On-Demand Sampling:
Learning Optimally from Multiple Distributions [*]

Nika Haghtalab, Michael I. Jordan, and Eric Zhao

University of California, Berkeley

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* [4], *group distributionally robust* [27], and *fair federated* learning [20]. In each of these settings, a learner seeks to minimize its worst-case loss over a set of n predefined 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 the sample complexity of learning a single distribution only by an additive factor of $\frac{n\log(n)}{\varepsilon^2}$. These improve upon the best known sample complexity of agnostic federated learning by Mohri et al. [20] by a multiplicative factor of n, the sample complexity of collaborative learning by Nguyen and Zakynthinou [22] by a multiplicative factor $\frac{\log n}{\varepsilon^3}$, and give the first sample complexity bounds for the *group DRO* objective of Sagawa et al. [27]. To achieve optimal sample complexity, our algorithms learn to sample and learn from distributions on demand. Our algorithm design and analysis is enabled by our extensions of stochastic optimization techniques for solving stochastic zero-sum games. In particular, we contribute variants of Stochastic Mirror Descent that can trade off between players' access to cheap one-off samples or more expensive reusable ones.

1 Introduction

Pervasive needs for robustness, fairness, and multi-agent collaboration in learning have given rise to multi-distribution learning paradigms (e.g., [4, 27, 20, 10]). In these settings, we seek to learn a model that performs well on *any distribution* in a pre-defined set of interest. For fairness considerations, these distributions may represent heterogeneous populations of different protected or socio-economic attributes; in robustness applications, they may capture a learner's uncertainty regarding the true underlying task; and in muti-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, learning tasks of varying difficulty require different numbers of samples. 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 distribution* they want them. On-demand sampling is especially appropriate when some population data may be scarce to start with (as in fairness mechanisms in which samples are amended [24]); when the designer can actively perturb datasets towards rare or atypical instances (such as in robustness applications [16, 34]); or when sample sets represent agents' contributions to an interactive multi-agent system [20, 5].

[*] Authors' addresses: nika@berkeley.edu, jordan@cs.berkeley.edu, eric.zh@berkely.edu. Authors are ordered alphabetically.

Problem	Sample Complexity	Thm	Best Previous Result
Collab. Learning UB	$\varepsilon^{-2}\left(\log\lvert\mathcal{H}\rvert + n\log(\frac{n}{\delta})\right)$	[4.1]	$\varepsilon^{-5}\log\left(\frac{1}{\varepsilon}\right)\log(\frac{n}{\delta})(\log\lvert\mathcal{H}\rvert + n)$ [22]
Collab. Learning LB	$\varepsilon^{-2}(\log\lvert\mathcal{H}\rvert + n\log(\frac{k}{\delta}))$	[4.3]	$\varepsilon^{-1}n\log(k/\delta)$ [4]
GDRO/AFL UB	$\varepsilon^{-2}\left(\log\lvert\mathcal{H}\rvert + n\log(\frac{n}{\delta})\right)$	[4.1]	$\varepsilon^{-2}\left(n\log\lvert\mathcal{H}\rvert + n\log(\frac{n}{\delta})\right)$ [20]
GDRO/AFL UB	$\varepsilon^{-2}\left(D_{\mathcal{H}} + n\log(\frac{n}{\delta})\right)$	[5.1]	N/A
(Training error convg.)	$\varepsilon^{-2}\left(D_{\mathcal{H}} + n\log(\frac{n}{\delta})\right)$	[5.2]	$\varepsilon^{-2}D_{\mathcal{H}}$ (expected convergence only) [27]

Table 1: This table gives upper (UB) and lower bounds (LB) on the sample complexity of learning model class H on n distributions. For the collaborative learning and AFL settings, the sample complexity upper bounds refer to the problem of learning a randomized model of worst-case error $\mathrm{OPT} + \varepsilon$ or a deterministic classifier of worst-case error $2\mathrm{OPT} + \varepsilon$. For the GDRO setting, sample complexity refers to learning a deterministic model with worst-case error of $\text{R-OPT} + \varepsilon$, where R-OPT is the best worst-case error attainable in a convex compact model space H. $D_{\mathcal{H}}$ denotes the Bregman radius of H, and $k = \min\{n, \log\lvert\mathcal{H}\rvert\}$. Sample complexity bounds of 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. [4] demonstrated the benefit of on-demand sampling in the *collaborative learning* setting, where all data distributions are realizable with respect to the same target classifier. This line of work established that learning n distributions on-demand takes $\widetilde{O}(\log(n))$ times the sample complexity of learning a single realizable distribution [4, 6, 22], whereas relying on batched uniform convergence takes $\widetilde{\Omega}(n)$ times that of learning a single distribution [4]. 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 $\widetilde{O}\left(\log(n)/\varepsilon^3\right)$ times the sample complexity of agnostically learning a single distribution [22]. On the other hand, agnostic federated learning bounds [20] have been studied only on algorithms that sample in one large batch and thus require $\widetilde{\Omega}(n)$ times the sample complexity of learning a single task. Moreover, the test-time performance of some key multi-distribution methods, such as group distributionally robust optimization [27], have not been studied from a theoretical 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 settings consider a set $\mathcal{D}$ of n distributions and a model class $\mathcal{H}$. They evaluate the performance of a model h (or a distribution over models) by its worst-case performance, $\max_{D\in\mathcal{D}} \mathrm{loss}_D(h)$. As a benchmark, they consider the worst-case loss of the best model, i.e., $\mathrm{OPT} = \min_{h^*\in\mathcal{H}}\max_{D\in\mathcal{D}} \mathrm{loss}_D(h^*)$. Importantly, all of our sample complexity upper bounds demonstrate only an *additive increase of* $\varepsilon^{-2}n\log(n/\delta)$ *over the sample complexity of learning a single task*, compared to the multiplicative factor increase required by existing works.

- *Collaborative learning of Blum et al. [4]:* For agnostic collaborative learning, our Theorem 4.1 gives a randomized and a deterministic model that achieve performance guarantees of $\mathrm{OPT} + \varepsilon$ and $2\mathrm{OPT} + \varepsilon$, respectively. Our algorithms have an optimal sample complexity of $O\left(\frac{1}{\varepsilon^2}(\log(\lvert H\rvert) + n\log(\frac{n}{\delta}))\right)$. This improves upon the work of Nguyen and Zakynthinou [22] in two ways. First, it provides error bounds of $\mathrm{OPT} + \varepsilon$ for randomized classifiers, where only $2\mathrm{OPT} + \varepsilon$ was previously established. Second, it improves the upper bound of Nguyen and Zakynthinou [22] by a multiplicative factor of $\log(n)/\varepsilon^3$. In Theorem 4.3, 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. [27]:* For group DRO, we consider a convex and compact model space $\mathcal{H}$. Our Theorem 5.1 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 $\mathcal{O}\left(\frac{1}{\varepsilon^2}(D_H + n\log(\frac{n}{\delta}))\right)$. 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. [27] by a factor of n.

- *Agnostic federated learning of [20]:* For agnostic federated learning, we consider a finite class of hypotheses. Our Theorems 4.1 and 5.1 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. [20]. Our results also imply matching high-probability bounds to Mohri et al. [20] on the convergence of the training error in the batched setting.

To achieve these results, we contribute new insights and techniques for solving stochastic zero-sum games with sources of randomization that differ in both cost and quality. We frame the multi-distribution learning problems as a stochastic zero-sum game with uncertain payoffs and utilize stochastic mirror descent and a variational perspective to solve the game. In this case, the maximizing player can be interpreted as a weight vector for distributions $\mathcal{D}$, specifying from which distributions future on-demand samples should be taken. These on-demand samples form a stochastic gradient for the players. However, the quality of these estimators, the number of samples needed for them, and whether they can be reused later on, differs between the two players. We extend the Stochastic Mirror Descent framework to optimally trade off these asymmetric needs for samples. In Section 3 we give an overview of this approach and its technical challenges and contributions.

1.1 Related Work

Learning models. Three independent lines of work study multi-distribution learning, with different motivating applications. *Collaborative learning* interprets multiple distributions as *players* that each seek to learn a model with low error on their data distributions [4, 22, 6]. *Agnostic federated learning* interprets these distributions as *clients* in a federated learning system [20]. *Group distributionally robust optimization* interprets these distributions as data attributes or sources that a learner should avoid linking spuriously to labels [14, 27, 28]. Formally, these learning objectives are all equivalent but have been studied from different points of view and with different technical tools.

Existing work on group DRO has assumed that data is pre-collected and has studied the convergence of multi-distribution training error. The agnostic federated learning literature has studied a single-batch approach and derived data-dependent generalization bounds that suggest how much of the batch should be collected from each distribution. Finally, the collaborative learning literature has studied an on-demand framework for collecting data from each distribution. This approach also relates to a line of work on multi-source learning and domain adaptation [3, 18].

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 [26, 12, 23, 7, 8]. One such method, online mirror descent (OMD), can also approximate minima of convex functions with high probability using noisy first-order information [25, 21, 2]. This allows OMD to efficiently find min-max equilibria even in stochastic convex-concave zero-sum games [13]. We bring these online learning tools to bear on the problem of finding equilibria in robust optimization formulations.

2 Preliminaries

Let $\mathcal{X}$ be an instance space, $\mathcal{Y}$ a label space, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ a space of datapoints. A data distribution D is a joint probability distribution over $\mathcal{Z}$. We consider a hypothesis class $\mathcal{H}$ of a subset of functions mapping $\mathcal{X}$ to $\mathcal{Y}$. We work with loss functions $\ell : \mathcal{H} \times \mathcal{Z} \to [0, 1]$ that measure the loss of hypothesis h on data point $z \in \mathcal{Z}$. When $\mathcal{Y} = \{0, 1\}$, ℓ is the misclassification error. We denote the expected loss, i.e. risk, of a hypothesis $h \in \mathcal{H}$ under a data distribution $D \in \mathcal{D}$ by:

$$\mathrm{Risk}_D(h) \coloneqq \mathop{\mathbb{E}}_{(x,y)\sim D} \left[\ell\left(h, (x, y)\right)\right].$$

For a distribution over the hypothesis class, $p \in \Delta\mathcal{H}$, and a distribution over data distributions, $q \in \Delta\mathcal{D}$, we refer to their expected loss by $\mathrm{Risk}_q(p) \coloneqq \mathbb{E}_{D\sim q}\left[\mathbb{E}_{h\sim p}\left[\mathrm{Risk}_D(h)\right]\right]$.

Collaborative Learning. We will use the *collaborative PAC learning model* of Blum et al. [4] and its agnostic extensions by Nguyen and Zakynthinou [22]. The overall goal of this setting is to guarantee small risk for *every* distribution in a collection of distributions. Formally, we consider a set of data distributions $\mathcal{D} \coloneqq \{D_1, \ldots, D_n\}$. The goal of the learner is to learn a hypothesis h such that, with probability $1 - \delta$,

$$\max_{D \in \mathcal{D}} \operatorname{Risk}_D(h) \leq \mathrm{OPT} + \varepsilon, \text{ where } \mathrm{OPT} \coloneqq \min_{h \in \mathcal{H}} \max_{D \in \mathcal{D}} \operatorname{Risk}_D(h). \tag{1}$$

Group Distribution Robustness. We will also study the closely related setting of *group distributionally robust optimization (Group DRO)* of Sagawa et al. [27]. Formally, the group DRO setting considers a model set Θ that is a convex compact subset of the Euclidean space and a convex loss function $\ell : \Theta \times \mathcal{Z} \to [0, 1]$ that is assumed to be differentiable over Θ. Given a set of data distributions $\mathcal{D} \coloneqq \{D_1, \ldots, D_n\}$, the learner seeks a model $\theta \in \Theta$, such that, with probability $1 - \delta$,

$$\max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{(x,y) \sim D} [\ell(\theta, (x, y))] \leq \text{R-OPT} + \varepsilon, \text{ where R-OPT} \coloneqq \min_{\theta \in \Theta} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{(x,y) \sim D} [\ell(\theta, (x, y))]. \tag{2}$$

There is a close relationship between the Group DRO setting and collaborative learning. In particular, when $\Theta = \Delta(\mathcal{H})$ and $\mathcal{H}$ is finite, the two goals are analogous but with two exceptions: first, the Group DRO could return a distribution over functions while collaborative learning requires the solution to be a deterministic function, and second, allowing for randomized hypothesis leads to R-OPT being potentially more competitive than OPT. We note that the group DRO setting is equivalent to the agnostic federated learning framework of [20], thus our results for DRO extend to that setting as well.

