$ be any linear cost sequence where $\max_{t \in [T]} \left\| c^{(t)} \right\|_\infty \leq 1$ and $A = \Delta_n$. When $\eta = \sqrt{\log(n)/T}$, the actions $a^{(1:T)}$ chosen by Hedge satisfy $\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) \leq 2\sqrt{\log(n)/T}$.*

There also exist *partial feedback* no-regret algorithms—also known as semi-bandit algorithms—that only need to observe the cost functions at each timestep for a few meta-actions (i.e., along a few basis vectors). We can formalize these partial feedback (semi-bandit) algorithms as returning not only an action $a^{(t)} \in \Delta_n$ at each timestep t but also returning the meta-actions $I^{(t)} \subseteq [n]$ whose costs it will observe. We can therefore, somewhat unconventionally, write these algorithms as a mapping

$$\{c^{(1)}(e_i)\}_{i \in I^{(1)}}, \ldots, \{c^{(t-1)}(e_i)\}_{i \in I^{(t-1)}} \mapsto a^{(t)}, I^{(t)}.$$

The well-known partial feedback algorithm Exp3 chooses $a^{(t)} = \mathrm{Hedge}(\widetilde{c}^{(1:t-1)})$ and $I^{(t)} = \{i^{(t)}\}$ at each timestep, where $i^{(t)} \overset{\text{i.i.d.}}{\sim} a^{(t)}$ and $\widetilde{c}^{(t)}(a) = a_{i^{(t)}} c^{(t)}(e_{i^{(t)}}) / (a^{(t)}_{i^{(t)}} + \lambda)$ and where $\lambda \geq 0$ is a stepsize [41]. An alternatie partial feedback algorithm is ELP which, when given a partition P of the meta-actions $[n]$ into k subsets, guarantees $I^{(t)} \in P$ at each timestep. That is, it fixes a grouping of the meta-actions a priori and at each timestep only observes the costs of meta-actions belonging to a particular group.

Lemma 2.2 ([35]). *Let $c^{(1:T)}$ be arbitrary linear costs where $\max_{t\in[T]} \left\|c^{(t)}\right\|_\infty \leq 1$ and $A = \Delta_n$. For any $\delta \in (0,1)$ and partition P of $[n]$, the actions $a^{(1:T)}$ chosen by ELP satisfy $\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) \leq 2\sqrt{|P|\log(n/\delta)/T}$ with probability $1-\delta$. Moreover, only cost components from one element of P are observed per timestep: $\left|I^{(t)}\right| \in P$.*

We emphasize that the results in this manuscript are stated to accommodate general choices of online learning algorithms, with different guarantees and tradeoffs arising depending on which specific online learning algorithms one employs.

3 Overview of Our Approach

In this section, we provide an overview of our general approach for studying the sample complexity of multi-distribution learning. Our approach consists of two steps: (1) reducing multi-distribution learning to the problem of finding the equilibrium of a convex-concave zero-sum game, and (2) implementing game dynamics to efficiently find an equilibrium using only stochastic feedback.

3.1 Multi-Distribution Learning as a Zero-Sum Game

The multi-distribution learning problem corresponds to a zero-sum game with a minimizing player having action set $\mathcal{H}$, a maximizing player having action set $\mathcal{D}\times\mathcal{L}$, and a payoff function $\phi(h,(D,\ell)) = \mathcal{R}_{D,\ell}(h)$. Intuitively, the minimizing player can be interpreted as a learner who proposes candidate solutions while the maximizing player can be interpreted as an auditor who tries to pick a data distribution and loss function for which the learner's hypothesis performs poorly. It is not hard to see that any ε-min-max equilibrium (h, D) of this game corresponds to a 2ε-optimal solution.

Fact 3.1. *Given a multi-distribution learning problem, $(\mathcal{D},\mathcal{L},\mathcal{H})$, define the zero-sum game $(\mathcal{A}_-,\mathcal{A}_+,\phi)$ where $\mathcal{A}_- = \mathcal{H}$, $\mathcal{A}_+ = \mathcal{D}\times\mathcal{L}$, and $\phi(p,q) = \mathcal{R}_q(p)$. In any ε-min-max equilibrium (p,q), p is a 2ε-optimal solution.*

Proof. If (p,q) is an ε-min-max equilibria, the following holds by definition

$$\mathcal{R}_q(p) \leq \min_{h^*\in\mathcal{H}} \mathcal{R}_q(h^*) + \varepsilon \text{ and } \mathcal{R}_q(p) \geq \max_{D^*\in\mathcal{D},\ell^*\in\mathcal{L}} \mathcal{R}_{D^*,\ell^*}(p) - \varepsilon.$$

Rearranging gives $\max_{D^*\in\mathcal{D},\ell^*\in\mathcal{L}} \mathcal{R}_{D^*,\ell^*}(p) \leq \min_{h^*\in\mathcal{H}} \mathcal{R}_q(h^*) + 2\varepsilon \leq \mathrm{OPT} + 2\varepsilon$. □

A multi-distribution learning problem $(\mathcal{D},\mathcal{L},\mathcal{H})$ with *convex losses* can similarly be written as a convex-concave zero-sum game where a minimizing player chooses from the actions Θ, a maximizing player chooses from the actions $\mathcal{D}\times\mathcal{L}$, and the payoff function is defined as $\phi(p,q) = \mathcal{R}_q(h_p)$. As we previously noted, as the payoff function is convex-concave, a min-max equilibrium of this game must exist.

Many tools have been developed for efficiently finding the min-max equilibria of convex-concave zero-sum games. The connection between multi-distribution learning and zero-sum games allows us to draw on these tools to derive efficient learning algorithms.

Unknown payoff functions. The main challenge we will encounter is that of efficiently estimating the payoff function of the multi-distribution learning game, given that evaluating the function $\phi(p,q) = \mathcal{R}_q(h_p)$ requires computing expectations for an unknown data distribution. Typically, to compute the min-max equilibrium of a convex-concave game, one needs a first-order approximation for the payoff function ϕ at various strategy profiles—that is, we require the gradients $\nabla_p\phi(p,q)$ and $\nabla_q\phi(p,q)$ for various choices of actions $p\in\mathcal{A}_-$ and $q\in\mathcal{A}_+$. We will achieve this by designing *noisy first-order oracles* that, when queried with a strategy profile (p,q), return unbiased estimates of the gradient $\nabla_p\phi(p,q)$ or $\nabla_q\phi(p,q)$. To control the variance of these oracles, we will also ask that their estimates be bounded in norm, as we will formalize in the sequel. Behind the scenes, we will implement these first-order approximations by querying example oracles.

A complication to implementing these noisy first-order oracles efficiently is that payoff estimation is more costly for the maximizing player than the minimizing player. Indeed, consider a strategy profile (p,q) in the multi-distribution learning game. Obtaining an unbiased bounded estimate of the minimizing player

(learner)'s payoff gradient requires only drawing a single datapoint from the mixture distribution specified by the other player (the auditor), since the learner only needs a counterfactual estimate of how well each hypothesis would have performed on the mixture. However, obtaining an unbiased bounded estimate of the maximizing player (the auditor)'s payoff gradient requires drawing n datapoints, since the auditor needs a counterfactual estimate of how well the minimizing player's hypothesis would have performed on each potential data distribution $D_1, \ldots, D_n$. This intuitive arugment is formalized as follows.

Fact 3.2. *Consider a multi-distribution learning problem $(\mathcal{D}, \mathcal{L}, \mathcal{H})$ with 1-smooth losses and a strategy profile $h_\theta \in \mathcal{H}$ and $q \in \Delta(\mathcal{D} \times \mathcal{L})$. The gradient vector $\nabla_\theta \ell(h_\theta, z)$, where $z \overset{\text{i.i.d.}}{\sim} D$ and $(D, \ell) \overset{\text{i.i.d.}}{\sim} q$, is an unbiased bounded estimate of the first-order information $\nabla_\theta \mathcal{R}_q(h_\theta)$; i.e., $\mathbb{E}_{z \sim D, (D,\ell) \sim q} [\nabla_\theta \ell(h_\theta, z)] = \nabla_\theta \mathcal{R}_q(h_\theta)$ and $\|\nabla_\theta \ell(h_\theta, z)\| \leq 1$. Similarly, the vector $[1 - \ell_j(h_\theta, z_i)]_{i \in [n], j \in [m]}$ where $z_i \overset{\text{i.i.d.}}{\sim} D_i$ is an unbiased bounded estimate of the first-order information vector $1 - \nabla_q \mathcal{R}_q(h_\theta)$.*

3.2 Equilibrium Computation in Stochastic Convex-Concave Games

Algorithm 1 Finding Equilibria in Convex-Concave Games with Asymmetric Costs.

Input: Action sets $\mathcal{A}_-, \mathcal{A}_+$, steps T, first-order oracles g_-, g_+, and online learning algorithms $\mathcal{Q}_{\mathcal{A}_-}, \mathcal{Q}_{\mathcal{A}_+}$;
for $t = 1, 2, \ldots, T$ **do**
  Let $p^{(t)} = \mathcal{Q}_{\mathcal{A}_-} \left(\left\{ p \mapsto \left\langle g_-(p^{(\tau)}, q^{(\tau)}), p \right\rangle \right\}_{\tau \in [t-1]} \right)$;
  Let $q^{(t)} = \mathcal{Q}_{\mathcal{A}_+} \left(\left\{ q \mapsto \left\langle g_+(p^{(\tau)}, q^{(\tau)}), q \right\rangle \right\}_{\tau \in [t-1]} \right)$;
end for
Return $\overline{p} = \frac{1}{T} \sum_{t=1}^{T} p^{(t)}$ and $\overline{q} = \frac{1}{T} \sum_{t=1}^{T} q^{(t)}$;

We now describe an online learning framework for finding equilibria in stochastic games using game dynamics. We will later see this framework, which is described by Algorithm 1, easily accommodates the asymmetric costs of estimating each player's payoff gradients. Lemma 3.1 outlines a guarantee of returning an approximate min-max equilibrium with high probability. We note that the guarantee of Lemma 3.1 is stated in terms of the regret bounds $\gamma_T(\mathcal{Q}_{\mathcal{A}_-})$ and $\gamma_T(\mathcal{Q}_{\mathcal{A}_+})$ of the online learning algorithms that we plug into Algorithm 1. This means that choosing different online learning algorithms to be $\mathcal{Q}_{\mathcal{A}_-}$ and $\mathcal{Q}_{\mathcal{A}_+}$ will yield different guarantees.

Lemma 3.1. *Consider a convex-concave zero-sum game $(\mathcal{A}_-, \mathcal{A}_+, \phi)$ with an L-smooth payoff ϕ. Assume that*

1. *g_- is a noisy first-order oracle that returns unbiased, bounded, and independent estimates of $\nabla_p \phi(p, q)$. That is, for all $p \in \mathcal{A}_-$ and $q \in \mathcal{A}_+$, we have $\|g_-(p, q)\| \leq L$ with probability one and $\mathbb{E}[g_-(p, q)] = \nabla_p \phi(p, q)$.*
2. *g_+ is a noisy first-order oracle that returns unbiased, bounded, and independent estimates of $-\nabla_q \phi(p, q)$.*
3. *The action sets $\mathcal{A}_-$ and $\mathcal{A}_+$ have a diameters of at most R in the dual norm $\|\cdot\|_*$, i.e. $\max_{p, p' \in \mathcal{A}_-} \|p - p'\|_* \leq R$ and $\max_{q, q' \in \mathcal{A}_+} \|q - q'\|_* \leq R$.*

Then Algorithm 1 returns an ε-min-max equilibrium with probability $1 - \delta$ if

$$T \geq \frac{4L^2}{\varepsilon^2} \left(32R^2 \log(2/\delta) + 25\gamma_T(\mathcal{Q}_{\mathcal{A}_-}) + 25\gamma_T(\mathcal{Q}_{\mathcal{A}_+}) \right). \tag{5}$$

Informally, we can interpret the regret bound $\gamma_T(\mathcal{Q}_{\mathcal{A}_-})$ as the difficulty of making rational choices for the minimizing player and $\gamma_T(\mathcal{Q}_{\mathcal{A}_+})$ as the difficulty of making rational choices for the maximizing player. This lemma then says that the number of iterations required to find an equilibrium depends on the *additive combination* of the complexity seen by each player.

Before we proceed to a proof of this lemma, we recall some standard results on game dynamics and online learning. First, we recall that no-regret dynamics efficiently learns equilibria in convex-concave games [21]. This fact implies that, in order to learn the equilibria of our multi-distribution learning game, it suffices to design suitable no-regret algorithms.

Fact 3.3. *Let $(\mathcal{A}_-, \mathcal{A}_+, \phi)$ be a convex-concave zero-sum game. For any actions $p^{(1:T)} \in \mathcal{A}_-$ and $q^{(1:T)} \in \mathcal{A}_+$ with regret $\mathrm{Reg}\left(p^{(1:T)}, \left\{\phi(\cdot, q^{(t)})\right\}_{t\in[T]}\right) \le T\varepsilon$ and $\mathrm{Reg}\left(q^{(1:T)}, \left\{-\phi(p^{(t)}, \cdot)\right\}_{t\in[T]}\right) \le T\varepsilon$, the average actions $\frac{1}{T}\sum_{t=1}^T p^{(t)}$ and $\frac{1}{T}\sum_{t=1}^T q^{(t)}$ form a 2ε-min-max equilibrium.*

Proof. By convexity, $\overline{p} \coloneqq \frac{1}{T}\sum_{t=1}^T p^{(t)} \in \mathcal{A}_-$ and $\overline{q} \coloneqq \frac{1}{T}\sum_{t=1}^T q^{(t)} \in \mathcal{A}_+$. Since ϕ is concave in its second argument, we can apply Jensen's inequality to the regret bound of the minimizing player to get

$$\mathrm{Reg}\left(\{\overline{p}\}, \{\phi(\cdot, \overline{q})\}\right) = \phi(\overline{p}, \overline{q}) - \min_{p^*\in\mathcal{A}_-} \phi(p^*, \overline{q}) \le \frac{1}{T}\sum_{t=1}^T \phi(\overline{p}, q^{(t)}) - \min_{p^*\in\mathcal{A}_-} \phi(p^*, \overline{q}) \le \varepsilon.$$

Since ϕ is convex in its first argument, we can again apply Jensen's inequality, this time to the regret bound of the maximizing player, to get

$$\mathrm{Reg}\left(\{\overline{q}\}, \{-\phi(\overline{p}, \cdot)\}\right) = \max_{q^*\in\mathcal{A}_+} \phi(\overline{p}, q^*) - \phi(\overline{p}, \overline{q}) \le \max_{q^*\in\mathcal{A}_+} \phi(\overline{p}, q^*) - \frac{1}{T}\sum_{t=1}^T \phi(p^{(t)}, \overline{q}) \le \varepsilon.$$

Summing these inequalities yields that $\max_{q^*\in\mathcal{A}_+} \phi(\overline{p}, q^*) - \min_{p^*\in\mathcal{A}_-} \phi(p^*, \overline{q}) \le 2\varepsilon$. □

Next, we recall that, in a no-regret learning problem with linear costs $c^{(1:T)}$, a player can run any online learning algorithm directly on independent, unbiased, bounded estimates $\widehat{c}^{(1:T)}$ of its costs $c^{(1:T)}$ and expect only a constant factor increase in its worst-case regret bound. This fact, which is classical in both optimization theory [40, 28] and online learning theory [21], follows by a standard martingale argument. That no-regret learning algorithms generalize well on stochastic costs will mean that we can efficiently implement no-regret dynamics on stochastic games using noisy payoff observations that need only be unbiased and bounded. Importantly, this means we do not need to obtain ε-accurate estimates of each players' payoff at each iteration, which would make no-regret dynamics prohibitively expensive in terms of sample complexity.

In the sequel, given a linear cost function $c : A \to \mathbb{R}$, we will abuse notation and use c to also denote the vector such that $c(a) = \langle c, a\rangle$ for all $a \in A$. We will also use $\|c\|$ to denote the norm of the vector c.

Fact 3.4. *Let $\widehat{c}^{(1:T)}$ be independent, unbiased estimates of a set of linear costs $c^{(1:T)}$, where $\left\|c^{(t)}\right\| \le L$ and $\left\|\widehat{c}^{(t)}\right\| \le L$ at all steps $t \in [T]$. Assume an action diameter of R, i.e. $\max_{a,a'\in A} \|a - a'\|_* \le R$. The actions $a^{(t)} = \mathcal{Q}_A(\widehat{c}^{(1:t-1)})$ chosen by applying an online learning algorithm $\mathcal{Q}_A$ to the estimated costs $\widehat{c}^{(1:T)}$ satisfies the following generalization bound with probability $1 - \delta$:*

$$\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) - \mathrm{Reg}(a^{(1:T)}, \widehat{c}^{(1:T)}) \le 4L\sqrt{T}\left(R\sqrt{2\log(1/\delta)} + \sqrt{\gamma_T(\mathcal{Q}_A)}\right). \tag{6}$$

Proof. We first upper bound the generalization error by the regret of actions $a^{(1:T)}$ on the cost differences $\left\{c^{(t)} - \widetilde{c}^{(t)}\right\}$. Since $\max_{a^*\in A}\sum_{t=1}^T \left\langle c^{(t)}, a^*\right\rangle - \max_{a^*\in A}\sum_{t=1}^T \left\langle \widehat{c}^{(t)}, a^*\right\rangle \le \max_{a^*\in A}\sum_{t=1}^T \left\langle c^{(t)} - \widehat{c}^{(t)}, a^*\right\rangle$, we can bound generalization error Δ by

$$\Delta \coloneqq \mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) - \mathrm{Reg}(a^{(1:T)}, \widehat{c}^{(1:T)}) \le \mathrm{Reg}(a^{(1:T)}, c^{(1:T)} - \widehat{c}^{(1:T)}).$$

The remainder of the proof is dedicated to bounding this regret term, which we can write explicitly as

$$\mathrm{Reg}(a^{(1:T)}, c^{(1:T)} - \widehat{c}^{(1:T)}) = \max_{a^*\in A}\sum_{t=1}^T \left\langle a^{(t)} - a^*, c^{(t)} - \widehat{c}^{(t)}\right\rangle.$$

We will ultimately control this regret term with a martingale argument, appealing to the fact that at each timestep the noisy costs we observe are unbiased even conditioned on previous cost observations. However, we first need to control the variational term a^*, which we will do with a standard approach of introducing a shadow term $\varepsilon^{(t)} = \mathcal{Q}_A(\{c^{(\tau)} - \widehat{c}^{(\tau)}\}_{\tau\in[t-1]})$. That is, $\varepsilon^{(1:T)}$ is the result of (hypothetically) running the online learning algorithm $\mathcal{Q}_A$ on the cost sequences $\{c^{(\tau)} - \widehat{c}^{(\tau)}\}$. Adding and subtracting the shadow terms $\varepsilon^{(1:T)}$ from the inner product,

$$\max_{a^*\in A}\sum_{t=1}^T \left\langle a^{(t)} - a^*, c^{(t)} - \widehat{c}^{(t)}\right\rangle = \max_{a^*\in A}\sum_{t=1}^T \left\langle a^{(t)} - \varepsilon^{(t)} + \varepsilon^{(t)} - a^*, c^{(t)} - \widehat{c}^{(t)}\right\rangle$$

$$= \sum_{t=1}^{T} \left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle + \max_{a^* \in A} \sum_{t=1}^{T} \left\langle \varepsilon^{(t)} - a^*, c^{(t)} - \widehat{c}^{(t)} \right\rangle.$$

Since we constructed $\varepsilon^{(1:T)}$ with our online learning algorithm $\mathcal{Q}_A$, we obtain a regret guarantee for the action sequence $\varepsilon^{(1:T)}$ on the cost sequence $\left\{c^{(t)} - \widehat{c}^{(t)}\right\}$, which yields

$$\max_{a^* \in A} \sum_{t=1}^{T} \left\langle \varepsilon^{(t)} - a^*, c^{(t)} - \widehat{c}^{(t)} \right\rangle \in 4L\sqrt{\gamma_T(\mathcal{Q}_A)T}. \tag{7}$$

The term $4L$ appears in this bound since $c^{(t)}(a) - \widehat{c}^{(t)}(a) \in [-2L, 2L]$ must be normalized to $[0, 1]$.

It remains to bound $\sum_{t=1}^{T} \left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle$. This expression is a martingale because, for each summand $\left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle$, the left-hand side $a^{(t)} - \varepsilon^{(t)}$ is conditionally (on previous summands) independent of $c^{(t)} - \widehat{c}^{(t)}$. Formally, we define the filtration $\{\mathcal{F}^{(t)}\}_{t=0}^{T}$ as the sigma algebra generated by $\{\widehat{c}^{(t)}\}_{t=1}^{T}$. By construction, we know that $(a^{(t)} - \varepsilon^{(t)})$ is $\mathcal{F}^{(t-1)}$-measurable and thus $\left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle$ is $\mathcal{F}^{(t)}$-measurable. Since $\widehat{c}^{(t)}$ is unbiased, we also have that $\mathbb{E}\left[\left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle \middle| \mathcal{F}^{(t-1)}\right] = 0$. Finally, we can observe that the difference sequence of our martingale can be bounded with Holder's inequality as

$$\left| \left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle \right| \leq \left\| a^{(t)} - \varepsilon^{(t)} \right\|_* \left\| c^{(t)} - \widehat{c}^{(t)} \right\| \leq 4RL.$$

By the Azuma-Hoeffding inequality, we can thus bound, for any $\varepsilon > 0$,

$$\Pr\left(\sum_{t=1}^{T} \left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle \geq \varepsilon \right) \leq \exp\left(-\frac{\varepsilon^2}{32TR^2L^2} \right).$$

We can rewrite this as saying, with probability $1 - \delta$, that $\sum_{t=1}^{T} \left\langle a^{(t)} - \varepsilon^{(t)}, c^{(t)} - \widehat{c}^{(t)} \right\rangle \leq 4RL\sqrt{2T\log(1/\delta)}$. In combination with Equation 7, this inequality yields the desired bound on Δ. □

We intend to apply online learning algorithms to our convex-concave games, where the payoff function is not necessarily linear. To overcome the fact that the concentration result in Fact 3.4 is specific to linear costs, we now turn to showing that one can linearize any online learning problem with convex costs. That is, we can reduce online learning on differentiable convex costs to online learning on linear costs, allowing us to apply Fact 3.4. Specifically, we will use the concept of variational error, which is usually defined as $\mathrm{VErr}(a^{(1:T)}, c^{(1:T)}) \coloneqq \mathrm{Reg}(a^{(1:T)}, \widetilde{c}^{(1:T)})$ where $\widetilde{c}^{(t)}(a) = \left\langle a, \nabla c^{(t)}(a^{(t)}) \right\rangle$. We now formalize the fact that variational error yields an upper bound on regret: $\mathrm{VErr}(a^{(1:T)}, c^{(1:T)}) \geq \mathrm{Reg}(a^{(1:T)}, c^{(1:T)})$.

Fact 3.5. *Let $c^{(1:T)} : A \to \mathbb{R}$ be convex functions on a convex compact domain A and let $\partial c^{(t)}(a^{(t)})$ be a partial subgradient of $c^{(t)}$ at $a^{(t)}$. For any sequence $a^{(1:T)} \in A$,*

$$\mathrm{Reg}(a^{(1:T)}, c^{(1:T)}) \coloneqq \sum_{t=1}^{T} c^{(t)}(a^{(t)}) - \min_{a^* \in A} \sum_{t=1}^{T} c^{(t)}(a^*) \leq \mathrm{VErr}(a^{(1:T)}, c^{(1:T)}) \coloneqq \max_{a^* \in A} \sum_{t=1}^{T} \left\langle \partial c^{(t)}(a^{(t)}), a^{(t)} - a^* \right\rangle.$$

Proof. By the convexity of ϕ, $\sum_{t=1}^{T} \left\langle \partial c^{(t)}(a^{(t)}), a^{(t)} - a^* \right\rangle \geq \sum_{t=1}^{T} c^{(t)}(a^{(t)}) - c^{(t)}(a^*)$ for any fixed $a^* \in A$. □

We now turn to proving Lemma 3.1.