Sample complexity. We are interested in the design of algorithms that achieve the above goals while using smallest number of samples from distributions $D_1, \ldots, D_n$. We formalize the sample complexity by the total number of calls made to *example oracles* $\mathrm{EX}(D_i)$. Each call $\mathrm{EX}(D)$ produces an i.i.d. sample from D. We note that these example oracles also allow us to sample from any mixture distribution $q \in \Delta\mathcal{D}$, e.g., by first selecting a D_i according to the mixture and then calling $\mathrm{EX}(D_i)$.

2.1 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 finite two-player zero-sum game is described by the tuple (A_{-}, A_{+}, ϕ) where $A_{-} = \{1, \ldots, n\}$ and $A_{+} = \{1, \ldots, m\}$ are finite sets of actions and where $\phi : A_{-} \times A_{+} \to [0, C]$. In this game, the players choose *mixed strategies* over actions sets. These are distributions that are denoted by a vector of probabilities $p \in \Delta A_{-}$ and $q \in \Delta A_{+}$. The expected payoff of mixed strategies is denoted by $\phi(p, q) = \mathbb{E}_{i \sim p, j \sim q} [\phi(i, j)]$. The goal of the minimizing player is to minimize this expected payoff and the maximizer seeks to maximize the expected payoff; that is, to solve

$$\min_{p \in \Delta A_{-}} \max_{q \in \Delta A_{+}} \phi(p, q).$$

A pair (p, q) that solves this optimization problem is called a *min-max equilibrium.* Similarly, a solution is called an ε*-min-max* equilibrium if neither player can unilaterally improve their objective by more than ε. Formally, (p, q) is an ε-min-max equilibrium if both players' regrets are at most ε, i.e., $\text{Reg-Min}(p, q) \coloneqq \phi(p, q) - \min_{i^* \in A_{-}} \phi(i^*, q) \leq \varepsilon$ and $\text{Reg-Max}(p, q) \coloneqq \max_{j^* \in A_{+}} \phi(p, j^*) - \phi(p, q) \leq \varepsilon$. We will next describe methods that find ε-min-max equilibria by finding solutions (p, q) for which $\text{Reg-Min}(p, q) + \text{Reg-Max}(p, q)$ is at most ε. We describe a more general formulation for convex-concave zero-sum games in Appendix A.1 which we will use for the Group DRO problem.

No-Regret Learning. We consider an online setting where an arbitrary set of *operators*, $g^{(1)}, \ldots g^{(T)} \in \mathcal{E}^*$, is revealed sequentially to a learner who must choose a matching sequence of actions, $w^{(1)}, \ldots w^{(T)}$, from a convex compact set $Z \subseteq \mathcal{E}$. Here, $\mathcal{E}$ and $\mathcal{E}^*$ respectively refer to an arbitrary Euclidean space and its dual.

We focus on a setting where an online learner commits to action $w^{(t)} \in Z$ before seeing $g^{(t)}, g^{(t+1)}, \dots$ and aims to achieve vanishing *variational error* $\mathrm{Err}_{\mathrm{V}}(w^{(1:T)})$ defined by

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) := \max_{w^* \in Z} \frac{1}{T} \sum_{t=1}^{T} \left\langle g^{(t)}, w^{(t)} - w^* \right\rangle . \tag{3}$$

We will denote no-regret algorithms by their update rule $\mathcal{Q} : \{Z \times \mathcal{E}^*\} \to Z$, where $\{Z \times \mathcal{E}^*\}$ denotes the space of arbitrary length sequences of action-operator pairs. Given a history sequence $w^{(1)}, \dots, w^{(t)} \in Z$ and operator sequence $g^{(1)}, \dots, g^{(t)} \in \mathcal{E}^*$, the algorithm returns $w^{(t+1)} = \mathcal{Q}\left(\left\{w^{(1)}, g^{(1)}\right\}, \dots, \left\{w^{(t)}, g^{(t)}\right\}\right)$. When the history is clear from context, we write $w^{(t+1)} = \mathcal{Q}\left(w^{(t)}, g^{(t)}\right)$ as shorthand. For the particular case where $Z = \Delta^n$ is a probability simplex, one such algorithm is Exponential Gradient Descent (also known as Hedge):

$$\mathcal{Q}_{\mathrm{hedge}}\left(\left\{w^{(1)}, g^{(1)}\right\}, \dots, \left\{w^{(t)}, g^{(t)}\right\}\right) := \frac{\widetilde{w}}{\|\widetilde{w}\|_1} \text{ where } \widetilde{w}_i := w_i^{(t)} \exp\left\{-\eta g_i^{(t)}\right\}, \tag{4}$$

where η is a user-defined step size, and w_1 is a user-defined initial iterate. By default, we take $w_1 = \left[\frac{1}{n}\right]^n$. The following lemma is a classical result on the variational error of exponential gradient descent.

Lemma 2.1 ([30]). *Let $g^{(1)}, \dots, g^{(T)} \in \mathbb{R}^n$ and $Z = \Delta^n$. Further assume $\left\|g^{(t)}\right\|_\infty \le C$ for all timesteps $t = 1, \dots, T$. Choosing $\eta = \sqrt{\log n / T}$, after T iterations of exponential gradient descent, the output $\{w\}_{t=1}^T$ satisfies,*

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \le \frac{3C}{2} \sqrt{\frac{KL\left(w^{(T)} || w^{(1)}\right)}{T}}.$$

3 Technical Overview of Our Approach

In this section, we provide an overview of our technical approach for addressing the sample complexity of collaborative learning and group DRO problems. In later sections, we will refer to the approach outlined in this section to sketch proofs and design algorithms. We will focus our exposition on collaborative learning and briefly indicate how the same approach applies to the group DRO setting.

At a high level, we first frame collaborative learning as a zero-sum game with uncertain payoffs and aim to use a variational perspective to learn its minmax equilibrium. We specifically choose the variational perspective (instead of an arbitrary online learning approach), since it allows us to linearize the effect of uncertain payoffs on the resulting error. We then use stochastic gradients to solve the variational problem. Our stochastic gradients will rely on i.i.d. samples from the distributions to estimate gradients both with respect to distributions over $\mathcal{H}$ and mixtures over $\mathcal{D}$ but with an asymmetric bound on the bias and variance of the estimates. Along the way, we develop tools and formalisms that handle the asymmetric cost of stochastic gradients and obtain optimal sample complexity results. We now address these steps in more detail.

Collaborative Learning as Zero-Sum Games. When the hypothesis class $\mathcal{H}$ is finite, the collaborative learning problem with distribution set $\mathcal{D}$ corresponds to a zero-sum game (A_-, A_+, ϕ) with $A_- = \mathcal{H}, A_+ = \mathcal{D}, \phi(i, j) = \mathrm{Risk}_j(i)$, such that the value of the min-max solution is equivalent to R-OPT. It is not hard to see that any ε-min-max equilibrium (p, q) of this game corresponds to a 2ε collaborative learning solution, i.e.,

$$\mathop{\mathbb{E}}_{h \sim p}\left[\max_{D \in \mathcal{D}} \mathrm{Risk}_D(h)\right] \le \mathrm{OPT} + 2\varepsilon. \tag{5}$$

This enables us to use tools that have been developed for solving zero-sum games in order to address collaborative learning and group DRO settings. We will use a similar construction when hypothesis class $\mathcal{H}$ has finite VC dimension, where A_- will instead refer to an appropriate ε-cover of $\mathcal{H}$.

Using VI to deal with Payoff Uncertainty. A sufficient condition for minimizing regret, and thus finding ε-min-max equilibrium, is minimizing the variational error (Equation 3). In particular, for any finite zero-sum game (A_{-}, A_{+}, ϕ), defining $Z = [\Delta A_{-}, \Delta A_{+}]$ and operators

$$g^{(t)} = \left[\left\{\partial_{p_i}\phi(p^{(t)}, q^{(t)})\right\}_{i\in A_{-}}, \left\{-\partial_{q_j}\phi(p^{(t)}, q^{(t)})\right\}_{j\in A_{+}}\right], \tag{6}$$

ensures that variational error provides an upper bound on regret: $\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \geq \text{Reg-Min}(p, q) + \text{Reg-Max}(p, q)$, where $w = (p, q)$ (see Fact B.1). In collaborative learning, when $p^{(t)}$ is the min-player's distribution over hypotheses and $q^{(t)}$ is max-player's distribution over the mixtures, the gradient vectors refer to

$$g^{(t)} = [g_{-}^{(t)}, g_{+}^{(t)}], \quad g_{-}^{(t)} = \left\{\mathrm{Risk}_{q^{(t)}}(h)\right\}_{h\in\mathcal{H}}, \quad g_{+}^{(t)} = \left\{\mathrm{Risk}_{D}(p^{(t)})\right\}_{D\in\mathcal{D}}. \tag{7}$$

In the collaborative learning setting, we can only create noisy estimates $\widehat{g}$ for these gradients from samples. This is where no-regret algorithms that minimize variational error become advantageous. By linearizing the effect of noise, $\varepsilon^{(t)} := g^{(t)} - \widehat{g}^{(t)}$, they decompose the variational error into the *training* and *generalization* error as follows

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \leq \max_{w^*\in\Delta^n} \frac{1}{T}\sum_{t=1}^{T}\left\langle \widehat{g}^{(t)}, w^{(t)} - w^*\right\rangle + \max_{w^*\in\Delta^n} \frac{1}{T}\sum_{t=1}^{T}\left\langle \varepsilon^{(t)}, w^{(t)} - w^*\right\rangle. \tag{8}$$

In contrast, generic no-regret algorithms that do not solve the variational inequality (e.g., when one player plays Hedge and another plays clairvoyant best-response as used in existing work in collaborative learning due to Blum et al. [4], Nguyen and Zakynthinou [22], Chen et al. [6]) nest the generalization and training errors which leads to a multiplicative increase in sample complexity.

Leveraging Noisy Stochastic Gradients. We will work with stochastic estimators of g. These are functions $\widehat{g} : \boldsymbol{\xi} \times \Delta A_{-} \times \Delta A_{+}$ of some external source of randomness, $\xi \in \boldsymbol{\xi}$, and a strategy profile of interest. For collaborative learning, the randomness source ξ is an i.i.d.-sampled data point from an appropriate mixture of distributions and the estimator $\widehat{g}$ is then the empirical loss on this sample, which is an unbiased and bounded estimator in the range of the loss function, i.e., $[0, 1]$.

Interestingly, estimators of these stochastic gradients have an asymmetric need for data. As seen in Equation 7, the min-player's gradient $g_{-}(p, q)$ includes the risk of every hypothesis $h \in \mathcal{H}$ for the same data distribution q. Therefore, an unbiased estimator $\widehat{g}_{-}(p, q)$ can be constructed from a single call to an example oracle $\mathrm{EX}(q)$. We call this source of randomness ξ^q and say that its cost is $r_{-} = 1$. While ξ^q costs 1 unit, the randomness it provides is specialized to the point of inquiry, that is, it cannot be used for estimating other $\widehat{g}_{-}(p, q')$. We call this source of randomness and its associated unbiased estimation a *locally* unbiased estimator.

On the other hand, the max-player's gradient $g_{+}(p, q)$ includes the risk of the same hypothesis p on *every distribution* $D \in \mathcal{D}$. Therefore, an unbiased estimator $\widehat{g}_{+}(p, q)$ *requires n samples,* i.e., a call to every example oracle $\mathrm{EX}(D_i)$. We call this source of randomness producing n samples ξ^p and say that its cost is $r_{+} = n$. Importantly, while ξ^p costs n unit, the randomness it provides can be reused for estimating other gradients, that is, it can provide an unbiased and bounded estimators for all $\widehat{g}_{+}(p', q')$. We call this source of randomness and its associated unbiased estimator a *globally* unbiased estimator. To emphasize the fact that this source of randomness is agnostic to (p, q) we refer to it by $\xi^{\perp}$ hereafter. We refer the reader to Appendix A.2 for a more formal definition and description of these asymmetries.

Minimizing Regret with Asymmetric Cost. With the goal of minimizing sample complexity in mind, it is essential that we reuse randomness $\xi^{\perp}$ across n time steps of variational algorithms. To do this, we introduce a stochastic variational approach in Algorithm 1 that accommodates different sampling frequencies for the minimizing and maximizing players. This will decouple the sample complexity of the minimizing agent (who requires a time horizon of at least $\log(A_{-}) \approx \log(\mathcal{H})$) and the maximizing agent. This decoupling will lead to additive $n + \log(\mathcal{H})$ sample complexity instead of the multiplicative $n \log(\mathcal{H})$.

Algorithm 1 uses the same randomness $\xi^{\perp(a)}$ of cost r for estimating $g_{+}(p^t, q^t)$ for all $t \in [ar{+}1, \ldots, a(r{+}1)]$. On the other hand, the algorithm uses fresh randomness $\xi^{(t)}$ of cost 1 to estimate $g_{-}(p^t, q^t)$ for every time

Algorithm 1 Finding Equilibria in Finite Zero-Sum Games with Asymmetric Costs.

```
Output: Mixed strategy profile $(p, q) \in \Delta A_{\text{-}} \times \Delta A_{+}$;
Input: Action sets $A_{\text{-}}, A_{+}$, cost $r \in \mathbb{Z}_{+}$, timesteps $T$, iterates $p^{(1)}, q^{(1)}$, gradient estimators $\widehat{g}_{\text{-}}, \widehat{g}_{+}$;
for $a = 1, 2, \ldots, \lceil T/r \rceil$ do
    Realize $\xi^{\perp^{(a)}}$ at cost $r$;                // Sample datapoints from every distribution.
    for $t = ar + 1 - r, \ldots, ar$ do
        Realize $\xi^{q^{(t)}}$ at cost 1;              // Sample from adversary-selected distribution.
        Estimate gradients: $\widehat{g}_{+}^{(t)} = \widehat{g}_{+}\left(\xi^{\perp^{(a)}}, p^{(t)}, q^{(t)}\right)$,   $\widehat{g}_{\text{-}}^{(t)} = \widehat{g}_{\text{-}}\left(\xi^{q^{(t)}}, p^{(t)}, q^{(t)}\right)$;
        Run Hedge updates: $p^{(t+1)} = \mathcal{Q}_{\text{hedge}}\left(p^{(t)}, \widehat{g}_{+}^{(t)}\right), q^{(t+1)} = \mathcal{Q}_{\text{hedge}}\left(q^{(t)}, \widehat{g}_{+}^{(t)}\right)$;
    end for
end for
Return the uniformly mixed strategies $\overline{p} = \frac{1}{T}\sum_{t=1}^{T} p^{(t)}$ and $\overline{q} = \frac{1}{T}\sum_{t=1}^{T} q^{(t)}$;
```

step t. We note that the total randomness cost of this algorithm is $2t$ because iteration of the outer loop incurs $2r$ cost.

Lemma 3.1. *Let $(A_{\text{-}}, A_{+}, \phi)$ be a finite zero-sum game. Assume there exists $\xi^{q^{(t)}}$ of cost 1 providing locally unbiased estimates $\widehat{g}_{\text{-}}(\cdot)$ and there exists $\xi^{\perp^{(a)}}$ of cost r providing globally unbiased estimates $\widehat{g}_{+}(\cdot)$. With probability $1 - \delta$, Algorithm 1 returns an ε-min-max equilibrium of the game, so long as*

$$T \geq \frac{18}{\varepsilon^2}\left(\max\left\{\frac{9\log|A_{\text{-}}|}{4}, 8\log\left(\frac{r+1}{\delta}\right)\right\} + \max\left\{\frac{9\log|A_{+}|}{4}, \frac{8r^2}{r+1}\log\left(\frac{r+1}{\delta}\right)\right\}\right). \tag{9}$$

Moreover, the total cost of randomness incurred by the algorithm is at most $2t$.

Proof sketch. Our approach uses Equation 8 to decompose the variational error into training error and generalization error. Since exponential gradient descent is known to bound the training error (as shown in Lemma B.4), it only remains to bound the generalization error (the second term in Equation 3). We note that in expectation each summand $\left\langle \varepsilon^{(t)}, w^{(t)} - w^{*} \right\rangle$ is zero. This is because $\varepsilon^{(t)} = g^{(t)} - \widehat{g}^{(t)}$ and $\widehat{g}^{(t)}$ are unbiased estimators. Therefore, the sum of these terms has an intuitive martingale interpretation and could be bounded by the Azuma-Hoeffding inequality.

There is a subtlety here, however. When we reuse the maximizing player's randomness over r rounds, we create correlations between these terms in the generalization error that cannot be directly accommodated by a martingale. The trick here is to note that these correlations are entirely contained in r-length periods. So, we can partition our sequence to r martingales and bound each one. This completes the proof. See Appendix B.1 for detailed proof of this lemma. □

Derandomization. The ε-min-max equilibria $(\overline{p}, \overline{q})$ returned by Exponentiated Gradient Descent gives a probability distribution $\overline{p}$ over the hypothesis class that achieves the collaborative learning bound. To obtain a deterministic hypothesis, we can instead work with h_p^{Maj} whose predictions are p-weighted majority votes over the hypotheses. As stated below, the error of this deterministic classifier is approximately bounded by the expected error of $\overline{p}$.

Lemma 3.2. *For any $p \in \Delta\mathcal{H}$, $\max_{D \in \mathcal{D}} \mathrm{Risk}_D(h_p^{Maj}) \leq 2\max_{D \in \mathcal{D}} \mathrm{Risk}_D(p)$.*

This lemma in particular implies that for any ε-min-max equilibria $(\overline{p}, \overline{q})$, we have

$$\max_{D \in \mathcal{D}} \mathrm{Risk}_D(h_{\overline{p}}^{Maj}) \leq 2\text{R-OPT} + 4\varepsilon \leq 2\text{OPT} + 4\varepsilon.$$

4 Collaborative Learning Bounds

In this section, we characterize the sample complexity of collaborative learning by providing tight upper and lower bounds for this problem.

Algorithm 2 On-Demand Agnostic Collaborative Learning.

```
Input: Hypothesis class $\mathcal{H}$, distribution set $\mathcal{D}$ with $n \coloneqq |\mathcal{D}|$;
Initialize: $p^{(1)} = [1/|\mathcal{H}|]^{|\mathcal{H}|}, q^{(1)} = [1/n]^n$, and iterations $T = \frac{36}{\varepsilon^2}\left(9\log\left(|\mathcal{H}|\right) + 35n\log(n/\delta)\right)$;
for $a = 1, 2, \dots, \lceil T/n \rceil$ do
    For all $D \in \mathcal{D}$, sample datapoint $z_D^a$ from EX$(D)$ .
    for $t = an + 1 - n, \dots, an$ do
        Sample $z^{(t)}$ from EX$(q^{(t)})$ and estimate $\widehat{g}_-^{(t)} = [\ell(h, z^{(t)})]_{h\in\mathcal{H}}, \widehat{g}_+^{(t)} = [\ell(p^{(t)}, z_D^a)]_{D\in\mathcal{D}}$;
        Run Hedge updates: $p^{(t+1)} = \mathcal{Q}_{\text{hedge}}\left(p^{(t)}, \widehat{g}_-^{(t)}\right), q^{(t+1)} = \mathcal{Q}_{\text{hedge}}\left(q^{(t)}, \widehat{g}_+^{(t)}\right)$;
    end for
end for
Return: probability distribution over $\mathcal{H}$ given by the uniform mixture $\frac{1}{T}\sum_{t=1}^{T} p^{(t)}$.
```

4.1 Sample Complexity Upper Bounds

We are now prepared to describe our collaborative learning algorithm and guarantees, using the tools we developed in Section 3. Algorithm 2 is a direct application of Algorithm 1 to a zero-sum game with action sets $A_- = \mathcal{H}$, $A_+ = \mathcal{D}$ and payoff $\phi(h, D) = \text{Risk}_D(h)$. Here, $\xi^{q^{(t)}}$ makes one call to EX$(q^{(t)})$ and $\xi^{\perp(a)}$ makes one call to EX(D) for each $D \in \mathcal{D}$. In other words, Algorithm 2 constructs distributions $p^{(t)} \in \Delta\mathcal{H}$ and $q^{(t)} \in \Delta\mathcal{D}$ by running the Hedge update. The gradient estimators used by Hedge are the empirical losses on a set of independent random variables. In particular, the minimizing player uses gradients $\ell(h, z^{(t)})$ for all $h \in \mathcal{H}$ for a single sample $z^{(t)} \sim \text{EX}(q^{(t)})$ and the maximizing player uses gradients $\ell(p^{(t)}, z_D^a)$ for all distributions $D \in \mathcal{D}$ where a single sample $z_D^a \sim \text{EX}(D)$ is drawn per distribution and is reused for all time steps $t \in [(a-1)n + 1, \dots, an]$.

Our main result in this section bounds the sample complexity of Algorithm 2.

Theorem 4.1. *For any finite hypothesis class $\mathcal{H}$ and unknown set of distributions $\mathcal{D}$, with probability $1 - \delta$, Algorithm 2 returns a distribution $\overline{p} \in \Delta\mathcal{H}$ such that*

$$\mathop{\mathbb{E}}_{h\sim\overline{p}}\left[\max_{D\in\mathcal{D}}(h)\right] \leq OPT + \varepsilon \quad and \quad \max_{D\in\mathcal{D}}(h_{\overline{p}}^{Maj}) \leq 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{\log|\mathcal{H}| + n\log(n/\delta)}{\varepsilon^2}\right)$.

Proof sketch. By construction, Lemma 3.1 guarantees that with probability at least $1 - \delta$, the pair $(\overline{p}, \overline{q})$ is an $\varepsilon/2$-min-max equilibrium for the corresponding zero-sum game. As shown by Equation 5, $\overline{p}$ is a randomized classifier that meets the collaborative learning objective, i.e., its expected worst-case error is $\text{OPT} + \varepsilon$. By Lemma 3.2, the corresponding deterministic classifier $h_{\overline{p}}^{Maj}$ has worst-case error of $2\text{OPT} + \varepsilon$. This bounds the error of the resulting classifier.

To bound the sample complexity, Lemma 3.1 shows that the randomness cost of Algorithm 1 is at most $2t$. Since the cost of randomness is exactly the total number of samples we take from our example oracles, the total sample complexity of Algorithm 2 is $2t \in \mathcal{O}\left(\frac{\log|\mathcal{H}| + n\log(n/\delta)}{\varepsilon^2}\right)$. □

A similar result holds for the case of infinite hypothesis classes of bounded VC dimension. In this case, one can instead run Algorithm 2 with a hypothesis class $\mathcal{H}'$ that is an ε-net with respect to every distribution in $\mathcal{D}$. We note that such ε-nets of size $n\varepsilon^{-2\text{VCD}(\mathcal{H})}$ necessarily exist (see, e.g., [1]); for example, the union of ε-nets with respect to each distribution $D \in \mathcal{D}$. When such $\mathcal{H}'$ is known in advance, we may run Algorithm 2 with $\mathcal{H}'$ and incur a sample complexity that now replaces $\log(|\mathcal{H}'|) = O\left(d\log(1/\varepsilon)\right)$ in the sample complexity of Theorem 4.1.