Proof of Lemma 3.1. Let $\widehat{c}_{-}{}^{(\tau)}(p) = \left\langle g_{-}(p^{(\tau)}, q^{(\tau)}), p \right\rangle$ and $\widehat{c}_{+}{}^{(\tau)}(q) = \left\langle g_{+}(p^{(\tau)}, q^{(\tau)}), q \right\rangle$ denote the cost functions that the online learning algorithms $\mathcal{Q}_{\mathcal{A}_-}$ and $\mathcal{Q}_{\mathcal{A}_+}$ are given in Algorithm 1. We first recall that the bounds on the regret for $\mathcal{Q}_{\mathcal{A}_-}$ and $\mathcal{Q}_{\mathcal{A}_+}$ yield the empirical regret bounds

$$\mathrm{Reg}\left(p^{(1:T)}, \widehat{c}_{-}{}^{(1:T)}\right) \leq L\sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_-})T}, \quad \mathrm{Reg}\left(q^{(1:T)}, \widehat{c}_{+}{}^{(1:T)}\right) \leq L\sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_+})T},$$

as $\widehat{c}_{-}{}^{(1:T)}$ and $\widehat{c}_{+}{}^{(1:T)}$ are linear cost functions with a bounded norm of at most L. By Fact 3.4, with probability $1 - 2\delta$, we can bound the generalization error with the true cost functions $c_{-}{}^{(\tau)}(p) = \left\langle \nabla_{p^{(\tau)}} \phi(p^{(\tau)}, q^{(\tau)}), p \right\rangle$ and $c_{+}{}^{(\tau)}(q) = -\left\langle \nabla_{q^{(\tau)}} \phi(p^{(\tau)}, q^{(\tau)}), q \right\rangle$ by

$$\mathrm{Reg}\left(p^{(1:T)}, \widehat{c}_{-}{}^{(1:T)}\right) - \mathrm{Reg}\left(p^{(1:T)}, c_{-}{}^{(1:T)}\right) \leq 4L\sqrt{T}\left(R\sqrt{2\log(1/\delta)} + \sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_-})}\right),$$

$$\mathrm{Reg}\left(q^{(1:T)}, \widehat{c}_+{}^{(1:T)}\right) - \mathrm{Reg}\left(q^{(1:T)}, c_+{}^{(1:T)}\right) \le 4L\sqrt{T}\left(R\sqrt{2\log(1/\delta)} + \sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_+})}\right).$$

Next, we observe that the costs c_- and c_+ are constructed so that regret on these costs coincides with variational error on ϕ; i.e., $\mathrm{Reg}\left(p^{(1:T)}, c_-{}^{(1:T)}\right) = \mathrm{VErr}(p^{(1:T)}, \{\phi(\cdot, q^{(t)})\}_{t\in[T]})$ and $\mathrm{Reg}\left(q^{(1:T)}, c_+{}^{(1:T)}\right) = \mathrm{VErr}(q^{(1:T)}, \{-\phi(p^{(t)}, \cdot)\}_{t\in[T]})$. Our empirical regret and generalization error bounds therefore imply

$$\mathrm{VErr}(p^{(1:T)}, \{\phi(\cdot, q^{(t)})\}_{t\in[T]}) \le L\sqrt{T}\left(4R\sqrt{2\log(1/\delta)} + 5\sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_-})}\right)$$
$$\mathrm{VErr}(q^{(1:T)}, \{-\phi(p^{(t)}, \cdot)\}_{t\in[T]}) \le L\sqrt{T}\left(4R\sqrt{2\log(1/\delta)} + 5\sqrt{\gamma_T(\mathcal{Q}_{\mathcal{A}_+})}\right).$$

For the stated choice of T, $\mathrm{VErr}(p^{(1:T)}, \{\phi(\cdot, q^{(t)})\}_{t\in[T]}) \le T\varepsilon$ and $\mathrm{VErr}(q^{(1:T)}, \{-\phi(p^{(t)}, \cdot)\}_{t\in[T]}) \le T\varepsilon$ with probability at least $1-2\delta$. By Fact 3.5, $\mathrm{Reg}(p^{(1:T)}, \{\phi(\cdot, q^{(t)})\}_{t\in[T]}) \le T\varepsilon$ and $\mathrm{Reg}(q^{(1:T)}, \{-\phi(p^{(t)}, \cdot)\}_{t\in[T]}) \le T\varepsilon$. Finally, by Fact 3.3, we have that $(\overline{p}, \overline{q})$ is an 2ε-min-max equilibrium with probability at least $1-2\delta$. □

4 Multi-Distribution Learning

In this section, we present Algorithm 2, a general recipe for multi-distribution learning. Algorithm 2 is a general framework into which one can plug any choice of online learning algorithm to obtain a variety of multi-distribution learning guarantees. The algorithm, which implements a form of stochastic game dynamics, uses the tools we outlined in Section 3 to reduce multi-distribution learning to the problem of solving a convex-concave game, and we then employ online learning algorithms to solve the game. In Theorem 4.1, we present one example of the multi-distribution learning guarantees that Algorithm 3 can provide for any online-learnable hypothesis class $\mathcal{H}$.

Algorithm 2 General Recipe for Multi-Distribution Learning.

Input: Hypothesis class $\mathcal{H}$ with parameter space Θ, example oracles $\mathrm{EX}(D_1), \ldots, \mathrm{EX}(D_n)$, iterations T, online learning algorithm $\mathcal{Q}_\Theta$ and a partial feedback online learning algorithm $\mathcal{Q}_{\Delta_{n\times m}}$;
Initialize: $\theta^{(1)} = \mathcal{Q}_\Theta(\emptyset)$ and $w^{(1)}, I^{(1)} = \mathcal{Q}_{\Delta_{n\times m}}(\emptyset)$;
for $t = 2, \ldots, T$ **do**
  Sample $(i,j) \sim w^{(t)}$ and a datapoint $z^{(t-1)} \overset{\text{i.i.d.}}{\sim} \mathrm{EX}(D_i)$;
  Update the learner's action $\theta^{(t)} = \mathcal{Q}_\Theta(\{\theta \mapsto \langle \nabla_\theta \ell_j(h_\theta, z^{(\tau)}), \theta\rangle\}_{\tau\in[t-1]})$;
  For all $(i,j) \in I^{(t-1)}$, sample a datapoint $z_i{}^{(t-1)} \overset{\text{i.i.d.}}{\sim} \mathrm{EX}(D_i)$ for every unique i;
  Update the auditor's action $w^{(t)}, I^{(t)} = \mathcal{Q}_{\Delta_{n\times m}}(\{w \mapsto 1 - \sum_{i=1}^n \sum_{j=1}^m w_{ij}\ell_j(h_{\theta^{(\tau)}}, z_i{}^{(\tau)})\}_{\tau\in[t-1]})$;
end for
Return: $h_{\overline{\theta}}$ where $\overline{\theta} = \frac{1}{T}\sum_{t=1}^T \theta^{(t)}$;

Theorem 4.1. *Consider a multi-distribution learning problem $(\mathcal{D}, \mathcal{L}, \mathcal{H})$ with convex and 1-smooth losses and a parameter space Θ of diameter R: $\max_{\theta,\theta'\in\Theta} \|\theta - \theta'\|_* \le R$. Let $\mathcal{Q}_{\Delta_{n\times m}}$ be a high-probability [41] variant of the ELP algorithm [35] implemented on the partition $P = [\{(i,j)\}_{j\in m}]_{i\in n}$. For any choice of online learning algorithm $\mathcal{Q}_\Theta$, with probability $1-\delta$, Algorithm 2 returns an ε-optimal solution $h_{\overline{\theta}} \in \mathcal{H}$ where*

$$\varepsilon \in O\left(\sqrt{T^{-1}\left(\gamma_T(\mathcal{Q}_\Theta) + n\log(mn/\delta) + R\log(1/\delta)\right)}\right).$$

The sample complexity of the algorithm is $2T$.

Proof. Algorithm 2 implements Algorithm 1 on the convex-concave game $(\Theta, \Delta(\mathcal{D}\times\mathcal{L}), \phi)$, where the payoff function ϕ is 1-smooth and defined as $\phi(\theta, (D,\ell)) = \mathcal{R}_{D,\ell}(h_\theta)$.

We now turn to verifying that the conditions of Lemma 3.1 are satisfied. Since we assume that all losses are 1-smooth in some norm $\|\cdot\|$, the learner's payoff gradient $\nabla_p\phi(p,q)$ is always bounded by 1 in the same norm, while we assume the the learner's action set diameter is at most R as measured by the dual norm $\|\cdot\|_*$. By linearity of expectation, we also have that the auditor's payoff gradient $\nabla_q\phi(p,q)$ is always bounded by 1

in the infinity norm, while the auditor's action set diameter—a probability simplex—is at most 1 as measured by the 1-norm. By Fact 3.2, the gradient estimators used in Algorithm 1, i.e. $\theta \mapsto \left\langle \nabla_\theta \ell_j(h_\theta, z^{(\tau)}), \theta \right\rangle$ and $w \mapsto 1 - \sum_{i=1}^n \sum_{j=1}^m w_{ij} \ell_j(h_{\theta^{(\tau)}}, z_i{}^{(\tau)})$, are unbiased, i.i.d., and 1-bounded estimates of the payoff gradients $\nabla_\theta \phi(\theta, w^{(\tau)})$ and $\nabla_w \phi(\theta^{(\tau)}, w)$ respectively. Thus, all conditions of Lemma 3.1 are satisfied.

We therefore know that $(h_{\overline{\theta}}, \frac{1}{T}\sum_{t=1}^T w^{(t)})$ is an ε-equilibrium with probability $1-\delta$ if

$$T \geq \frac{128}{\varepsilon^2}\left(R^2 \log(2/\delta) + \gamma_T(\mathcal{Q}_\Theta) + \gamma_T(\mathcal{Q}_{\Delta_{nm}})\right).$$

Recalling that the regret bound of the ELP algorithm is $\gamma_T(\mathcal{Q}'_{\Delta_{nm}}) \in O(n \log(nm/\delta))$ (Lemma 2.2), it suffices if $T \geq \frac{C}{\varepsilon^2}\left(R^2 \log(2/\delta) + \gamma_T(\mathcal{Q}_\Theta) + n\log(mn/\delta))\right)$ for some universal constant C. By Fact 3.1, it thus follows that $\overline{h} \coloneqq \frac{1}{T}\sum_{t=1}^T h^{(t)}$ is a 2ε-optimal solution with probability $1-2\delta$.

We now resolve the sample complexity of our instantiation of Algorithm 2. At every timestep, the learner draws one datapoint $z^{(t)}$. The number of datapoints that the auditor draws in any given iteration is the number of unique values of i in the set $\left\{(i,j) \in I^{(t-1)}\right\}$. Concretely, recall that $I^{(t-1)}$ denotes which cost components that the partial feedback algorithm $\mathcal{Q}_{\Delta_{n\times m}}$ chooses to observe from step $t-1$, where an entry $(i,j) \in I^{(t-1)}$ indicates that the auditor wishes to estimate the (in hindsight) outcome of auditing the learner on the data distribution D_i and loss function ℓ_j. We implement the ELP algorithm on the partition $P = [\{(1,1),\ldots,(1,m)\},\ldots,\{(n,1),\ldots,(n,m)\}]$, where all elements of a given group $I \in P$ correspond to the same data distribution D_i but different choices of loss functions ℓ_j, meaning that $\text{Unique}(\{i \mid (i,j) \in I\}) = 1$. Since ELP guarantees that $I^{(t-1)} \in P$, we can conclude the auditor only observes one datapoint per iteration. The total sample complexity of the algorithm is therefore $2T$. □

One interpretation of Theorem 4.1 is that the worst-case sample complexity of multi-distribution learning is not significantly larger than the worst-case sample complexity of learning a single data distribution with an online-to-batch reduction. More specifically, handling multiple data distributions and loss functions only adds an additive factor to one's sample complexity. It may seem surprising that our sample complexity bound for multi-distribution learning—a stochastic setting—is characterized by complexity of online decision-making—an adversarial setting. However, multi-distribution learning is inherently an online decision-making problem, as it requires one to strategize adaptively regarding the choice of data distribution to collect additional samples from. This is in contrast to the usual single-distribution learning setting, where there is no explicit decision-making involved.

5 Collaborative Learning

In this section, we present our main result on collaborative learning: a tight bound on the sample complexity of collaborative learning in agnostic settings. In particular, we show that the collaborative learning of a finite hypothesis class $\mathcal{H}$ on n data distributions requires $\Theta(\frac{\log(|\mathcal{H}|)+n\log(n/\delta)}{\varepsilon^2})$ samples. This means that, when characterizing hypothesis class complexity by $\log(|\mathcal{H}|)$, the worst-case sample complexity of learning n distributions is not significantly larger than the worst-case sample complexity of learning one data distribution. Specifically, it requires at most a constant factor or an additive $O(n\log(n/\delta)/\varepsilon^2)$ factor additional samples.

5.1 Sample Complexity Upper Bound

Theorem 5.1 states our sample complexity upper bound for agnostic collaborative learning. It is a direct implication of the sample complexity of multi-distribution learning because, as we noted previously (Fact 2.1), one can easily reduce agnostic collaborative learning to multi-distribution learning. Theorem 5.1 improves over the best-known sample complexity for agnostic collaborative learning by Nguyen and Zakynthinou [42] in two ways, giving an $\text{OPT} + \varepsilon$ bound for randomized classifiers instead of their $2\text{OPT} + \varepsilon$ bound, and improving their sample complexity of $O\left(\frac{1}{\varepsilon^5}\left(\log(n)\log(|\mathcal{H}|)\log\left(\frac{1}{\varepsilon}\right) + n\log\left(\frac{n}{\delta}\right)\right)\right)$ by a multiplicative factor of $\frac{1}{\varepsilon^3}\log(n)\log\left(\frac{1}{\varepsilon}\right)$.