We remark that 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 $\mathcal{D}$. In Appendix B.5, we explore a range of assumptions that allow us to compute such an ε-net from samples, without incurring a significant increase in the sample complexity of Theorem 4.1. As an example, here we mention two such assumptions. *Assumption 1: we know the marginal distribution for all $D \in \mathcal{D}$,* or a weaker *Assumption 2: we have access to n marginal* *distributions* $P_1, \dots, P_n$ *such that for all* $x \in \mathcal{X}$, $d_i(A) \le p_i(A)\mathrm{poly}(1/\varepsilon, \mathrm{VCD}(\mathcal{H}), n)$ *for all* $A \subseteq \mathcal{X}$, *where* p_i *and* d_i *are the densities of* P_i *and* D_i*, respectively.* These assumptions allow one to construct ε-nets of small size, e.g., by projecting $\mathcal{H}$ on a sufficiently large set of random feature vectors generated from distributions P_i. We refer the reader to Appendix B.5 for more detail on how these assumptions can be used to construct ε-nets.

Theorem 4.2. *For any* $\mathcal{H}$ *of VC dimension* d *and unknown set of distributions* $\mathcal{D}$ *for which Assumption 1 or 2 is met, there is an algorithm that, with probability* $1-\delta$*, returns a distribution* $\overline{p} \in \Delta\mathcal{H}$ *with,*

$$\mathop{\mathbb{E}}_{h\sim\overline{p}}\left[\max_{D\in\mathcal{D}}(h)\right] \le OPT + \varepsilon \quad \textit{and} \quad \max_{D\in\mathcal{D}}(h^{Maj}_{\overline{p}}) \le 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{d\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.

We end this subsection with two remarks about our sample complexity upper bound.

Remark 4.1. *Theorem 4.1 improves over the best-known sample complexity for agnostic collaborative learning by Nguyen and Zakynthinou [22] in two ways. First, it provides* $OPT+\varepsilon$ *for randomized classifiers whereas Nguyen and Zakynthinou [22] gave a* $2OPT+\varepsilon$ *bound. Second, it improves their sample complexity of* $\mathcal{O}\left(\frac{1}{\varepsilon^5}\left(\log(n)\log(|\mathcal{H}|)\log\left(\frac{1}{\varepsilon}\right)+n\log\left(\frac{n}{\delta}\right)\right)\right)$ *by a multiplicative factor of* $\frac{1}{\varepsilon^3}\log(n)\log\left(\frac{1}{\varepsilon}\right)$.

Remark 4.2. *For constants* ε *and* δ*, our sample complexity of* $\mathcal{O}(\log(|\mathcal{H}|)+n\log n)$ *appears to violate the lower bound of* $\Omega(\log(|\mathcal{H}|)\log n + n\log\log|\mathcal{H}|)$ *due to Chen, Zhang, and Zhou [6]. 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. In the next subsection, we give lower bounds on the sample complexity of collaborative learning that match our upper bounds.*

4.2 Sample Complexity Lower Bound

We now provide matching lower bounds for agnostic collaborative learning. Our lower bounds hold for collaborative learning algorithms obtaining error of R-OPT$+\varepsilon$, using a randomized or deterministic hypothesis. We call an algorithm an (ε,δ)-collaborative learning algorithm if for any collaborative instances it attains an error of R-OPT $+\,\varepsilon$ with probability at least $1-\delta$.

Theorem 4.3. *Take any* $n, d \in Z_+$, $\varepsilon, \delta \in (0, 1/8)$*, and* (ε,δ)*-collaborative learning algorithm* A*. There exists a collaborative learning problem* $(\mathcal{H},\mathcal{D})$ *with* $|\mathcal{D}| = n$ *and* $|\mathcal{H}| = 2^d$*, on which* A *takes at least* $\Omega\left(\frac{1}{\varepsilon^2}\left(\log|\mathcal{H}| + |\mathcal{D}|\log(\min\{|\mathcal{D}|, \log|\mathcal{H}|\}/\delta)\right)\right)$ *samples.*

Proof sketch. We defer the formal proof of this theorem to Appendix B.3 and sketch the main ideas here. We use $\mathcal{X} = \{1, \dots, d\}$, $\mathcal{Y} = \{+,-\}$, and let $\mathcal{H}$ be the set of all functions $\mathcal{X} \to \mathcal{Y}$. Our construction combines two types of hard distributions. We describe the ideas for the case of $n = d$. First, we use a hard construction for agnostic learning of hypothesis classes with VC dimension d as the distribution of one of the agents. This give us the $\Omega\left(\log(|\mathcal{H}|)/\varepsilon^2\right)$ part of the lower bound. Second, we construct n hard instances each of VC dimension 1 on n independent points. Since the learning algorithms has to solve each problem it has to incur a loss of $n\log(n/\delta)/\varepsilon^2$. □

5 Group DRO and Agnostic Federated Learning

The results we describe in the collaborative learning setting can be generalized to the group DRO setting, and equivalently, agnostic federated learning.

Theorem 5.1. *Consider a group distributionally robust problem* $(\Theta,\mathcal{D})$ *with convex compact unit-diameter parameter space* Θ *of Bregman radius* D_Θ *(Definition A.11), and convex loss* $\ell : \Theta \times \mathcal{Z} \to [0, C]$*. A variant of Algorithm 2 (in particular Algorithm 4 in Appendix 4.1), returns* $\overline{\theta} \in \Theta$ *such that* $\max_{D\in\mathcal{D}} \mathbb{E}_{z\sim D}[\ell(\theta, z)] \le$ *R-OPT* $+\,\varepsilon$*, using a number of samples that is* $\mathcal{O}\left(\frac{D_\Theta C^2 + nC^2\log(n/\delta)}{\varepsilon^2}\right)$.

The proof of this lemma is deferred to Appendix 4.1 and is similar to the proof of Theorem 4.1 except that it uses a generalization of Lemma 3.1 for general convex-concave games. This theorem establishes a generalization bound for the problem of group distributionally robust optimization [27] and improves, by a factor of n, existing sample complexity bounds for agnostic federated learning [20]. This improvement is attained by sampling data on-demand, whereas [20] only chooses a fixed distribution over groups/clients to sample from; this highlights the importance of adapting one's sampling strategy on-the-fly when learning robust models.

Another important question is how fast the training error of stochastic gradient descent converges for the group DRO/AFL settings and was considered by Sagawa et al. [27]. We can transfer our generalization guarantees for on-demand settings into batch settings and achieve the following corollary, which improves on the convergence guarantees of Sagawa et al. [27] by a factor of n.

Corollary 5.2. *Under the same assumptions of Theorem 5.1, we give a procedure (see Appendix 4.1) that minimizes GDRO/AFL training error within ε of R-OPT with probability at least $1-\delta$ in fewer samples than $\mathcal{O}\left(\frac{D_\Theta C^2+nC^2\log(n/\delta)}{\varepsilon^2}\right)$.*

6 Empirical Analysis of On-Demand Sampling for Group DRO

In this section, we empirically analyze an on-demand sampling approach to Group DRO [27].

While traditional Group DRO algorithms follow an approach similar to Algorithm 4—learning adversarial weights over potential distributions using Hedge—group DRO uses the adversary weights to importance-weight datapoints for an ERM learner. Our Corollary 5.2 proves that, by instead using adversarial weights to adjust the sampling frequency of datapoints, we can accelerate convergence by a factor at least linear in n. In other words, our on-demand sampling results suggest that Group DRO should *resample* not *reweight* rare datapoints.

Interestingly, the advantage of *resampling* over *reweighting* has recently been studied for static distribution weights [28]. Our results in this section can be interpreted as empirically demonstrating that *resampling* is also preferable to *reweighting* for adversarially selected weights.

6.1 Experiment Results

We exactly replicate the Group DRO experiments of Sagawa et al. [27] and compare the performance of their reweighting-based implementation of Group DRO with our resampling-based implementation. In particular, we fine-tune Resnet-50 models (convolutional neural networks) [15] and BERT models (transformer-based network) [9] on the image classification datasets Waterbirds [27, 31] and CelebA [17] and the natural language dataset MultiNLI [32] respectively. For each dataset, we compare three methods of training their respective neural network: traditional empirical risk minimization, traditional Group DRO (reweighting), and resampling-based Group DRO.

Sagawa et al. [27] found that, while ERM has high average accuracy on these tasks, ERM has extremely low worst-case accuracy on under-represented distributions. They further found that the use of Group DRO allows for significantly higher worst-case accuracy, but that the method is extremely sensitive to regularization parameters. We will show that resampling can further improve worst-case accuracy, and that resampling-based GDRO does not suffer from the same regularization sensitivity as reweighting-based GDRO.

In our Table 2 and Figure 1, we replicate the Table 1 and Figure 2 of Sagawa et al. [27] respectively, appending our additional results on resampling-based GDRO. These experiments were run identically to their counterparts in Sagawa et al. [27]. They evaluate ERM, reweighting-based Group DRO (GDRO), and resampling-based Group DRO (GDRO-R) in three settings: standard training, under significant weight decay (ℓ-2) regularization, and under early stopping.

Resampling consistently outperforms reweighting. In every dataset and in almost every setting, GDRO-R significantly outperforms GDRO and ERM in worst-case accuracy. Recall that worst-case accuracy, not average accuracy, is the primary performance metric in distributionally robust optimization. We also observe that while GDRO and ERM can have large gaps in worst-case and average accuracy, our GDRO-R has extremely close worst-case and average accuracies. This indicates that resampling is more effective

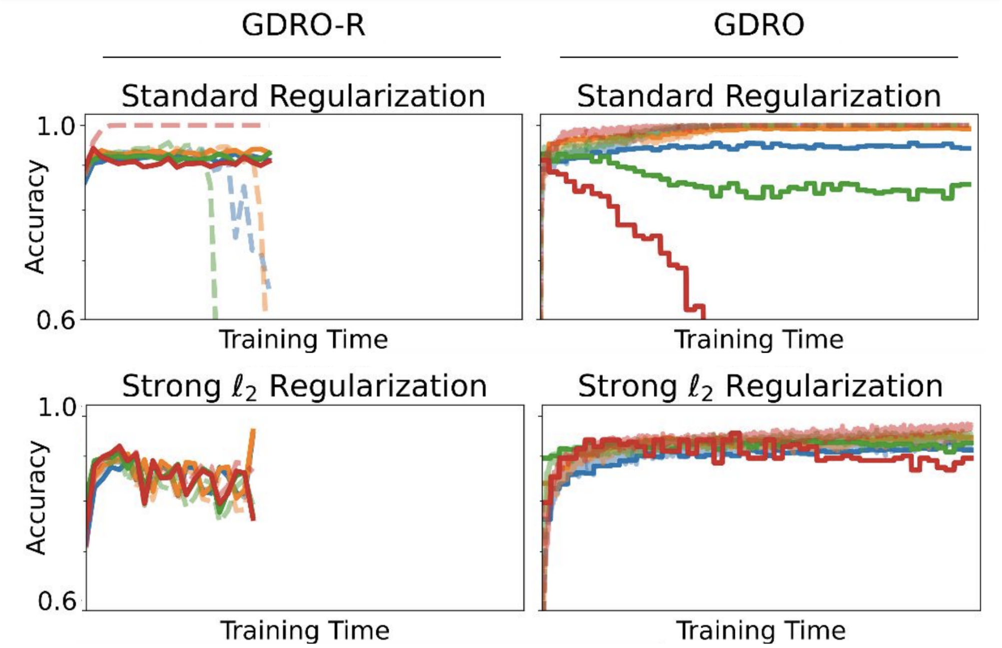

Figure 1: Training (light, dashed) and validation (dark, solid) accuracies for traditional reweighting-based Group DRO (GDRO) and our proposed resampling-based Group DRO (GDRO-R) during training, plotted on a log scale. Note that GDRO-R validation accuracy will be noisier than those of GDRO as we constrain GDRO-R 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 GDRO-R maintains a high red-group accuracy by heavily sampling from the red group, as reflected in the near-perfect red-group training error.