Theorem 5.1. *Given a set of data distributions $\mathcal{D} = \{D_1,\ldots,D_n\}$, a hypothesis class $\mathcal{H} \in \mathcal{Y}^\mathcal{X}$, and a $[0,1]$-bounded loss ℓ, consider the collaborative learning problem $(\mathcal{H},\mathcal{D})$. Consider the output $h \in \Delta(\mathcal{H})$ of*

applying Theorem 4.1's algorithm to the multi-distribution learning problem $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$ *where the online learning algorithm* $\mathcal{Q}_{\Delta(\mathcal{H})}$ *is Hedge. With probability* $1-\delta$*,* h *is an* ε*-optimal solution (see* (2)*) to* $(\mathcal{H}, \mathcal{D})$ *and the sample complexity is* $O\left(\varepsilon^{-2}\left(\log(|\mathcal{H}|) + n\log(n/\delta)\right)\right)$*.*

Proof. The reduction of collaborative learning to multi-distribution learning (Fact 2.1) implies that any ε-optimal solution to the multi-distribution learning problem $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$ is an ε-optimal solution to the collaborative learning problem $(\mathcal{H}, \mathcal{D})$. Fact 2.1 also establishes that $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$ has convex and 1-smooth losses, where smoothness is measured in the infinity norm. Since $\Delta(\mathcal{H})$ is a probability simplex, we also have that its diameter is at most 2 in the 1-norm. The guarantees of Theorem 4.1 thus hold in our setting.

Since we choose to instantiate the online learning algorithm $\mathcal{Q}_{\Delta(\mathcal{H})}$ used in Algorithm 2 with Hedge, we recall that Hedge provides the regret guarantee (Lemma 2.1) of $\gamma_T(\mathcal{Q}_{\Delta(\mathcal{H})}) \in O(\log(|\mathcal{H}|))$. Thus, we can write the statement of Theorem 4.1 as guaranteeing that Algorithm 2 takes at most $2T$ datapoints in total and that, with probability $1-\delta$, the output h of Algorithm 2 is an ε-optimal solution to $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$, where $\varepsilon \in O\left(\sqrt{T^{-1}\left(\log(|\mathcal{H}|) + n\log(n/\delta)\right)}\right)$. □

For constants ε and δ, our sample complexity of $O\left(\log(|\mathcal{H}|) + n\log(n)\right)$ appears to violate the lower bound of $\Omega\left(\log(|\mathcal{H}|)\log(n) + n\log(\log(|\mathcal{H}|))\right)$ due to Chen, Zhang, and Zhou [13]. This discrepancy is due to a small error in the proof of that lower bound, which we have verified in private communications with the authors.

Recall that we can convert any randomized solution to a deterministic one by taking a majority vote (Fact 2.2). Our sample complexity bound on finding randomized solutions therefore also implies a sample complexity bound on finding deterministic improper solutions to collaborative learning problems.

Corollary 5.2 (Theorem 5.1 and Fact 2.2)**.** *Consider a collaborative learning problem* $(\mathcal{H}, \mathcal{D})$ *on a set of binary classifiers. There is an algorithm with a sample complexity of* $O\left(\varepsilon^{-2}\left(\log(|\mathcal{H}|) + n\log(n/\delta)\right)\right)$ *that, with probability* $1-\delta$*, returns a deterministic improper solution* $h_{Maj} \in \mathcal{Y}^{\mathcal{X}}$ *such that* $\max_{D\in\mathcal{D}} \mathcal{R}_D(h_{Maj}) \leq 2\mathrm{OPT} + \varepsilon$*.*

In the next subsection, we will show that this sample complexity upper bound is tight up to double-log factors and exactly tight in the regime where $n \in O(\log(|\mathcal{H}|)$.

5.2 Sample Complexity Lower Bound

We now provide matching lower bounds on the sample complexity of agnostic collaborative learning. We note that these lower bounds hold for any collaborative learning algorithm that returns ε-optimal solutions, regardless of whether those algorithms perform sampling on-demand and regardless of whether the algorithms return randomized or deterministic and proper or improper solutions. We also note that the data distributions we construct to establish these lower bounds are not exotic. For example, to prove these lower bounds, we construct a set of data distributions where all data distributions share the exact same feature distribution and all but one distribution share the exact same label distribution.

Theorem 5.3 states our lower bound. In this theorem, we refer to an algorithm as an (ε, δ)-collaborative learning algorithm if, for every collaborative learning problem $(\mathcal{H}, \mathcal{D})$, the algorithm returns an ε-optimal solution h, i.e., satisfying Equation 2, with probability at least $1-\delta$. We say that a learning algorithm $\mathcal{Q}$ is an (ε, δ)-optimal collaborative learning algorithm for a specific set of collaborative learning problems $\mathbb{V} \coloneqq \{(\mathcal{H}_i, \mathcal{D}_i)\}_i$ if, given any problem $(\mathcal{H}_i, \mathcal{D}_i) \in \mathbb{V}$, with probability at least $1-\delta$ the output of $\mathcal{Q}$ is an ε-optimal solution.

Theorem 5.3. *Take any* $n, d \in \mathbb{Z}_+$*,* $\varepsilon, \delta \in (0, 1/8)$*, and* (ε, δ)*-collaborative learning algorithm* $\mathcal{Q}$*. There exists a collaborative learning problem* $(\mathcal{H}, \mathcal{D})$ *with* $|\mathcal{D}| = n$ *and* $|\mathcal{H}| = 2^d$*, on which* $\mathcal{Q}$ *takes at least* $\Omega\left(\frac{1}{\varepsilon^2}\left(d + n\log(\min\{n, d\}/\delta)\right)\right)$ *samples. When* $n \leq d$*, this lower bound becomes* $\Omega\left(\frac{1}{\varepsilon^2}\left(d + n\log(n/\delta)\right)\right)$*.*

Before we proceed to a proof of this theorem, we first define a notion of expected sample complexity. Take any collaborative learning problem $V = (\mathcal{H}, \mathcal{D})$; we use $N_{\mathcal{Q}}(V)$ to denote the expected sample complexity of a collaborative learning algorithm $\mathcal{Q}$ on the problem V, where the expectation is taken both over the randomness of data samples and the algorithm's randomness. Similarly, given a probability distribution $\mathbb{P}$ over a set of collaborative learning problems $\mathbb{V} \coloneqq \{(\mathcal{H}_i, \mathcal{D}_i)\}_i$, we define expected sample complexity as $N_{\mathcal{Q}}(\mathbb{P}) = \mathbb{E}_{V\sim\mathbb{P}}\left[N_{\mathcal{Q}}(V)\right]$.

We now prove two lemmas, Lemma 5.4 and Lemma 5.5, that directly imply Theorem 5.3. Lemma 5.4 is a standard lower bound on the sample complexity of agnostic PAC learning, and provides the unsurprising $\Omega(\frac{d}{\varepsilon^2})$ lower bound summand. Lemma 5.5 is more involved and provides the $\Omega(\frac{n\log(\min\{n,d\}/\delta)}{\varepsilon^2})$ summand in our lower bound.

Lemma 5.4. *Take any $n, d \in \mathbb{Z}_+$, $\varepsilon, \delta \in (0, 1/8)$, and collaborative learning algorithm $\mathcal{Q}$. There exists a set of collaborative learning problems $\mathbb{V}$ on which, if $\mathcal{Q}$ is (ε, δ)-optimal, $\mathcal{Q}$ takes at least $\Omega\left(\frac{\log|\mathcal{H}|}{\varepsilon^2}\right)$ samples and where, for every $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}$, $|\mathcal{D}| = n$ and $|\mathcal{H}| = 2^d$.*

Proof. This claim follows directly from the standard lower bound on sample complexity of agnostic probably-approximately-correct (PAC) learning [54], since we can reduce any single-distribution learning problem to multi-distribution learning problem by defining multiple copies of a data distribution. We defer interested readers to Ehrenfeucht et al. [20]. □

Lemma 5.5. *Take any $n, d \in \mathbb{Z}_+$, $\varepsilon, \delta \in (0, 1/8)$, and (ε, δ)-collaborative learning algorithm $\mathcal{Q}$. There exists a set of collaborative learning problems $\mathbb{V}$ on which $\mathcal{Q}$ takes at least $\Omega\left(\frac{1}{\varepsilon^2}\left(n\log(k/\delta)\right)\right)$ samples and where, for every $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}$, $|\mathcal{D}| = n$ and $|\mathcal{H}| = 2^d$ with $k \coloneqq \min\{n, d\}$.*

Proof. We prove this lower bound constructively by defining multiple sets of collaborative learning instances: $\{\mathbb{V}_{w\eta,w}\}_{w,\eta\in\mathbb{N}}$. At a high-level, the proof of this lower bound will follow from proving that multi-distribution learning allows one to solve multiple single-distribution learning problems simultaneously with constant probability using a boosting-like algorithm.

We now detail our fairly technical construction of these collaborative learning instances. For every set of instances $\mathbb{V}_{u,w}$, we require all instances $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}_{u,w}$ to share a feature space $\mathcal{X} = \{1, \ldots, w\}$, label space $\mathcal{Y} = \{\pm 1\}$, hypothesis class $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$, and 0/1 loss ℓ. For every $x \in [w]$ and $y \in \{\pm 1\}$, we define distributions D_x and D'_x as having the probability mass functions $\Pr_{D_x}(x, y) = \frac{1}{2} - 2y\varepsilon$ and $\Pr_{D'_x}(x, y) = \frac{1}{2} + 4y\varepsilon$. Let $\mathcal{D}_- = \bigcup_{x\in[w]}\{D_x\}^\eta$ be an ordered list of distributions, and for every $x \in [w]$ and $i \in [\eta]$, define $\mathcal{D}_{x,i}$ to be a set of distributions identical to $\mathcal{D}_-$ except with the ith copy of distribution D_x replaced with distribution D'_x. Let $\mathbb{P}_{\eta w,w}$ be a distribution over collaborative learning instances that, with probability $\frac{1}{2}$, returns $(\mathcal{H}, \mathcal{D}_-)$ and for every $i \in [\eta], x^* \in [w]$, with probability $\frac{1}{2w\eta}$ returns $(\mathcal{H}, \mathcal{D}_{x^*,i})$. Observe that $\mathbb{P}_{\eta w,w}$ is a distribution over collaborative learning problems where $|\mathcal{H}| = 2^w$ and $|\mathcal{D}| = u$. The following claims characterize sample complexity lower bounds on $\mathbb{P}_{u,w}$.

Claim 5.1. *Consider any $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1)$, and collaborative learning algorithm $\mathcal{Q}$ that is (ε, δ)-optimal for $\mathbb{V}_{\eta,1}$. The expected sample complexity of $\mathcal{Q}$ is at least $\frac{\eta}{256\varepsilon^2}\log(1/2\delta)$.*

Claim 5.2. *Consider any $\varepsilon \in (0, 1/2)$ and $\delta \in (0, 1)$. Suppose there exists a collaborative learning algorithm $\mathcal{Q}$ that is (ε, δ)-optimal for $\mathbb{V}_{\eta w,w}$ and has an expected sample complexity of N under $\mathbb{P}_{\eta w,w}$. Then there exists an $(\varepsilon, \frac{8\delta}{7w})$-learning algorithm $\mathcal{Q}'$ for $\mathbb{V}_{\eta,1}$ under $\mathbb{P}_{\eta,1}$ with an expected sample complexity on $\mathbb{P}_{\eta,1}$ of $\frac{8}{7w}N$.*

Since our desired lower bound is weakly monotonic in n, d, we fix the smallest choice of $\eta, d \in \mathbb{Z}_+$ and $\varepsilon, \delta \in (0, 1/8)$ such that $n = \eta \cdot d$. Combining claims 5.1 and 5.2, we see that any (ε, δ) collaborative learning algorithm $\mathcal{Q}$ for $\mathbb{V}_{n,d}$ has an expected sample complexity on $\mathbb{P}_{n,d}$ of at least $N \geq \frac{7n}{2048\varepsilon^2}\log\left(\frac{7d}{16\delta}\right)$. By the probabilistic method, for at least some collaborative learning problem in the set $\mathbb{V}_{n,d}$, our learning algorithm $\mathcal{Q}$ must have a sample complexity of $\Omega\left(\frac{7n}{2048\varepsilon^2}\log\left(\frac{7d}{16\delta}\right)\right)$. □

Proof of Claim 5.1. Consider η two-sided coins. Under a H_0 hypothesis, all coins are biased towards tails with probability $1/2 + 2\varepsilon$. Under a H_i hypothesis, the ith coin is biased towards heads with probability $1/2 + 4\varepsilon$. Let Pr be a probability distribution on $H \in \{H_i\}_{i=0}^\eta$ with $\Pr(H_0) = 1/2$ and $\Pr(H_1) = \cdots = \Pr(H_\eta) = \frac{1}{2\eta}$. Given an (ε, δ)-algorithm $\mathcal{Q}$ for $\mathbb{V}_{\eta,1}$ with an expected sample complexity of N (under $\mathbb{P}_{\eta,1}$), we can construct a coin algorithm $\mathcal{Q}'$ with an expected sample complexity of N (under Pr) and that, under any hypothesis, with probability at least $1 - \delta$, can identify whether H_0 is false.