		Average Accuracy			Worst-Case Accuracy		
		ERM	GDRO	GDRO-R	ERM	GDRO	GDRO-R
Standard Reg.	Waterbirds	**97.3 (0.2)**	**97.4 (0.2)**	94.5 (0.3)	60.0 (1.9)	76.9 (1.7)	**86.4 (1.4)**
	CelebA	**94.8 (0.2)**	**94.7 (0.2)**	92.3 (0.2)	41.1 (3.7)	41.7 (3.7)	**88.9 (2.3)**
	MultiNLI	**82.5 (0.1)**	82.2 (0.1)	74.8 (0.1)	66.3 (1.6)	66.6 (1.6)	**70.3 (1.5)**
Strong Reg.	Waterbirds	95.7 (0.3)	**96.6 (0.2)**	89.8 (0.4)	21.3 (1.6)	84.6 (1.4)	**89.4 (1.2)**
	CelebA	**95.8 (0.1)**	93.5 (0.2)	90.0 (0.2)	37.8 (3.6)	86.7 (2.5)	**88.8 (2.3)**
Early Stop	Waterbirds	**93.8 (0.3)**	93.2 (0.3)	92.7 (0.3)	6.7 (1.0)	85.8 (1.4)	**87.1 (1.3)**
	CelebA	**94.6 (0.2)**	91.8 (0.2)	91.3 (0.2)	25.0 (3.2)	88.3 (2.4)	**90.6 (2.2)**
	MultiNLI	**82.8 (0.1)**	81.4 (0.1)	61.4 (0.1)	66.0 (1.6)	**77.7 (1.4)**	43.1 (1.7)

Table 2: Average accuracy and worst-case accuracy (accuracy on lowest-accuracy group) percentages for empirical risk minimization (ERM), traditional reweighting-based Group DRO (GDRO), and our proposed resampling-based Group DRO (GDRO-R). Each method and dataset is evaluated in three settings: with standard ERM hyperparameters (Standard Reg.), with inflated weight decay regularization (Strong Reg.), and with early stopping (Early Stop). These accuracies reflect the testing split of their respective datasets. Standard deviation of measurement is in parentheses. Our proposed GDRO-R consistently outperforms GDRO in worst-case accuracy and performs reliably both with and without inflated regularization.

than reweighting at prioritizing learning on difficult groups. Although our theory (Theorem B.2) predicts that GDRO-R should significantly outperform GDRO in an online setting, it is surprising that simulating on-demand sampling is so effective in offline settings.

Resampling-based GDRO is effective even without strong regularization. The primary observation made by Sagawa et al. [27] in their investigation of Group DRO is that strong regularization is critical for the performance of Group DRO methods. Our experimental results challenge this view. In Table 2, we see GDRO-R retains a high worst-case accuracy both with and without strong regularization. This suggests that the sensitivity of traditional GDRO to hyperparameters is a consequence of reweighting—when a resampling implementation is used, these issues no longer arise.

Resampling-based GDRO converges faster than ERM or reweighting-based GDRO. The GDRO-R methods in Table 2 used a fraction of the training epochs that their GDRO counterparts used. The ratio of GDRO-R 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 5.2.

References

[1] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 1999. ISBN 978-0-521-57353-5. doi: 10.1017/CBO9780511624216. URL `https://www.cambridge.org/core/books/neural-network-learning/665C8C7EB5E2ABC5367A55ADB04E2866`.

[2] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, May 2003. ISSN 0167-6377. doi: 10.1016/S0167-6377(02)00231-6. URL `https://www.sciencedirect.com/science/article/pii/S0167637702002316`.

[3] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Learning theory and kernel machines*, pages 567–580. Springer, 2003.

[4] A. Blum, N. Haghtalab, A. D. Procaccia, and M. Qiao. Collaborative PAC Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://papers.nips.cc/paper/2017/hash/186a157b2992e7daed3677ce8e9fe40f-Abstract.html`.

[5] 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. *arXiv:2103.03228 [cs]*, Mar. 2021. URL `http://arxiv.org/abs/2103.03228`. arXiv: 2103.03228.

[6] J. Chen, Q. Zhang, and Y. Zhou. Tight Bounds for Collaborative PAC Learning via Multiplicative Weights. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/ed519dacc89b2bead3f453b0b05a4a8b-Abstract.html`.

[7] C. Daskalakis, A. Deckelbaum, and A. Kim. Near-Optimal No-Regret Algorithms for Zero-Sum Games. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 235–254. Society for Industrial and Applied Mathematics, Jan. 2011. ISBN 978-0-89871-993-2. doi: 10.1137/1.9781611973082.21. URL `https://epubs.siam.org/doi/abs/10.1137/1.9781611973082.21`.

[8] C. Daskalakis, M. Fishelson, and N. Golowich. Near-Optimal No-Regret Learning in General Games. *arXiv:2108.06924 [cs]*, Aug. 2021. URL `http://arxiv.org/abs/2108.06924`. arXiv: 2108.06924.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical Report arXiv:1810.04805, arXiv, May 2019. URL `http://arxiv.org/abs/1810.04805`. arXiv:1810.04805 [cs] type: article.

[10] J. Duchi and H. Namkoong. Learning Models with Uniform Performance via Distributionally Robust Optimization. *arXiv:1810.08750 [cs, stat]*, July 2020. URL `http://arxiv.org/abs/1810.08750`. arXiv: 1810.08750.

[11] 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, Sept. 1989. ISSN 0890-5401. doi: 10.1016/0890-5401(89)90002-3. URL `https://www.sciencedirect.com/science/article/pii/0890540189900023`.

[12] 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, Aug. 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504. URL `https://www.sciencedirect.com/science/article/pii/S002200009791504X`.

[13] S. Hart and A. Mas-Colell. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica*, 68(5):1127–1150, 2000. ISSN 1468-0262. doi: 10.1111/1468-0262.00153. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00153`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0262.00153.

[14] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. Technical Report arXiv:1512.03385, arXiv, Dec. 2015. URL `http://arxiv.org/abs/1512.03385`. arXiv:1512.03385 [cs] type: article.

[16] 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. Technical Report arXiv:1904.11621, arXiv, Apr. 2019. URL `http://arxiv.org/abs/1904.11621`. arXiv:1904.11621 [cs] type: article.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[18] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21, 2008.

[19] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, MM '10, pages 1485–1488, New York, NY, USA, Oct. 2010. Association for Computing Machinery. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1874254. URL `https://doi.org/10.1145/1873951.1874254`.

[20] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic Federated Learning. Technical Report arXiv:1902.00146, arXiv, Jan. 2019. URL `http://arxiv.org/abs/1902.00146`. arXiv:1902.00146 [cs, stat] type: article.

[21] A. S. Nemirovskij and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience, 1983. Publisher: Wiley-Interscience.

[22] H. L. Nguyen and L. Zakynthinou. Improved Algorithms for Collaborative PAC Learning. *arXiv:1805.08356 [cs, stat]*, Oct. 2018. URL `http://arxiv.org/abs/1805.08356`. arXiv: 1805.08356.

[23] S. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 26, 2013.

[24] V. V. Ramaswamy, S. S. Y. Kim, and O. Russakovsky. Fair Attribute Classification through Latent Space De-biasing. Technical Report arXiv:2012.01469, arXiv, Apr. 2021. URL `http://arxiv.org/abs/2012.01469`. arXiv:2012.01469 [cs] type: article.

[25] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. Publisher: JSTOR.

[26] J. Robinson. An Iterative Method of Solving a Game. *Annals of Mathematics*, 54(2):296–301, 1951. ISSN 0003-486X. doi: 10.2307/1969530. URL `https://www.jstor.org/stable/1969530`. Publisher: Annals of Mathematics.

[27] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[28] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. *arXiv:1911.08731 [cs, stat]*, Apr. 2020. URL `http://arxiv.org/abs/1911.08731`. arXiv: 1911.08731.

[29] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, Nov. 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL `https://doi.org/10.1145/1968.1972`.

[30] N. K. Vishnoi. *Algorithms for Convex Optimization.* Cambridge University Press, 2021. doi: 10.1017/9781108699211.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. *Computation & Neural Systems Technical Report*, 2011. Publisher: California Institute of Technology.

[32] A. Williams, N. Nangia, and S. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`. event-place: New Orleans, Louisiana.

[33] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. Technical Report arXiv:1910.03771, arXiv, July 2020. URL `http://arxiv.org/abs/1910.03771`. arXiv:1910.03771 [cs] type: article.

[34] S. Zakharov, W. Kehl, and S. Ilic. DeceptionNet: Network-Driven Domain Randomization. Technical Report arXiv:1904.02750, arXiv, Aug. 2019. URL `http://arxiv.org/abs/1904.02750`. arXiv:1904.02750 [cs] type: article.

[35] C. Zhang. Information-theoretic lower bounds of PAC sample complexity, Sept. 2019. URL `https://zcc1307.github.io/courses/csc665fa19/notes/lower_bound.pdf`.

Contents

A Full Formulation

In this section, we formally describe our formulations of stochastic convex-concave games and multi-distribution learning problems.

A.1 Convex-Concave Zero-Sum Game

In this subsection, we give a formal definition of a convex-concave zero-sum game and its min-max equilibria. We also introduce assumptions on these games for efficiently finding saddle-points.

Definition A.1. *A* convex-concave two-player zero-sum game *is described by the tuple* (A_-, A_+, ϕ)*, where* $A_- \subset \mathcal{E}_-$ *is a subset of Euclidian space* $\mathcal{E}_-$*,* $A_+ \subset \mathcal{E}_+$ *is a subset of Euclidian space* $\mathcal{E}_+$*, and* $\phi : A_- \times A_+ \to \mathbb{R}$ *is a Lipschitz continuous convex-concave function.*

On a *convex-concave two-player zero-sum game* (A_-, A_+, ϕ), we can define both exact and approximate notions of min-max equilibria in terms of player regrets.

Definition A.2. *The minimizing and maximizing player's* regrets *at a strategy profile* $(p, q) \in A_- \times A_+$ *are denoted Reg-Min, Reg-Max respectively, and defined as,*

$$\textit{Reg-Min}(p, q) := \phi(p, q) - \min_{p^* \in A_-} \phi(p^*, q), \quad \textit{Reg-Max}(p, q) := \max_{q^* \in A_+} \phi(p, q^*) - \phi(p, q).$$

Definition A.3. *A strategy profile* $(p, q) \in A_- \times A_+$ *is a* min-max equilibrium *if both players have zero regret: Reg-Min*$(p, q) = 0$ *and Reg-Max*$(p, q) = 0$*. More weakly,* $(p, q) \in A_- \times A_+$ *is an* ε*-min-max equilibrium if both players have at most* ε *regret: Reg-Min*$(p, q) \leq \varepsilon$ *and Reg-Max*$(p, q) \leq \varepsilon$*.*

In this paper, we may also impose the following assumptions on a convex-concave zero-sum game.

Assumption 1. *The action sets* A_-, A_+ *are compact, convex, and have diameters* R_-, R_+ *respectively:*

$$\forall p, p' \in A_- : \|p - p'\| \leq R_-, \quad \forall q, q' \in A_+ : \|q - q'\| \leq R_+.$$

Assumption 2. *At any* $p, q \in A_- \times A_+$*, the partial subdifferential of the payoff function* ϕ *is non-empty. Furthermore, every partial subgradient vector has a bounded norm:*

$$\|\partial_p \phi(p, q)\|_{\mathcal{E}_-^*} \leq C_-, \quad \|\partial_q \phi(p, q)\|_{\mathcal{E}_+^*} \leq C_+.$$

A.2 Stochastic Settings

In this subsection, we give a formal definition of an asymmetric stochastic setting for a zero-sum game. Our formulation of stochastic first-order oracles observes the convention of representing all randomness in stochastic oracles—and by extension, in any stochastic optimization process—in terms of an i.i.d. sequence of random variables. One nuance our formulation addresses is how randomness can be re-used by stochastic first-order oracles. We do this by formalizing our stochastic setting in terms of multiple i.i.d. sequences of random variables, where the sequence to which a random variable belongs specifies how randomness corresponding to the random variable can be used.

We begin by introducing the notion of a coupled random variable. In the context of a two-player game, a random variable may be coupled to a minimizing player's strategy profile, a maximizing player's strategy profile, neither or both. Our definition formalizes the notion that a random variable can only be interpreted in the context of the mixed strategy to which it is coupled.

Definition A.4. *For any* $p \in A_-$*, we define a random variable* η *to be* p-coupled *if its range is a measurable space* E_p *defined by* p*. Similarly, for any* $q \in A_+$*, we define a random variable* η *to be* q-coupled *if it's range is a measurable space defined by* q*. A random variable* η *is* (p, q)-coupled *if it's range is a measurable space defined by* (p, q)*.*

For convenience, we will denote p-coupled random variables with superscript η^p and, similarly, q-coupled random variables with superscript η^q. Random variables that are not coupled will be denoted by $\eta^\perp$ when such clarification is necessary.

We will now define stochastic first-order oracles that express their randomness in terms of sequences of i.i.d. coupled random variables.

Definition A.5. *In a zero-sum two-player game, the minimizing player's* randomness source *is defined as a set* $\boldsymbol{\xi}_- \subseteq \{\boldsymbol{\xi}_-^q \mid q \in A_+ \bigcup \{\perp\}\}$*, where* $\boldsymbol{\xi}_-^q := \left\{\xi_{-;i}^q\right\}_{i \in \mathbb{Z}}$ *is a sequence of i.i.d. random variables all coupled with* $q \in A_+$*. In addition, all random variables in all sequences in* $\boldsymbol{\xi}_-$ *are independent.*

Definition A.6. *In a zero-sum two-player game, the maximizing player's* randomness source *is defined as a set* $\boldsymbol{\xi}_+ \subseteq \left\{\boldsymbol{\xi}_+^p \mid p \in A_- \bigcup \{\perp\}\right\}$*, where* $\boldsymbol{\xi}_+^p := \left\{\xi_{+;i}^p\right\}_{i \in \mathbb{Z}}$ *is a sequence of i.i.d. random variables all coupled with* $p \in A_+$*. In addition, all random variables in all sequences in* $\boldsymbol{\xi}_+$ *are independent.*

Definition A.7. *For any* $q \in A_+$*, consider the function* $\widehat{g}_-^q : E^q \times A_- \times A_+ \to \mathcal{E}_-^*$*. The minimizing player has a* locally unbiased first-order oracle *if there exists, for all* $q \in A_+$*, a* $\widehat{g}_-^q$ *such that for all* $p \in A_-$ *and* $i \in \mathbb{Z}$*:*

$$\mathop{\mathbb{E}}_{\xi_{-;i}^q}\left[\widehat{g}_-^q(\xi_{-;i}^q, p, q)\right] = \partial_p \phi(p, q).$$

We analogously define locally unbiased oracles for the maximizing player.

When q is clear from context, we write $\widehat{g}_-^q$ as $\widehat{g}_-$. We can also define a globally unbiased oracle.

Definition A.8. *For any* $q \in A_+$*, consider the function* $\widehat{g}_-^\perp : A_- \times A_+ \to \mathcal{E}_-^*$*. The minimizing player has a* globally unbiased first-order oracle *if there exists* $\widehat{g}_-^\perp$ *where for all* $q \in A_+$ *and* $p \in A_-$ *and* $i \in \mathbb{Z}$*:*

$$\mathop{\mathbb{E}}_{\xi_{-;i}^\perp}\left[\widehat{g}_-^\perp(\xi_{-;i}^\perp, p, q)\right] = \partial_p \phi(p, q).$$

We analogously define globally unbiased first-order oracles for the maximizing player.

Finally, we may impose the following norm-bound assumption on the first-order oracles we discuss.

Assumption 3. *Every globally unbiased first-order oracle has a range with bounded norm:* $\left\|\widehat{g}_-^\perp(\cdot)\right\|_{\mathcal{E}_-^*} \leq C_-$, $\left\|\widehat{g}_+^\perp(\cdot)\right\|_{\mathcal{E}_+^*} \leq C_+$*. Furthermore, every locally unbiased first-order oracle also has a range with bounded norm: for all* $p, q \in A_-, A_+$, $\|\widehat{g}_-^q(\cdot)\|_{\mathcal{E}_-^*} \leq C_-$, $\left\|\widehat{g}_+^p(\cdot)\right\|_{\mathcal{E}_+^*} \leq C_+$,

A.3 Multi-Distribution Learning

In this subsection, we give a formal definition of multi-distribution learning that unifies the problem formulations of collaborative learning [4], agnostic federated learning [20], and group DRO [27]. We further introduce assumptions that characterize two special cases of multi-distribution learning: *convex multi-distribution learning* and *binary classifier multi-distribution learning.*

We begin by reviewing some common definitions from convex optimization.

Definition A.9. *Let* Z *be a convex compact subset of a Euclidian space* $\mathcal{E}$ *with norm* $\|\cdot\|$*. A distance generating function on* Z *is a function* $\omega : Z \to \mathbb{R}$*, where:*

1. *ω is continuous and strongly convex, modulus 1, w.r.t to $\|\cdot\|$ on Z.*
2. *There exists a non-empty subset $Z^o \subset Z$ where the subdifferential $\partial\omega$ is non-empty and $\partial\omega$ admits a continuous selection on Z^o.*

Furthermore, the center of Z *w.r.t.* ω *is defined as* $z^c := \arg\min_{z \in Z^o} \omega(z)$*.*

Definition A.10. *The prox function* $V : Z^o \times Z \to \mathbb{R}^+$ *associated with a distance generating function* $\omega : Z \to \mathbb{R}$ *is defined as:*

$$V(z, u) := \omega(u) - \omega(z) - \langle \omega'(z), u - z \rangle .$$

The prox function is also known as the Bregman divergence.

Definition A.11. *Given a convex set Z with a distance generating function ω satisfying Definition A.9, the Bregman radius is defined as $\max_{u \in Z} V(z^c, u) \leq D_Z$, where z^c is the center of Z as defined in Definition A.9.*

We state our most general formulation of multi-distribution learning as follows.

Definition A.12. *Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be a space of datapoints and $\mathcal{D} = \{D_i\}_{i=1}^n$ be a finite set of n joint probability distributions over $\mathcal{Z}$. Let Θ denote a set of parameters and $\ell : \Theta \times \mathcal{Z} \to [0, L]$ be a loss function. Then the tuple $(\Theta, \mathcal{D}, \ell)$ describes a multi-distribution learning problem, w.r.t. $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$.*

One case of multi-distribution learning we study in this paper is *convex multi-distribution learning*, which includes as special cases the problem formulations of Sagawa et al. [27] and Mohri et al. [20]. *Convex multi-distribution learning* also encompasses the problem formulation of Blum et al. [4] for finite hypothesis spaces, i.e., when $|\mathcal{H}| < \infty$.

Definition A.13. *The tuple $(\Theta, \mathcal{D}, \ell)$ describes a convex multi-distribution learning problem when Θ is convex compact, ℓ is convex in Θ, and there exists a distance generating function $\omega : \Theta \to \mathbb{R}$ on our parameter space Θ.*

Definition A.14. *The diameter of the parameter space Θ is an $R_\Theta > 0$ satisfying:*

$$\forall \theta, \theta' \in \Theta : \|\theta - \theta'\| \leq R_\Theta.$$

Assumption 4. *Given a convex multi-distribution learning problem $(\Theta, \mathcal{D}, \ell)$, we assume that, for any datapoint z in the supports of the distributions in $\mathcal{D}$ and any $\theta \in \Theta^o$, the partial subgradient of $\ell(\theta, z)$ w.r.t. θ has bounded norm:*

$$\|\partial_\theta \ell(\theta, z)\|_{\mathcal{E}^*} \leq C.$$

Assumption 5. *Given a convex multi-distribution learning problem $(\Theta, \mathcal{D}, \ell)$, we assume there exists a distance generating function ω where Θ has bounded Bregman radius D_Θ.*

Remark A.1. *As ω is strongly convex modulus 1 by definition, any Θ satisfying Assumption 5 has a finite diameter $R_\Theta \leq 2\sqrt{2D_\Theta}$*

Another important case of multi-distribution learning is *binary classifier multi-distribution learning*, which includes the problem formulations of Blum et al. [4] both for finite hypothesis spaces ($|\mathcal{H}| < \infty$) and finite VC dimension hypothesis spaces ($\text{VCD}(\mathcal{H}) < \infty$).

Definition A.15. *The tuple $(\Theta, \mathcal{D})$ describes a binary classifier multi-distribution learning problem when $\mathcal{Z} = \mathcal{X} \times \{0, 1\}$, Θ is the set of probability distributions over a set of binary classification rules $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ and $\ell(\theta, (x, y)) := \mathbb{E}_{h \sim \theta}[1[h(x) = y]]$.*

Remark A.2. *A binary classifier multi-distribution learning problem $(\Theta, \mathcal{D}, \ell)$ is equivalent to a convex multi-distribution learning problem $(\Theta, \mathcal{D}, \ell)$ when the support of Θ is finite, i.e., Θ is a probability distribution over a finite number of binary classification rules.*

Finally, we can define a multi-distribution analogue to probably-approximately-correct learning [29].

Definition A.16. *An example oracle $EX(D)$ is an infinite set of i.i.d. samples from a probability distribution D over datapoints. Colloquially, a "new call" to example oracle $EX(D)$ refers to realizing a previously unrealized sample in $EX(D)$.*

Definition A.17. *A learning algorithm A is an (ε, δ) multi-distribution learning algorithm for a set of multi-distribution learning problems $\mathbb{V} := \{(\Theta_i, \mathcal{D}_i, \ell_i)\}_i$ if, given any problem $(\Theta_i, \mathcal{D}_i, \ell_i) \in \mathbb{V}$, accessing only the tuple $(\Theta_i, \ell_i, \{EX(D) \mid D \in \Delta\mathcal{D}\})$, A outputs a parameter $\theta \in \Theta_i$ that satisfies, with probability at least $1 - \delta$:*

$$\max_{D \in \mathcal{D}} \text{Risk}_D(\theta) \leq \inf_{\theta^* \in \Theta} \max_{D \in \mathcal{D}} \text{Risk}_D(\theta^*) + \varepsilon.$$

We use (ε, δ)-algorithm as a shorthand for (ε, δ) multi-distribution learning algorithm.