To see this, have $\mathcal{Q}'$ run $\mathcal{Q}$ by simulating draws from the ith distribution by flipping the ith coin. If all coins are biased towards tails with probability $1/2 + 2\varepsilon$, any ε-error hypothesis h must satisfy $\Pr(h(1) = +) > 1/2$. Conversely, if one coin is biased towards heads, any ε-error hypothesis h must satisfy $\Pr(h(1) = +) < 1/2$.

Suppose $\mathcal{Q}'$, conditioned on H_0, correctly predicts H_0 with probability at least $1-\delta$. Then, suppose $\mathcal{Q}'$, under H_0, takes no more than T_i flips from the ith coin. Let $p_{i,j_1:j_2}$ be a probability distribution over $\{0,1\}$ corresponding to the outcomes of the j_1st to j_2nd coin toss by $\mathcal{Q}'$ under H_i. Let p_j^* be a uniform distribution over $\{0,1\}^j$. Since $p_{i,j:j}$ and p_j^* are Bernoulli distributions with a parameter within 4ε of $1/2$, for $\varepsilon < 1/2$, $\mathrm{KL}(p_{i,j:j}, p_1^*) < 128\varepsilon^2$ [60]. Moreover, $\mathrm{KL}(p_{i,1:j}, p_j^*) < 128j\varepsilon^2$ by tensorization and $\mathrm{TV}(p_{i,1:j}, p_j^*) \leq 8\varepsilon\sqrt{j}$ by Pinsker's inequality. Let E be the set of outcomes of T_i flips under which $\mathcal{Q}'$ predicts H_0. By correctness under H_0, we have that $\Pr_{H_0}(E) \geq 1-\delta$. Thus, total variation distance implies $1-\delta-8\varepsilon\sqrt{j} < \Pr_{H_i}(E)$. Since $\Pr_{H_i}(E) < \delta$, we have that $\frac{1}{128\varepsilon^2}(1-2\delta)^2 < T_i$. Thus, if $\mathcal{Q}'$ is δ accurate under all hypotheses, under H_0, $\mathcal{Q}'$ must take at least $\frac{\eta}{128\varepsilon^2}(1-2\delta)^2 < \frac{\eta}{128\varepsilon^2}\log(1/2\delta)$ samples from each distribution. Thus, the expected sample complexity of $\mathcal{Q}'$—and similarly that of $\mathcal{Q}$ under $\mathbb{P}_{\eta,1}$—must be at least $\frac{\eta}{256\varepsilon^2}\log(1/2\delta)$. □

Proof of Claim 5.2. This claim is similar to the lower bounds of Blum et al. [9] and Karp and Kleinberg [30]. We construct $\mathcal{Q}'$ as follows. Define the shorthand $I_j := [(j-1)\eta+1, j\eta]$. Consider any problem $V' = (\mathcal{H}, \mathcal{D}) \in \mathbb{V}_{\eta,1}$.

1. $\mathcal{Q}'$ draws an imaginary problem $(\mathcal{H}, \mathcal{D}') \in \mathbb{V}_{\eta w, w}$ and chooses an index $i \in [w]$ uniformly at random.
2. $\mathcal{Q}'$ simulates algorithm $\mathcal{Q}$ on $(\mathcal{H}, \mathcal{D}')$: when $\mathcal{Q}$ tries to sample a datapoint from distribution D'_j where $j \notin I_i$, return a sample from D'_j; when $j \in I_i$, return a sampled datapoint from $D_{j-(i-1)\eta}$.
3. When $\mathcal{Q}$ terminates and returns a classifier h, $\mathcal{Q}'$ checks whether, for every $j \neq i$: $\max_{r\in I_j} \mathcal{R}_{D_r}(h) < \frac{1}{2}$. If this condition is satisfied, $\mathcal{Q}'$ returns $h(1) = h(i)$. If not, we repeat from Step 1. We denote the number of total iterations by T.

Consider the probability p_i that, in the third step, for every $j \neq i$ we have $\max_{r\in I_j} \mathcal{R}_{D_r}(h) < \frac{1}{2}$ but $\max_{r\in I_i} \mathcal{R}_{D_r}(h) \geq \frac{1}{2}$. Let E_t denote the event that $\mathcal{Q}'$ returns an at least ε-error hypothesis after t iterations of our procedure. Noting that E_t can only occur if $\mathcal{Q}$ failed all $t-1$ iterations before and at the tth iteration, Step 3 fails to catch the bad hypothesis for D_i. By assumption, $\delta \geq \sum_{i=1}^w p_i$. By symmetry of our construction $\mathbb{V}$ and recalling $\delta < 1/8$: $\sum_{t=1}^\infty \Pr(E_t) \leq \sum_{t=1}^\infty \delta^{t-1}\frac{1}{w}\sum_{i=1}^w p_i \leq \sum_{t=1}^\infty \delta^t/w \leq \frac{8\delta}{7w}$. Thus, $\mathcal{Q}'$ is an $(\varepsilon, \frac{8\delta}{7w})$-algorithm for $\mathbb{P}_{\eta,1}$.

We now bound the sample complexity of $\mathcal{Q}'$. Let $N_{\mathcal{Q}'}(t)$ denote the number of samples that $\mathcal{Q}'$ takes from V' on the tth iteration. Note that $N_{\mathcal{Q}'}(1), N_{\mathcal{Q}'}(2), \ldots$ are i.i.d. In addition, by the symmetry of $\mathbb{V}$ and linearity of expectation, $\mathbb{E}_{V'\in\mathbb{P}_{\eta,1}}[N_{\mathcal{Q}'}(t)] = m/w$. Thus, $\mathbb{E}_{V'}\left[\sum_{t=1}^T N_{\mathcal{Q}'}(t)\right] = \mathbb{E}_{V'}[T]\,\mathbb{E}_{V'}[N_{\mathcal{Q}'}(1)] = \mathbb{E}_{V'}[T]\,m/w$. We can upper bound T by observing that our procedure only repeats if $\mathcal{Q}$ fails: $\mathbb{E}_{V'}[T] = \sum_{t=1}^\infty \Pr(T \geq t) \leq \sum_{t=0}^\infty \delta^t \leq \frac{1}{1-\delta} \leq \frac{8}{7}$. Thus, $\mathcal{Q}'$ has an expected sample complexity of at most $\frac{8m}{7w}$. □

6 Group DRO and Agnostic Federated Learning

In this section, we present our main result on the sample complexity of the group distributionally robust optimization framework of Sagawa et al. [50] and the agnostic federated learning framework of Mohri et al. [39]. We show that the worst-case sample complexity of group DRO, and equivalently agnostic federated learning, is greater than that of online convex optimization by only a constant factor and an additive $O(n\log(n/\delta)/\varepsilon^2)$ samples. This sample complexity upper bound is tight for a difficult class of problems—a class that coincides with collaborative learning. Since the settings of group DRO and agnostic federated learning are generally equivalent, we state the results explicitly for group DRO, with the understanding that the same results apply to agnostic federated learning.

Setup. Group distributionally robust optimization is typically studied in a convex optimization setting where the hypothesis class is parameterized by a convex compact parameter class and the loss function is smooth and convex in the parameterization. As noted previously, this means that the group DRO setting coincides with general setting of multi-distribution learning with a single smooth convex loss. Moreover, group DRO is usually formulated in a setting where the parameter space admits mirror descent approaches.

We first present the definitions which are necessary for describing mirror descent guarantees. A distance-generating function on a parameter space Θ is a continuous and strongly convex, modulus 1, function

$\omega : \Theta \to \mathbb{R}$, where there exists a non-empty subset of the parameter space $\Theta^o \subset \Theta$ where the subdifferential $\partial\omega$ is non-empty and $\partial\omega$ admits a continuous selection on Θ^o. The center of Θ with respect to ω is denoted as $\theta^c \coloneqq \arg\min_{\theta\in\Theta^o} \omega(\theta)$. The Prox function (Bregman divergence) $V : \Theta^o \times Z \to \mathbb{R}^+$ associated with a distance-generating function $\omega : Z \to \mathbb{R}$ is defined as $V(w,u) \coloneqq \omega(u) - \omega(w) - \langle \omega'(w), u - w\rangle$. Bregman radius, which is a measure for how difficult it is to learn a parameter class, is then defined as follows.

Definition 6.1. *Given a convex set Θ with a distance-generating function ω, the Bregman radius is defined as $D_\Theta \coloneqq \max_{u\in\Theta} V(\theta^c, u)$ where θ^c is the center of Θ.*

A bounded Bregman radius allows one to apply online mirror descent [6] as an online learning algorithm, with a regret guarantee of $\gamma_T(\mathcal{Q}_\Theta) \leq D_\Theta$. In group DRO, D_Θ is typically assumed to be small.

Sample complexity upper bound. Theorem 6.1 states our sample complexity bound for group distributionally robust optimization. This sample complexity bound is a direct implication of our multi-distribution learning sample complexity bound. This theorem establishes the first generalization bound for the problem of group distributionally robust optimization [50] and improves, by a factor of n, existing sample complexity bounds for agnostic federated learning [39]. This significant improvement in sample complexity over Mohri et al. [39] is attained by sampling data on-demand, whereas Mohri et al. [39] work with a distribution over groups/clients that is fixed a priori.

Theorem 6.1. *Given a set of data distributions $\mathcal{D} = \{D_1, \ldots, D_n\}$, a hypothesis class $\mathcal{H}$ with a Bregman radius of D_Θ and a diameter of R, and a 1-smooth loss ℓ, consider the group distributionally robust optimization problem $(\mathcal{D}, \{\ell\}, \mathcal{H})$. Consider the output $\theta \in \Theta$ arising from applying Theorem 4.1's algorithm, choosing the online learning algorithm $\mathcal{Q}$ to be online mirror descent. With probability $1-\delta$, h is an ε-optimal solution (see (3)) and the sample complexity is $O\left(\varepsilon^{-2}\left(D_\Theta + n\log(n/\delta) + R\log(1/\delta)\right)\right)$.*

Proof. This claim follows directly by Theorem 4.1 since group distributionally robust optimization is equivalent to multi-distribution learning on a single smooth convex loss. Recall that, for a convex parameter space with Bregman radius D_Θ for a distance-generating function ω, running the online mirror descent algorithm with respect to ω guarantees a regret bound of $\gamma_T(\mathcal{Q}_\Theta) \leq D_\Theta$ [6]. We directly plug this online convex optimization regret bound into Theorem 4.1. □

This sample complexity bound for finding a group DRO solution with low expected loss also trivially implies a bound on the number of mirror descent iterations that are necessary to find a group DRO or agnostic federated learning solution with low empirical training error. This question was considered by Sagawa et al. [50] who presented an iteration complexity bound that we improve upon by a factor of n.

Corollary 6.2 (Theorem 6.1)**.** *Consider a group distributionally robust optimization problem $(\mathcal{D}, \{\ell\}, \mathcal{H})$. For every $D \in \mathcal{D}$, let $B_D \sim D$ be a non-empty batch of i.i.d. datapoints and D' be the empirical distribution of B_D. There is an algorithm that only requires $O\left(\varepsilon^{-2}\left(D_\Theta + n\log(n/\delta) + R\log(1/\delta)\right)\right)$ iterations of mirror descent steps to output, with probability $1-\delta$, an* empirically *ε-optimal solution.*

Sample complexity lower bound. There exists a class of difficult group distributionally robust optimization problems for which our stated sample complexity upper bounds are tight. This is because we can reduce any collaborative learning problem to multi-distribution learning with a single smooth convex loss, and equivalently, group DRO. Thus, our sample complexity lower bound for collaborative learning directly implies a lower bound for group DRO for a class of difficult cases. We formally state this corollary of Theorem 5.3 below.

Corollary 6.3. *Take any $n, m \in \mathbb{N}$ and $\varepsilon, \delta \in (0, 1/8)$. There exists a finite set $\mathbb{V}$ of group distributionally robust optimization problems with 1-smooth losses and parameter spaces of unit diameter and finite Bregman radius D_Θ, where every (ε,δ)-algorithm $\mathcal{Q}$ has a sample complexity in $\Omega\left(\frac{D_\Theta + n\log(\min\{n, D_\Theta\}/\delta)}{\varepsilon^2}\right)$.*

7 Extensions to Infinite Classes of Binary Classifiers

In this section, we study the sample complexity of multi-distribution learning when the hypothesis class is infinite but combinatorially bounded. In particular, we will study multi-distribution learning problems involving binary classification tasks and hypothesis classes of finite VC dimension or finite Littlestone dimension [33]. For succinctness, we state all results in this section for the collaborative learning setting, but note that these results extend readily to the general multi-distribution learning setting.

Littlestone dimension. The Littlestone dimension of a set of binary classifiers quantifies the set's online learnability [33]. Formally, consider a supervised learning setting with domain $\mathcal{X}$ and a set of binary classifiers $\mathcal{H}$. Consider a full binary tree of depth d, such that each node in the tree is labeled by a feature $x \in \mathcal{X}$. We say the tree is shattered by $\mathcal{H}$ if for every set of labels $\{y_i\}_{i=1}^d \in \{\pm 1\}^d$, the root-to-leaf path $x_1, \ldots, x_d$ that is defined by starting at the root and moving to the left child if $y_i = +1$ and right if $y_i = -1$, there exists a classifier $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $i \in [d]$—that is, h agrees with the labels we used to reach nodes in the path. In other words, a tree is shattered if every path in the tree is labeled by some hypothesis $h \in \mathcal{H}$. We say that the Littlestone dimension of the classifiers $\mathcal{H}$ is d if d is the maximal depth of a tree that is shattered by $\mathcal{H}$.

It is not hard to see that Littlestone dimension upper bounds VC dimension and lower bounds log-cardinality $\log(|\mathcal{H}|)$. For binary classifier multi-distribution learning problems, we can strengthen our collaborative learning sample complexity upper bound of Theorem 5.1 to be stated in terms of the Littlestone dimension of a hypothesis class $\mathrm{LD}(\mathcal{H})$ rather than $\log(|\mathcal{H}|)$. This is because there exists an online learning algorithm that guarantees a regret bound of $\gamma_T(\mathcal{Q}_{\Delta(\mathcal{H})}) \in O(\mathrm{LD}(\mathcal{H}))$ that we can have the learner play instead of an algorithm like Hedge.

Theorem 7.1 (Littlestone Dimension Variant of Theorem 5.1)**.** *Given a set of data distributions $\mathcal{D} = \{D_1, \ldots, D_n\}$, a hypothesis class of binary classifiers $\mathcal{H} \in \{0,1\}^{\mathcal{X}}$, and a $[0,1]$-bounded loss ℓ, consider the collaborative learning problem $(\mathcal{H}, \mathcal{D})$. Consider the output $h \in \Delta(\mathcal{H})$ of applying Theorem 4.1's algorithm to the multi-distribution learning problem $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$ where the online learning algorithm $\mathcal{Q}_{\Delta(\mathcal{H})}$ is the agnostic Standard-Optimal-Algorithm of Alon et al. [2]. With probability $1-\delta$, h is an ε-optimal solution (see (2)) to $(\mathcal{H}, \mathcal{D})$ and the sample complexity is $O\left(\varepsilon^{-2}\left(\mathrm{LD}(\mathcal{H}) + n\log(n/\delta)\right)\right)$.*

Proof. By Fact 2.1, we can reduce the collaborative learning problem $(\mathcal{H}, \mathcal{D})$ to solving the multi-distribution learning problem $(\mathcal{D}, \{\ell\}, \Delta(\mathcal{H}))$ The agnostic SOA algorithm of Alon et al. [2] guarantees a regret bound of $\gamma_T(\mathcal{Q}_{\Delta(\mathcal{H})}) = \mathrm{LD}(\mathcal{H})$. Our claim therefore follows by Theorem 4.1. □

We remark that a similar sample complexity bound can be achieved using the original Standard Optimal Algorithm (SOA) of Littlestone [33] instead of the implicit algorithm of Alon et al. [2], as SOA guarantees a regret bound of $\gamma_T(\mathcal{Q}_{\Delta(\mathcal{H})}) \in O\left(\sqrt{\mathrm{LD}(\mathcal{H})T\log(T)}\right)$.

VC dimension. It is also nature to ask for the sample complexity of multi-distribution learning in terms of VC dimension $\mathrm{VC}(\mathcal{H})$, which characterizes the sample complexity of learning a single data distribution. For example, Blum et al. [9], Nguyen and Zakynthinou [42], Chen et al. [13] provided upper bounds for binary classification multi-distribution learning that are identical to their upper bounds in Table 1 but replacing $\log(|\mathcal{H}|)$ with $\mathrm{VC}(\mathcal{H})$. We now show a similar result to Theorem 5.1 also holds with dependence on the VC dimension of $\mathcal{H}$ only when additional mild assumptions hold. In particular, one can run Algorithm 2 on a hypothesis class $\mathcal{H}'$ that is known to be an ε-net of $\mathcal{H}$ with respect to each distribution in $\mathcal{D}$. Such an ε-net of size $(n/\varepsilon)^{O(\mathrm{VC}(\mathcal{H}))}$ necessarily exists (see, e.g., [3]). For example, we can project $\mathcal{H}$ onto the union of datapoints sampled from each distribution $D \in \mathcal{D}$. When such a $\mathcal{H}'$ is known in advance, we may directly run Algorithm 2 with $\mathcal{H}'$.

Corollary 7.2. *Given a set of data distributions $\mathcal{D} = \{D_1, \ldots, D_n\}$, a hypothesis class of binary classifiers $\mathcal{H} \in \{0,1\}^{\mathcal{X}}$ of VC dimension d, and a $[0,1]$-bounded loss ℓ, consider the collaborative learning problem $(\mathcal{H}, \mathcal{D})$. Suppose we are further given a set of classifiers of size $\mathrm{poly}\left((n/\varepsilon)^d, \varepsilon, d, n\right)$ that is an ε-net of $\mathcal{H}$ for each distribution $D \in \mathcal{D}$. There is an algorithm that, with probability $1-\delta$, returns an ε-optimal solution (see (2)) to $(\mathcal{H}, \mathcal{D})$ with a sample complexity of $O\left(\varepsilon^{-2}\left(d\log(dn/\varepsilon) + n\log(n/\delta)\right)\right)$.*

It is not strictly necessary to know an ε-net in advance. Instead, one can compute a net from samples or from other information about distributions in $\mathcal{D}$. There a range of assumptions that allow us to compute such an ε-net from samples, without incurring a significant increase in sample complexity. For example, when ε is sufficiently small, specifically $\varepsilon \in O(1/n)$ (*Assumption 1*), taking an ε-net only increases the sample complexity bound by constant factors versus knowing an ε-net in advance. Additional examples include:

- *Assumption 2*: we know the marginal distribution for all $D \in \mathcal{D}$;
- *Assumption 3*: we have access to n marginal distributions $P_1, \ldots, P_n$ such that for all $x \in \mathcal{X}$, $D_i(A) \leq p_i(A)\text{poly}(1/\varepsilon, d(\mathcal{H}), n)$ for all $A \subseteq \mathcal{X}$, where p_i and D_i are the densities of P_i and D_i, respectively.

These latter two assumptions allow one to construct ε-nets of small size for free.

Theorem 7.3. *Given a set of data distributions $\mathcal{D} = \{D_1, \ldots, D_n\}$, a hypothesis class of binary classifiers $\mathcal{H} \in \{0,1\}^{\mathcal{X}}$ of VC dimension d, and a $[0,1]$-bounded loss ℓ, consider the collaborative learning problem $(\mathcal{H}, \mathcal{D})$. If any of Assumptions 1, 2 or 3 is met, there is an algorithm that, with probability $1-\delta$, returns an ε-optimal solution (see (2)) to $(\mathcal{H}, \mathcal{D})$ with a sample complexity of $O\left(\varepsilon^{-2}\left(d\log(dn/\varepsilon) + n\log(n/\delta)\right)\right)$.*

Proof. For a data distribution D, we will use $D_{\mathcal{X}}$ to denote the marginal distribution of D. We also use the shorthand $d_\infty(P||Q) \coloneqq \sup_{x \in \mathcal{X}_Q} \frac{P(x)}{Q(x)}$, where $d_\alpha(P||Q) \coloneqq 2^{D_\alpha(P||Q)}$ can be understood as the power of the Renyi divergence $D_\alpha(P||Q)$. We first recall a standard fact about covering with projections.

Lemma 7.4 (Corollary 3.7 in Haussler and Welzl [25])**.** *Let $\mathcal{F}$ be a function class consisting of functions from $\mathcal{X}$ to $[0,1]$ and let $\mathcal{P}$ be a probability measure on $\mathcal{X}$. Given $N \geq \frac{8d}{\varepsilon}\log\frac{8d}{\varepsilon} + \frac{4}{\varepsilon}\log\frac{2}{\delta}$ independent samples $\mathbf{x}$ from $\mathcal{P}$, with probability at least $1-\delta$, the projection of $\mathcal{F}$ on $\mathbf{x}$ constitutes an ε-net. That is, for any $f_1, f_2 \in \mathcal{F}$ where $\Pr_{x\sim\mathcal{P}}(f_1(x) \neq f_2(x)) \geq \varepsilon$, $\|f_1(x) - f_2(x)\|_{\mathbf{x}} > 0$.*

The following corollaries of Theorem 5.1 directly imply Theorem 7.3.

Corollary 7.5 (Assumption 1)**.** *For $\varepsilon \in O(1/n)$, there is an algorithm that, with probability $1-\delta$, returns an ε-optimal solution $\overline{h} \in \Delta(\mathcal{H})$ using a number of samples that is $O\left(\frac{d\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.*

Proof. By Lemma 7.4, sampling $O\left(\frac{nd}{\varepsilon}\log(\frac{d}{\varepsilon}) + \frac{n}{\varepsilon}\log(\frac{n}{\varepsilon})\right)$ datapoints provides a covering of $\mathcal{H}$ is that is simultaneously an ε-net for every $D \in \mathcal{D}$ with probability at least $1-\delta$. Moreover, by the Sauer-Shelah lemma, this net is of size $O\left(\left(\frac{\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)^d\right)$. The claim then follows from Corollary 7.2, noting that since $\varepsilon \in O(1/n)$, we only needed to sample an additional $O\left(\frac{d}{\varepsilon^2}\log(\frac{d}{\varepsilon}) + \frac{n}{\varepsilon}\log(\frac{n}{\varepsilon})\right) \subset O\left(\frac{nd}{\varepsilon}\log(\frac{d}{\varepsilon}) + \frac{n}{\varepsilon}\log(\frac{n}{\varepsilon})\right)$ datapoints to form the cover. □

Corollary 7.6 (Assumption 2)**.** *We say an algorithm has weak unlabeled access if the algorithm can access, for each $D \in \mathcal{D}$, a marginal distribution $D'_{\mathcal{X}}$ such that $D_\infty(D'_{\mathcal{X}}||D_{\mathcal{X}}) \in \text{poly}(1/\varepsilon, d, n)$, with probability $1-\delta$. There is an algorithm that, given weak access, with probability $1-\delta$, returns an ε-optimal solution $\overline{h} \in \Delta(\mathcal{H})$ using a number of samples that is $O\left(\frac{d\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.*

Proof. Observe that when $D_\infty(D'_{\mathcal{X}}||D_{\mathcal{X}}) < \gamma$, $D'_{\mathcal{X}}$ can be written as a mixture over $D_{\mathcal{X}}$ with probability at least $\frac{1}{\gamma}$ and some other distribution $\widetilde{D}_{\mathcal{X}}$ with probability at most $1 - \frac{1}{\gamma}$. Once again invoking uniform convergence, we observe that sampling $\Theta\left(D_\infty(D'_{\mathcal{X}}||D_{\mathcal{X}})\frac{d\log(d/\varepsilon)+\log(1/\delta)}{\varepsilon^2}\right)$ i.i.d. samples from distribution $D'_{\mathcal{X}}$, with probability at least $1-\delta$, yields an ε-covering on D. By the Sauer-Shelah lemma, the resulting covering $\mathcal{H}'_D$ is of size $O\left((\text{poly}(1/\varepsilon, d, n))^d\right)$. Repeating this procedure for each $D \in \mathcal{D}$, with probability at least $1 - n\delta$, we have an ε-covering $\mathcal{H}'$ of $\mathcal{D}$ of size $|\mathcal{H}'| \in O\left(n(\text{poly}(1/\varepsilon, d, n))^d\right)$. We can then appeal directly to Theorem 5.1 for a sample complexity bound on learning $(\mathcal{H}', \mathcal{D})$. □