Definition A.18. *A multi-distribution learning algorithm A has a sample complexity of N (or "takes N samples") on a set of multi-distribution learning problems $\mathbb{V} := \{(\Theta_i, \mathcal{D}_i, \ell_i)\}_i$ if N is the smallest integer such that, given any problem $V \in \mathbb{V}$, the event that A takes more than N samples is measure-zero. If no such N exists, we say A has infinite sample complexity.*

B Omitted Proofs

B.1 Proof of Lemma B.1 (Generalization of Lemma 3.1)

Algorithm 3 Finding Equilibria in Convex-Concave Zero-Sum Games with Asymmetric Costs.

```
**Output:** Mixed strategy profile $(p, q) \in A_- \times A_+$.
**Input:** Action sets $A_-, A_+$, cost $r \in \mathbb{Z}_+$, timesteps $T$, initial actions $p^{(1)}, q^{(1)}$, and no-regret learning algorithms $\mathcal{Q}_- : \{A_- \times \mathcal{E}_-^*\} \to A_-$, $\mathcal{Q}_+ : \{A_+ \times \mathcal{E}_+^*\} \to A_+$.
**for** $t = 1, 2, \ldots, T$ **do**
    Realize randomness $\xi^{q^{(t)}}_{-;t}$ and $\xi^{\perp}_{+;\lceil \frac{t}{r} \rceil}$.
    Take estimates $\widehat{g}_-^{(t)} := \widehat{g}_- \left( \xi^{q^{(t)}}_{-;t}, p^{(t)}, q^{(t)} \right)$ and $\widehat{g}_+^{(t)} := \widehat{g}_+ \left( \xi^{\perp}_{+;\lceil \frac{t}{r} \rceil}, p^{(t)}, q^{(t)} \right)$.
    Run the no-regret updates:
        $p^{(t+1)} = \mathcal{Q}_- \left( \left\{ p^{(1)}, \widehat{g}_-^{(1)} \right\}, \ldots \left\{ p^{(t)}, \widehat{g}_-^{(t)} \right\} \right)$
        $q^{(t+1)} = \mathcal{Q}_+ \left( \left\{ q^{(1)}, \widehat{g}_+^{(1)} \right\}, \ldots \left\{ p^{(t)}, \widehat{g}_+^{(t)} \right\} \right)$
**end for**
Return the uniformly mixed strategies $\overline{p} = \frac{1}{T} \sum_{t=1}^{T} p^{(t)}$ and $\overline{q} = \frac{1}{T} \sum_{t=1}^{T} q^{(t)}$.
```

We first recall the following lemma from Section 3.

Lemma 3.1. *Let (A_-, A_+, ϕ) be a finite zero-sum game. Assume there exists $\xi^{q^{(t)}}$ of cost 1 providing locally unbiased estimates $\widehat{g}_-(\cdot)$ and there exists $\xi^{\perp(a)}$ of cost r providing globally unbiased estimates $\widehat{g}_+(\cdot)$. With probability $1 - \delta$, Algorithm 1 returns an ε-min-max equilibrium of the game, so long as*

$$T \geq \frac{18}{\varepsilon^2} \left(\max \left\{ \frac{9 \log |A_-|}{4}, 8 \log \left(\frac{r+1}{\delta} \right) \right\} + \max \left\{ \frac{9 \log |A_+|}{4}, \frac{8r^2}{r+1} \log \left(\frac{r+1}{\delta} \right) \right\} \right). \tag{9}$$

Moreover, the total cost of randomness incurred by the algorithm is at most $2t$.

We will prove a more general result, Lemma B.1, that implies Lemma 3.1 as a special case. Lemma B.1 provides sample complexity upper bounds for Algorithm 3, an algorithm for approximating the saddle-point of a convex-concave game with high-probability. Algorithm 3 is also a generalization of Algorithm 1.

Lemma B.1 (Generalization of Lemma 3.1)**.** *Let (A_-, A_+, ϕ) be a convex-concave game satisfying Definition A.1 and Assumptions 1 and 2. Suppose the minimizing player has a locally unbiased first-order oracle $\widehat{g}_-$ and the maximizing player has a globally unbiased first-order oracle $\widehat{g}_+$, with both oracles satisfying Assumptions 3. Take $\mathcal{Q}_-$ to be any no-regret algorithms with the guarantee that for, any sequence $g^{(1)}, \ldots, g^{(T)} \in \mathcal{E}_-^*$, if $\left\| g^{(i)} \right\|_{\mathcal{E}_-^*} \leq C$ for all $i \in [T]$, the $\mathcal{Q}_-$-learned sequence $w^{(1)}, \ldots, w^{(T)}$ satisfies:*

$$\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) \leq \sqrt{\frac{\gamma_-(T, A_-, C)}{T}}.$$

Take $\mathcal{Q}_+$ to be any no-regret algorithms with the guarantee that for, any sequence $g^{(1)}, \ldots, g^{(T)} \in \mathcal{E}_+^*$, if $\left\| g^{(i)} \right\|_{\mathcal{E}_+^*} \leq C$ for all $i \in [T]$, the $\mathcal{Q}_+$-learned sequence $w^{(1)}, \ldots, w^{(T)}$ satisfies:

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \leq \sqrt{\frac{\gamma_+(T, A_+, C)}{T}}.$$

Then, the mixed strategy profile $(\overline{p}, \overline{q})$ outputted by Algorithm 3 is an ε-min-max equilibrium with probability at least $1 - \delta$ so long as:

$$T \geq \frac{9}{\varepsilon^2} \left(\gamma_-(T, A_-, 2C_-) + 8R_-^2 C_-^2 \log \left(\frac{r+1}{\delta} \right) + \gamma_+(T, A_+, 2C_+) + \frac{8R_+^2 C_+^2 r^2}{r+1} \log \left(\frac{r+1}{\delta} \right) \right). \tag{10}$$

Moreover, exactly T elements of $\boldsymbol{\xi}_{\text{-}}$ and $\lceil T/r \rceil$ elements of $\boldsymbol{\xi}_{+}$ (defined in Definitions A.7 and A.8) will be realized. This means that if sampling from $\boldsymbol{\xi}_{\text{-}}$ incurs a unit cost and sampling from $\boldsymbol{\xi}_{+}$ incurs at most r unit cost, total cost will be at most $2r\lceil T/r \rceil$.

Before proving Lemma B.1, we review the following technical results.

First, we note an immediate consequence of working with a convex payoff function.

Fact B.1. *Let $\phi : Z \to \mathbb{R}$ be a convex function on a convex compact domain Z and $g^{(t)} = \partial\phi(w^{(t)})$ be a partial subgradient of ϕ at $w^{(t)}$. Then, for any $[w^{(t)}]_{t=1}^{T} \in Z$:*

$$\phi\left(\sum_{t=1}^{T} w^{(t)}\right) - \min_{w^* \in Z} \phi(w^*) \leq \mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \coloneqq \max_{w^* \in Z} \sum_{t=1}^{T} \left\langle g^{(t)}, w^{(t)} - w^* \right\rangle.$$

Proof. Fix any $w^* \in Z$. By our choice of g, we know that

$$\sum_{t=1}^{T} \left\langle g^{(t)}, w^{(t)} - w^* \right\rangle = \sum_{t=1}^{T} \left\langle \partial_{w^{(t)}} \phi(w^{(t)}), w^{(t)} - w^* \right\rangle.$$

By the convexity of ϕ, it follows that

$$\sum_{t=1}^{T} \left\langle \partial_{w^{(t)}} \phi(w^{(t)}), w^{(t)} - w^* \right\rangle \geq \sum_{t=1}^{T} \phi(w^{(t)}) - \phi(w^*) \geq \phi\left(\sum_{t=1}^{T} w^{(t)}\right) - \phi(w^*).$$

with equality when ϕ is bilinear. □

Fact B.2. *Let $\phi : A_{\text{-}} \times A_{+} \to \mathbb{R}$ be a convex-concave function on convex compact domains $A_{\text{-}}, A_{+}$ and define the operators $g_{\text{-}}^{(t)} \coloneqq \partial_{p^{(t)}} \phi\left(p^{(t)}, q^{(t)}\right)$ and $g_{+}^{(t)} \coloneqq \partial_{q^{(t)}} \phi\left(p^{(t)}, q^{(t)}\right)$. Given sequences $p^{(1)}, \ldots, p^{(T)} \in A_{\text{-}}$ and $q^{(1)}, \ldots, q^{(T)} \in A_{+}$, their ergodic averages $p^{(1:T)} \coloneqq \frac{1}{T}\sum_{t=1}^{T} p^{(t)}$ and $q^{(1:T)} \coloneqq \frac{1}{T}\sum_{t=1}^{T} q^{(t)}$ constitute an ε-equilibrium if $\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) \leq \varepsilon$ and $\mathrm{Err}_{\mathrm{V}}(q^{(1:T)}) \leq \varepsilon$.*

Proof. By Fact B.1, when variational errors are bounded as $\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) \leq \varepsilon$ and $\mathrm{Err}_{\mathrm{V}}(q^{(1:T)}) \leq \varepsilon$, we know player regrets are bounded: $\text{Reg-Min}(p^{(1:T)}, q^{(1:T)}) \leq \varepsilon$ and $\text{Reg-Max}(p^{(1:T)}, q^{(1:T)}) \leq \varepsilon$. This satisfies our Definition A.3 for an ε-min-max equilibria. □

We now claim concentration results for locally unbiased and globally unbiased first-order oracles.

Fact B.3. *Let $(A_{\text{-}}, A_{+}, \phi)$ be a convex-concave game satisfying Definition A.1 and Assumptions 1 and 2. Without loss of generality, let our player of interest be the minimizing player. Consider a play sequence $\left\{p^{(t)}, q^{(t)}\right\}_{t=1}^{T}$ with some complementary sequence $\left\{y^{(t)}\right\}_{t=1}^{T} \in A_{\text{-}}$. Suppose, at each timestep, the minimizing player uses a random variable $\widehat{g}_{\text{-}}^{(t)}$ to estimate $g_{\text{-}}^{(t)} \coloneqq \partial_{p^{(t)}} \phi\left(p^{(t)}, q^{(t)}\right)$. If the following assumptions hold:*

1. *For every $t \in [T]$, the subsequences $\left\{p^{(\tau)}, q^{(\tau)}, y^{(\tau)}\right\}_{\tau=1}^{t}$ is independent of $\widehat{g}_{\text{-}}^{(t)}, \ldots \widehat{g}_{\text{-}}^{(T)}$.*
2. *All estimates $\widehat{g}_{\text{-}}^{(1)}, \ldots, \widehat{g}_{\text{-}}^{(T)}$ are independent.*
3. *$\widehat{g}_{\text{-}}^{(t)}$ is an unbiased estimate of $g_{\text{-}}^{(t)}$ and additionally satisfies Assumption 3.*

We can then bound the error of the stochastic oracle, $\varepsilon_{\text{-}}^{(t)} \coloneqq g_{\text{-}}^{(t)} - \widehat{g}_{\text{-}}^{(t)}$, with respect to our play sequence as follows. With probability at least $1 - \delta$,

$$\max_{p^* \in A_{\text{-}}} \frac{1}{T} \sum_{t=1}^{T} \left\langle \varepsilon_{\text{-}}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \leq \sqrt{\frac{8R_{\text{-}}^2 C_{\text{-}}^2 \log\left(\frac{1}{\delta}\right)}{T}}. \tag{11}$$

Proof. Define the filtration $\left\{\mathcal{F}^{(t)}\right\}_{t=0}^{T}$ as the sigma algebra generated by $\left\{\widehat{g}_{-}^{(t)}\right\}_{t=1}^{T}$, with $\mathcal{F}^{(0)}$ being a singleton containing only the superset of our sigma algebra. Observe that $\varepsilon_{-}^{(t)}$ is independent of $\left\{p^{(\tau)}\right\}_{\tau=1}^{t}$ and $\left\{y^{(\tau)}\right\}_{\tau=1}^{t}$ by assumption. As $\widehat{g}_{-}(\cdot)$ is unbiased, for any $t' = 0, \dots, t-1$:

$$\mathbb{E}\left[\left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \mid \mathcal{F}^{(t')}\right] = 0.$$