Corollary 7.7 (Assumption 3)**.** *There is an algorithm that, given access to the marginal distribution $D_{\mathcal{X}}$ of every $D \in \mathcal{D}$, with probability $1-\delta$, returns an ε-optimal solution $\overline{h} \in \Delta(\mathcal{H})$ using a number of samples that is $O\left(\frac{d\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.*

Proof. By uniform convergence, taking $\Theta\left(\frac{d\log(d/\varepsilon)+\log(1/\delta)}{\varepsilon^2}\right)$ i.i.d. samples from distribution $D_{\mathcal{X}}$ for each $D \in \mathcal{D}$, with probability at least $1-\delta$, yields an ε-covering on every $D \in \mathcal{D}$. By the Sauer-Shelah lemma, the resulting covering $\mathcal{H}'_D$ is of size $O\left((n\varepsilon^{-2}(\log(d/\varepsilon)+\frac{1}{d}\log(1/\delta)))^d\right)$. We then appeal to Corollary 7.2. □

□

One question left open by these results is whether, for agnostic collaborative learning, it is possible to achieve sample complexity rates of $O\left(\varepsilon^{-2}\left(\log(n)\mathrm{VC}(\mathcal{H})+n\log(n/\delta)\right)\right)$ without any additional assumptions or a priori knowledge of an ε-net. It also remains an open question whether the $\log(n)$ factor in the $\log(n)\mathrm{VC}(\mathcal{H})/\varepsilon^2$ term is necessary for VC classes, as Theorem 5.1 proves that, for finite/online-learnable classes with sample complexities expressed in terms of $\log(|\mathcal{H}|)$ or Littlestone dimension $\mathrm{LD}(\mathcal{H})$, no such $\log(n)$ factor is necessary. We refer interested readers to Awasthi et al. [4] for a complete discussion of these open problems.

8 Empirical Analysis of On-Demand Sampling for Group DRO

This section describes experiments where we adapt our on-demand sampling-based multi-distribution learning algorithm for deep learning applications. In particular, we compare our algorithm against the de facto standard multi-distribution learning algorithm for deep learning, Group DRO (GDRO) [50]. As GDRO is designed for use with offline-collected datasets, to provide a meaningful comparison, we modify our algorithm to work on offline datasets (i.e., with no on-demand sample access).

		Worst-Group Accuracy			Gap in Avg. vs. Worst-Group Acc.		
		ERM	GDRO	R-MDL	ERM	GDRO	R-MDL
Standard Reg.	Waterbirds	60.0 (1.9)	76.9 (1.7)	**86.4 (1.4)**	37.3 (1.9)	20.5 (1.7)	**8.1 (1.4)**
	CelebA	41.1 (3.7)	41.7 (3.7)	**88.9 (2.3)**	53.7 (3.7)	53 (3.7)	**3.4 (2.3)**
	MultiNLI	66.3 (1.6)	66.6 (1.6)	**70.3 (1.5)**	16.2 (1.6)	15.6 (1.6)	**4.5 (1.5)**
Strong Reg.	Waterbirds	21.3 (1.6)	84.6 (1.4)	**89.4 (1.2)**	74.4 (1.6)	12 (1.4)	**0.4 (1.3)**
	CelebA	37.8 (3.6)	86.7 (2.5)	**88.8 (2.3)**	58 (3.6)	6.8 (2.5)	**1.2 (2.3)**
Early Stop	Waterbirds	6.7 (1.0)	85.8 (1.4)	**87.1 (1.3)**	87.1 (1.0)	7.4 (1.4)	**5.6 (1.3)**
	CelebA	25.0 (3.2)	88.3 (2.4)	**90.6 (2.2)**	69.6 (3.2)	3.5 (2.4)	**0.7 (2.2)**
	MultiNLI	66.0 (1.6)	**77.7 (1.4)**	43.1 (1.7)	16.8 (1.6)	**3.7 (1.4)**	18.3 (1.7)

Table 2: Worst-group accuracy (our primary performance metric) and the gap between worst-group accuracy and average accuracy, of empirical risk minimization (ERM), Group DRO (GDRO), and our R-MDL algorithm in three experiment settings—standard hyperparameters (Standard Reg.), inflated weight decay regularization (Strong Reg.), and early stopping (Early Stop)—and on three datasets—Waterbirds, CelebA, and MultiNLI. Figures are percentages evaluated on the test split of each dataset, with standard deviation in parentheses. R-MDL consistently outperforms GDRO and performs reliably with or without strong regularization.

Resampling Multi-Distribution Learning (R-MDL). To be more suitable for deep learning applications, we instantiate Algorithm 2 by choosing a minibatch gradient descent algorithm as the minimizing player's algorithm ($\mathcal{Q}_\Theta$) and a naive uniform-sampling bandit algorithm as the maximizing player's algorithm ($\mathcal{Q}_{\Delta(\mathcal{D})}$).

We can further adapt our algorithm to offline datasets by simulating on-demand sampling on the empirical distributions of datasets. This modified algorithm, R-MDL, is described in full in Algorithm 3.

Note that, in contrast, the original group DRO algorithm of Sagawa et al. [50] is also a minibatch gradient descent algorithm but samples minibatches uniformly from all distributions and weights datapoints via a no-regret algorithm that provides importance weights. Though effective, this method is brittle and requires tricks like unconventionally strong regularization [50]. Our theory of on-demand sampling suggests that R-MDL should mollify this brittleness, as it replaces GDRO's upweighting of low-accuracy distributions with upsampling of low-accuracy distributions. Interestingly, the advantage of *resampling* over *reweighting* has been previously observed when training neural networks on a dataset with fixed importance weights [51].

Experiment Setting In Table 2, we replicate the Group DRO experiments of Sagawa et al. [50] and compare the standard GDRO algorithm with our R-MDL algorithm (Algorithm 3). We fine-tune Resnet-50 models (convolutional neural networks) [26] and BERT models (transformer-based network) [17] on the image classification datasets Waterbirds [50, 56] and CelebA [34] and the natural language dataset MultiNLI [57] respectively. We train these models in 3 settings: with standard hyperpameters, under strong weight decay (ℓ-2) regularization, or under early stopping.

R-MDL consistently outperforms GDRO and ERM. In every dataset and in almost every setting, R-MDL significantly outperforms GDRO and ERM in worst-group accuracy. In addition, whereas GDRO and ERM have large gaps between worst-group accuracy and average accuracy, R-MDL has almost matching worst-group and average accuracies. This indicates that R-MDL is more effective at prioritizing learning on difficult groups.

R-MDL is robust to regularization strength. R-MDL retains high worst-group accuracy even without strong regularization. These results challenge the observation of Sagawa et al. [50] that strong regularization is critical for the performance of Group DRO methods. This suggests that the brittleness of GDRO is due to the reweighting rendering the adversary too weak. In contrast, R-MDL provides a robust multi-distribution learning method with significantly less hyperparameter sensitivity.

9 Conclusions

While learning from a single data distribution is a fundamental abstraction of data-driven pattern recognition, data-driven decision-making calls for a new perspective that captures learning problems involving multiple stakeholders and data sources. This work proposes multi-distribution learning as a unifying theoretical framework, bringing together a number of widely studied problem formulations such as group distributionally robust optimization and collaborative PAC learning under a single umbrella. This unifying perspective distills the challenges of these various learning problems to a fundamental question about the sample complexity of stochastic games. We answered this fundamental question by providing optimal rates for a broad class of problems including convex and Littlestone hypothesis classes, highlighting the importance of on-demand sampling for decoupling the complexity of learning and obtaining robustness. We believe these findings underscore a broader takeaway that adaptive data collection is fundamental for scalable learning outside the single-distribution paradigm of classical pattern recognition.

10 Acknowledgments

This work was supported in part by the National Science Foundation under grant CCF-2145898, a C3.AI Digital Transformation Institute grant, and the Mathematical Data Science program of the Office of Naval Research. This work was partially done while Haghtalab and Zhao were visitors at the Simons Institute for the Theory of Computing. The authors thank the authors of Chen et al. [13] and Sagawa et al. [50] for communication regarding their work. The authors also thank Tianyi Lin, Guy Rothblum, Abhishek Shetty, Tatjana Chavdarova, Lydia Zakynthinou, and Mingda Qiao for valuable discussions.

References

[1] N. Alon, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. From bandits to experts: a tale of domination and independence. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1610–1618. Curran Associates, Inc., 2013.

[2] N. Alon, O. Ben-Eliezer, Y. Dagan, S. Moran, M. Naor, and E. Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, volume 53, pages 447–455. ACM, 2021.

[3] M. Anthony and P. L. Bartlett. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press, 2002. ISBN 978-0-521-57353-5.

[4] P. Awasthi, N. Haghtalab, and E. Zhao. Open problem: The sample complexity of multi-distribution learning for vc classes. In *Proceedings of the Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research. PMLR, 2023.

[5] M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the Conference on Learning Theory (COLT)*, volume 23 of *Proceedings of Machine Learning Research*, pages 26.1–26.22. PMLR, 2012.

[6] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[7] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 567–580. Springer Berlin Heidelberg, 2003.

[8] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009.

[9] A. Blum, N. Haghtalab, A. D. Procaccia, and M. Qiao. Collaborative PAC learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2392–2401. Curran Associates, Inc., 2017.

[10] A. Blum, N. Haghtalab, R. L. Phillips, and H. Shao. One for one, or all for all: equilibria and optimality of collaboration in federated learning. In M. Meila and T. Zhang, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 1005–1014. PMLR, 2021.

[11] A. Blum, S. Heinecke, and L. Reyzin. Communication-aware collaborative learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 6786–6793. AAAI Press, 2021.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1): 1–122, Jan 2011.

[13] J. Chen, Q. Zhang, and Y. Zhou. Tight bounds for collaborative PAC learning via multiplicative weights. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3602–3611. Curran Associates, Inc., 2018.

[14] C. Daskalakis, A. Deckelbaum, and A. Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 22, pages 235–254. SIAM, 2011.

[15] C. Daskalakis, M. Fishelson, and N. Golowich. Near-optimal no-regret learning in general games. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34*, pages 27604–27616. Curran Associates, Inc., 2021.

[16] H. Daumé, J. M. Phillips, A. Saha, and S. Venkatasubramanian. Efficient protocols for distributed classification and optimization. In *Proceedings of the Algorithmic Learning Theory*, volume 7568 of *Lecture Notes in Computer Science*, pages 154–168. Springer, 2012.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics, 2019.

[18] J. C. Duchi and H. Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378 – 1406, 2021.

[19] C. Dwork, M. P. Kim, O. Reingold, G. N. Rothblum, and G. Yona. Outcome indistinguishability. In *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1095–1108. ACM, 2021.

[20] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[21] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[22] N. Haghtalab, M. Jordan, and E. Zhao. A unifying perspective on multi-calibration: Game dynamics for multi-objective learning. In *Advances in Neural Information Processing Systems 37*, 2022.

[23] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[24] T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In J. Dy and A. Krause, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 1934–1943. PMLR, 2018.

[25] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG '86, page 61–71. Association for Computing Machinery, 1986.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE Computer Society, 2016.

[27] L. Hu, C. Peale, and O. Reingold. Metric entropy duality and the sample complexity of outcome indistinguishability. In *Proceedings of the Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 515–552. PMLR, 2022.

[28] A. Juditsky, A. Nemirovski, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[29] A. Kar, A. Prakash, M.-Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler. Meta-Sim: learning to generate synthetic datasets. In *Proceedings of the International Conference on Computer Vision*, pages 4550–4559. IEEE, 2019.

[30] R. M. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 881–890. SIAM, 2007.

[31] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence, 2016.

[32] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: strategies for improving communication efficiency, 2016.

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

[34] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the International Conference on Computer Vision*, pages 3730–3738. IEEE Computer Society, 2015.

[35] S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems 24*, 2011.

[36] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain Adaptation with Multiple Sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1041–1048. Curran Associates, Inc., 2008.

[37] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the International Conference on Multimedia*, pages 1485–1488. ACM, 2010.

[38] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. Communication-efficient learning of deep networks from decentralized data. In A. Singh and J. Zhu, editors, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[39] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 2019.

[40] A. S. Nemirovskij and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[41] G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3168–3176, 2015.

[42] H. L. Nguyen and L. Zakynthinou. Improved algorithms for collaborative PAC learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7642–7650. Curran Associates, Inc., 2018.

[43] B. Peng. The sample complexity of multi-distribution learning, 2023.

[44] M. G. Rabbat and R. D. Nowak. Quantized incremental algorithms for distributed optimization. *IEEE Journal on Selected Areas in Communications*, 23(4):798–808, 2005.

[45] A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3066–3074, 2013.

[46] V. V. Ramaswamy, S. S. Kim, and O. Russakovsky. Fair attribute classification through latent space de-biasing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9301–9310, 2021.

[47] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[48] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951.

[49] G. N. Rothblum and G. Yona. Multi-group agnostic PAC learnability. In M. Meila and T. Zhang, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 9107–9115. PMLR, 2021.

[50] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview, 2020.

[51] S. Sagawa, A. Raghunathan, P. W. Koh, and P. Liang. An investigation of why overparameterization exacerbates spurious correlations. In H. Daumé III and A. Singh, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR, 2020.

[52] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In E. P. Xing and T. Jebara, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 32 of *Proceedings of Machine Learning Research*, pages 1000–1008. PMLR, 2014.

[53] C. J. Tosh and D. Hsu. Simple and near-optimal algorithms for hidden stratification and multi-group learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 21633–21657. PMLR, 2022.

[54] L. G. Valiant. A theory of the learnable. In *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 436–445. ACM, 1984.

[55] N. K. Vishnoi. *Algorithms for Convex Optimization*. Cambridge University Press, 2021.

[56] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[57] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1112–1122. Association for Computational Linguistics, 2018.

[58] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and others. Huggingface's transformers: state-of-the-art natural language processing, 2019.

[59] S. Zakharov, W. Kehl, and S. Ilic. DeceptionNet: network-driven domain randomization. In *Proceedings of the International Conference on Computer Vision*, pages 532–541. IEEE, 2019.

[60] C. Zhang. Information-theoretic lower bounds of PAC sample complexity, 2019.

[61] L. Zhang, P. Zhao, T. Yang, and Z. Zhou. Stochastic approximation approaches to group distributionally robust optimization, 2023.

[62] Z. Zhang, W. Zhan, Y. Chen, S. S. Du, and J. D. Lee. Optimal multi-distribution learning, 2023.

Algorithm 3 Resampling-based Multi-Distribution Learning (R-MDL)

Input: Parameter space Θ, iterations T, batch size B and adversary batch size B', and training and validation datasets $X_{\mathrm{tr},i}$ and $X_{\mathrm{val},i}$ for $i \in [n]$;
Initialize: $\theta^{(0)} \in \Theta$ and $w^{(0)} = [1/n]^n$;
for $t = 1, 2, \ldots, T$ **do**
For $i \in [n]$, randomly sample (with replacement) B' datapoints $x^{(t-1)}_{\mathrm{val},i,1}, \ldots, x^{(t-1)}_{\mathrm{val},i,B'}$ from $X_{\mathrm{val},i}$;
Let $w^{(t)} = \mathrm{Hedge}_{\Delta_n}\left(\left\{w \mapsto 1 - \frac{1}{B'}\sum_{j=1}^{B'}\sum_{i=1}^{n} w_i \ell(h_{\theta^{(\tau)}}, x^{(\tau)}_{\mathrm{val},i,j})\right\}_{\tau\in[t-1]}\right)$, see Equation 4;
Randomly sample (with replacement) the datapoints $x^{(t-1)}_{\mathrm{tr},1}, \ldots, x^{(t-1)}_{\mathrm{tr},B}$ from $\sum_{i=1}^{n} w_i^{(t-1)} D_i$;
Run a gradient descent update(s) $\theta^{(t)} = \mathrm{GradientDescent}_{\Theta}\left(\theta \mapsto \frac{1}{B}\sum_{j=1}^{B} \ell(h_\theta, x^{(\tau)}_{\mathrm{tr},j})\right)_{\tau\in[t-1]}$;
end for
Return: $\frac{1}{T}\sum_{t=1}^{T}\theta^{(t)}$;

A Experiment Details

R-MDL Algorithm. The R-MDL algorithm is defined in full in Algorithm 3. It instantiates (a batched version of) Algorithm 2 choosing $\mathcal{Q}_\Theta$ to be online gradient descent and $\mathcal{Q}_{\Delta_n}$ to be a naive bandit-to-full-information reduction algorithm that implements Exp3 but observes cost functions uniformly at random and re-uses cost function observations between rounds. This algorithm is an example of instantiating our general multi-distribution learning framework with more practical choices of learning algorithms. An example implementation, along with experiment replications, is provided in the Github repository ericzhao28/multidistributionlearning.

Additional Observation: R-MDL converges faster than ERM or GDRO. The R-MDL methods in Table 2 used a fraction of the training epochs that their GDRO counterparts used. The ratio of R-MDL to GDRO training epochs is 1:3, 2:5, 1:2 on the Waterbirds, CelebA, and MultiNLI datasets respectively. This fast convergence rate is predicted by our theory, particularly Corollary 6.2. In our Figure 1, we also replicate the Figure 2 of Sagawa et al. [50], appending our additional results on R-MDL. We again see a trend of faster test error convergence (solid lines) and more uniform per-group risks by the R-MDL algorithm.

Datasets. Our experiments were performed on three datasets: Multi-NLI, CelebA, and Waterbirds [50]. We use identical preprocessing settings and dataset splits as Sagawa et al. [50]. Our experiments, unless otherwise specified, replicate the exact hyperparameter settings adopted by Sagawa et al. [50] for their Table 2 experiments. This includes the choice of random seeds, batch sizes, learning rates, learning schedules, and regularization. We defer readers to Sagawa et al. [50] or to our public source code for replication details.

The **Multi-NLI dataset** [57] concerns the following natural language inference task: determine if one statement is entailed by, neutral with, or contradicts a given statement. This dataset is challenging because traditional ERM models are prone to spuriously correlating "contradiction" labels with the existence of negation words. The dataset is divided into 6 distributions: the Cartesian product of the label space (entailment, neutral, contradiction) and an indicator of whether the sentence contains a negation word. The label space annotations were annotated by [57] while negation labels were annotated by Sagawa et al. [50]. There are 206,175 datapoints available in the Multi-NLI dataset; the smallest distribution (entailment + negation) is represented by only 1,521 datapoints. We use a randomly shuffled 50-20-30 training-validation-testing split.

The **CelebA dataset** is a dataset of celebrity face images and a label space of potential physical attributes [34]. This dataset is challenging because traditional ERM models are prone to spuriously correlating attribute labels with demographic information such as race and gender. Following Sagawa et al. [50], we divide the dataset into 4 distributions: the Cartesian product of the blond vs dark hair attribute label ("Blond_Hair") with the "gender" attribute label ("Male"). Note that the authors of Liu et al. [34] limited the "gender" attribute label to binary options of male and not male. There are 162,770 datapoints available in

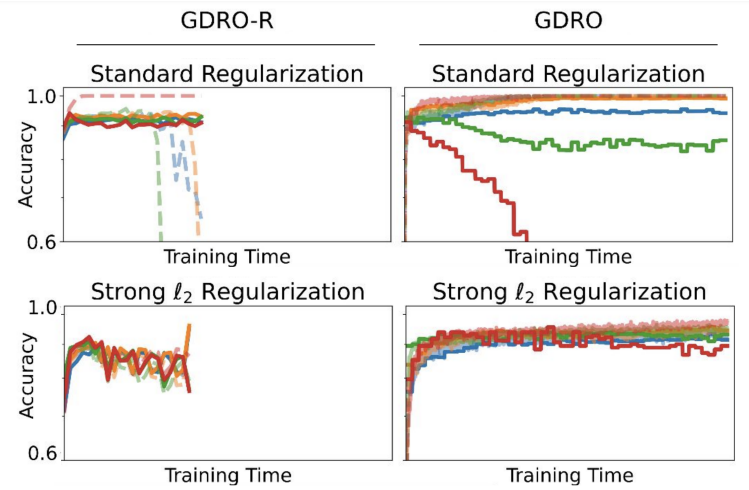

Figure 1: Training (light, dashed) and validation (dark, solid) accuracies for GDRO and R-MDL during training, plotted on a log scale. Note that R-MDL validation accuracy will be noisier than those of GDRO as we constrain R-MDL to limited samples (with replacement) from the validation set. In addition, in the left-most plot, training accuracy for all groups except the blond male group (red) dips to zero due to lack of data—this is because the blond male group (red) is the most challenging so the adversary eventually stops sampling from other groups. Under standard regularization, the red-group accuracy drops off in GDRO while R-MDL maintains a high red-group accuracy by heavily sampling from the red group, as reflected in the near-perfect red-group training error.

the CelebA dataset; the smallest distribution (blond-hair + male) is represented by only 1,387 datapoints. We use the official training-testing-validation dataset split.

The **Waterbirds dataset** is a dataset by Sagawa et al. [50] curated from a larger Caltech-UCSD Birds-200-2011 (CUB) dataset [56]. It concerns the task of predicting whether a bird is of some waterbird (sub)species from an image of said bird. This dataset is challenging because traditional ERM models are prone to spuriously correlating backgrounds with foreground subjects; for instance, a model may often predict that a bird is a waterbird only because the image of the bird was taken at a beach. The dataset has 4 distributions: the Cartesian product of the waterbird vs not waterbird label with whether the background of the picture is over water. There are 4,795 datapoints available in the Waterbirds dataset; the smallest distribution (waterbirds on land) is represented by only 56 examples.

Models. We use two classes of models in our experiments: Resnet-50 [26] and BERT [17]. We use the *torchvision* [37] implementation of the convolutional neural network Resnet-50, with a default choice of a stochastic gradient descent optimizer with momentum 0.9 and batch size 128. Batch normalization is used; data augmentation and dropout are not used. We use the *HuggingFace* [58] implementation of the language model BERT, with a default choice of an Adam optimizer with dropout and batch size 32.

Hyperparameters. In the *Standard Regularization* experiments, we use a Resnet-50 model with an ℓ-2 regularization parameter of $\lambda = 0.0001$ and a fixed learning rate of $\alpha = 0.001$ for both Waterbirds and CelebA datasets. The ERM and Group DRO baselines are trained on CelebA for 50 epochs and Waterbirds for 300 epochs. Our multi-distribution learning method is trained on CelebA for only 20 epochs and Waterbirds for 100 epochs; this is due to the faster training error convergence of our method. For the MultiNLI dataset, we use a BERT model with a linearly decaying learning rate starting at $\alpha_0 = 0.00002$ and no ℓ-2 regularization. The ERM and Group DRO baselines are trained on Multi-NLI for 20 epochs. Our multi-distribution learning method is trained on Multi-NLI for only 10 epochs. Our multi-distribution learning method uses adversary learning rates η_+ of 1, 1, 0.2 on Waterbirds, CelebA and MultiNLI respectively.

In the *Strong Regularization* experiments, we follow similar settings to the *Standard Regularization* experiments. The only change is that an ℓ-2 regularization parameter of $\lambda = 1$ is used for Waterbirds and

an ℓ-2 regularization parameter of $\lambda = 0.1$ is used for CelebA. Our multi-distribution learning method uses adversary learning rates η_+ of 1 and 0.2 on Waterbirds and CelebA respectively.

In the *Early Stopping* experiments, we follow similar settings to the *Standard Regularization* experiments. The only change is that all CelebA and Waterbird experiments are run for a single epoch. MultiNLI experiments are run for 3 epochs. Our multi-distribution learning method uses adversary learning rates η_+ of 1, 1, 1 on Waterbirds, CelebA and MultiNLI respectively.

The only hyperparameters we use that differ from prior literature are the number of training epochs and the adversary learning rates of our method (R-MDL). The choice of epoch was not fine-tuned, and was selected due to our observation of early training error convergence. We selected our adversary learning rate η_- by training our method, on each dataset, for both $\eta_- = 1$ and $\eta_- = 0.2$ and selecting the η_- yielding the highest validation-split worst-group accuracy.

Compute. The total amount of compute run for the experiments in this section is approximately 50 GPU hours. A "n1-standard-8" machine was leased from the Cloud computing service Google Cloud; the "n1-standard-8" machine was equipped with 8 Intel Broadwell chips and 1 NVIDIA Tesla V100 GPU. The cost of these computing resources totaled approximately USD $2 per hour, with a total cost of approximately USD $100. All results described in this section, with the exception of existing results cited from other works, were obtained with experiments on said machine. All experiments were implemented in Python and PyTorch.