We can thus construct the Doob martingale:

$$U = \left\{\mathbb{E}\left[\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \mid \mathcal{F}^{(t')}\right], \mathcal{F}^{(t')}\right\}_{t'=0}^{T},$$

and bound the difference sequence accordingly. For any $t' \in [T]$, we have the deterministic bound:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \mid \mathcal{F}^{(t')}\right] - \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \mid \mathcal{F}^{(t'-1)}\right] \\
&= \left|\left\langle \varepsilon_{-}^{(t')}, p^{(t')} - y^{(t')} \right\rangle\right| \\
&\leq \left\|\varepsilon_{-}^{(t')}\right\|_{*} \left\|p^{(t')} - y^{(t')}\right\|,
\end{aligned}$$

where the final inequality is Holder's. Since, by Assumption 1, the diameter of our action sets are bounded by R_{-}, we know $\left\|p^{(t')} - y^{(t')}\right\| \leq R_{-}$. Invoking Assumptions 2 and 3, we know $\left\|\varepsilon_{-}^{(t')}\right\|_{*} \leq 2C_{-}$. By the Azuma-Hoefdding inequality, we can thus bound, for any $\varepsilon > 0$,

$$\Pr\left(\frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle \geq \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2}{8TC_{-}^2 R_{-}^2}\right).$$

□

Fact B.4. *Let (A_{-}, A_{+}, ϕ) be a convex-concave game satisfying Definition A.1 and Assumptions 1 and 2. Without loss of generality, let our player of interest be the minimizing player. Consider a play sequence $\left\{p^{(t)}, q^{(t)}\right\}_{t=1}^{T}$. If the following assumptions hold:*

1. *$\widehat{g}_{-}^{(t)}$ is an estimate of $g_{-}^{(t)} := \partial_{p^{(t)}} \phi\left(p^{(t)}, q^{(t)}\right)$ that satisfies Assumption 3.*
2. *There exists the no-regret algorithm $\mathcal{Q}_{-}$ satisfying the assumptions of Lemma B.1.*

We can then bound the error of the stochastic oracle, $\varepsilon_{-}^{(t)} := g_{-}^{(t)} - \widehat{g}_{-}^{(t)}$, with respect to our play sequence as follows. Define $y^{(t+1)} := \mathcal{Q}_{-}\left(\left\{y^{(\tau)}, \varepsilon_{-}^{(\tau)}\right\}_{\tau=1}^{t}\right)$. With probability at least $1 - \delta$,

$$\max_{p^* \in A_{-}} \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - p^* \right\rangle \leq \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle + \sqrt{\frac{\gamma_{-}\left(T, A_{-}, 2C_{-}\right)}{T}}.$$

Proof. We can rewrite Equation 11 with respect to a sequence

$$\begin{aligned}
\max_{p^* \in A_{-}} \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - p^* \right\rangle &= \max_{p^* \in A_{-}} \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} + y^{(t)} - p^* \right\rangle \\
&= \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, p^{(t)} - y^{(t)} \right\rangle + \max_{p^* \in A_{-}} \frac{1}{T}\sum_{t=1}^{T} \left\langle \varepsilon_{-}^{(t)}, y^{(t)} - p^* \right\rangle
\end{aligned}$$

We will first bound the summand:

$$\max_{p^*\in A_{\text{-}}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \varepsilon_{\text{-}}^{(t)}, y^{(t)} - p^* \right\rangle.$$

By definition, our sequence $y^{(1)}, \dots y^{(T)}$ is a $\mathcal{Q}_{\text{-}}$-learned sequence for the operator errors $\varepsilon_{\text{-}}^{(1)}, \dots, \varepsilon_{\text{-}}^{(T)}$. By Assumptions 2 and 3, we enforce that all operators and operator estimates have bounded norm, i.e., $\|g_{\text{-}}(\cdot)\|_{\mathcal{E}^*} \le C_{\text{-}}$ and $\|\widehat{g}_{\text{-}}(\cdot)\|_{\mathcal{E}^*} \le C_{\text{-}}$. By triangle inequality, we can bound $\left\|\varepsilon_{\text{-}}^{(t')}\right\|_* \le 2C_{\text{-}}$. Hence, the guarantees of $\mathcal{Q}_{\text{-}}$ imply:

$$\max_{p^*\in A_{\text{-}}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \varepsilon_{\text{-}}^{(t)}, y^{(t)} - p^* \right\rangle \le \sqrt{\frac{\gamma_{\text{-}}\left(T, A_{\text{-}}, 2C_{\text{-}}\right)}{T}}.$$

□

We now prove our general claim.

Proof of Lemma B.1. By Fact B.2, it suffices to prove the variational error bounds for each player:

$$\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) \le \varepsilon, \quad \mathrm{Err}_{\mathrm{V}}(q^{(1:T)}) \le \varepsilon,$$

with respect to the operators,

$$g_{\text{-}}^{(t)} \coloneqq \partial_{p^{(t)}}\phi\left(p^{(t)}, q^{(t)}\right), \quad g_{+}^{(t)} \coloneqq \partial_{q^{(t)}}\phi\left(p^{(t)}, q^{(t)}\right).$$

In Algorithm 3, we estimate the true operators $\left\{g_{\text{-}}^{(t)}\right\}_{t=1}^{T}, \left\{g_{+}^{(t)}\right\}_{t=1}^{T}$ with the stochastic estimates:

$$\widehat{g}_{\text{-}}^{(t)} \coloneqq \widehat{g}_{\text{-}}\left(\xi_{\text{-};t}^{q^{(t)}}, p^{(t)}, q^{(t)}\right), \quad \widehat{g}_{+}^{(t)} \coloneqq \widehat{g}_{+}\left(\xi_{+;\lceil \frac{t}{r}\rceil}^{\perp}, p^{(t)}, q^{(t)}\right).$$

Let $\varepsilon_{\text{-}}^{(t)} \coloneqq g_{\text{-}}^{(t)} - \widehat{g}_{\text{-}}^{(t)}$ and let $\varepsilon_{+}^{(t)} \coloneqq g_{+}^{(t)} - \widehat{g}_{+}^{(t)}$ denote the difference between our true and estimated operators at each timestep. We can thus divide each variational error into a *training error* and *generalization error* component:

$$\begin{aligned}\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) &\le \max_{p^*\in A_{\text{-}}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \widehat{g}_{\text{-}}^{(t)}, p^{(t)} - p^* \right\rangle + \max_{p^*\in A_{\text{-}}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \varepsilon_{\text{-}}^{(t)}, p^{(t)} - p^* \right\rangle \\ \mathrm{Err}_{\mathrm{V}}(q^{(1:T)}) &\le \max_{q^*\in A_{+}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \widehat{g}_{+}^{(t)}, q^{(t)} - q^* \right\rangle + \max_{q^*\in A_{+}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \varepsilon_{+}^{(t)}, q^{(t)} - q^* \right\rangle\end{aligned}$$

We handle the training error first. Recall that $p^{(1)}, \dots p^{(T)}$ is a $\mathcal{Q}_{\text{-}}$-learned sequence for the operator sequence $\widehat{g}_{\text{-}}^{(1)}, \dots, \widehat{g}_{\text{-}}^{(T)}$. By Assumption 3, we enforce that all operator estimates have bounded norm, i.e., $\|\widehat{g}_{\text{-}}(\cdot)\|_{\mathcal{E}^*} \le C_{\text{-}}$. Hence, the guarantees of $\mathcal{Q}_{\text{-}}$ imply:

$$\max_{p^*\in A_{\text{-}}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \widehat{g}_{\text{-}}^{(t)}, p^{(t)} - p^* \right\rangle \le \sqrt{\frac{\gamma_{\text{-}}\left(T, A_{\text{-}}, C_{\text{-}}\right)}{T}}. \tag{12}$$

Similarly, $q^{(1)}, \dots q^{(T)}$ is a $\mathcal{Q}_{+}$-learned algorithms enjoying $\mathcal{Q}_{+}$'s guarantee:

$$\max_{q^*\in A_{+}} \frac{1}{T}\sum_{t=1}^{T}\left\langle \widehat{g}_{+}^{(t)}, q^{(t)} - q^* \right\rangle \le \sqrt{\frac{\gamma_{+}\left(T, A_{+}, C_{+}\right)}{T}}. \tag{13}$$

We now handle the generalization error. We first consider the minimization player. Observe that, for every $t \in [T]$, the play sequence $\left\{p^{(\tau)}, q^{(\tau)}\right\}_{\tau=1}^{t}$ is measurable by $\left\{\xi_{-;\tau}^{q^{(\tau)}}, \xi_{+;\lceil \tau/r \rceil}^{\perp}\right\}_{\tau=1}^{t-1}$, which $\xi_{-;t}^{q^{(t)}}$ is independent of by construction. We can thus invoke Facts B.3 and B.4 to bound, with probability at least $1-\delta$:

$$\max_{p^* \in A_-} \frac{1}{T} \sum_{t=1}^{T} \left\langle \varepsilon_-^{(t)}, p^{(t)} - p^* \right\rangle \leq \sqrt{\frac{8R_-^2 C_-^2 \log\left(\frac{1}{\delta}\right)}{T}} + \sqrt{\frac{\gamma_-\left(T, A_-, 2C_-\right)}{T}}. \tag{14}$$

We now consider the maximization player. First, we invoke Fact B.4 to separate:

$$\max_{q^* \in A_+} \frac{1}{T} \sum_{t=1}^{T} \left\langle \varepsilon_+^{(t)}, q^{(t)} - q^* \right\rangle \leq \frac{1}{T} \sum_{t=1}^{T} \left\langle \varepsilon_+^{(t)}, q^{(t)} - y^{(t)} \right\rangle + \sqrt{\frac{\gamma_+\left(T, A_+, 2C_+\right)}{T}}. \tag{15}$$

For notional convenience, let $i(j) = (j-1)r + i$ denote the ith timestep of the jth period. Also let $m_i \coloneqq \left|\{i(j)\}_{j=1}^{\infty} \bigcup [T]\right|$ denote the number of valid timesteps that can be written as $i(j)$. Observe that $m_i \leq \lceil T/r \rceil$. Fix a choice of $i \in [r]$. Observe that, for every $k \in [m_i]$, the play sequence $\left\{p^{(i(j))}, q^{(i(j))}\right\}_{j=1}^{k}$ is measurable by $\left\{\xi_{-;i(j)}^{q^{(i(j))}}, \xi_{+;j-1}^{\perp}\right\}_{j=1}^{k-1}$, which $\xi_{+;j}^{\perp}$ is independent of by construction. We can thus again invoke Fact B.3 to bound, with probability at least $1-\delta$:

$$\max_{q^* \in A_+} \frac{1}{m_i} \sum_{j=1}^{m_i} \left\langle \varepsilon_+^{(i(j))}, q^{(i(j))} - y^{(i(j))} \right\rangle \leq \sqrt{\frac{8R_+^2 C_+^2 \log\left(\frac{r}{\delta}\right)}{m_i}}.$$

Taking a union bound over said Azuma inequality for all $i \in [r]$, we have that with probability at least $1-\delta$,

$$\begin{aligned}
\max_{q^* \in A_+} \sum_{t=1}^{T} \left\langle \varepsilon_+^{(t)}, q^{(t)} - y^{(t)} \right\rangle &= \sum_{i=1}^{r} \sum_{j=1}^{m_i} \left\langle \varepsilon_+^{(i(j))}, q^{(i(j))} - y^{(i(j))} \right\rangle \\
&\leq \sum_{i=1}^{r} \sqrt{8 m_i R_+^2 C_+^2 \log(r/\delta)} \\
&\leq \sqrt{8 \frac{r^2}{r-1} T R_+^2 C_+^2 \log(r/\delta)} \qquad (16) \\
\text{(optional: assuming } r \geq 1) \qquad &\leq \sqrt{8(r+2) T R_+^2 C_+^2 \log(r/\delta)}.
\end{aligned}$$

Gluing together our bounds on training error (Equation 12, Equation 13) and generalization error (Equation 14, Equation 15, Equation 16) with triangle inequalities and union bounds, we have with probability at least $1-\delta$,

$$\begin{aligned}
\mathrm{Err}_{\mathrm{V}}(p^{(1:T)}) \leq& \sqrt{\frac{\gamma_-(T, A_-, C_-)}{T}} + \sqrt{\frac{\gamma_-(T, A_-, 2C_-)}{T}} + \sqrt{\frac{8R_-^2 C_-^2 \log\left(\frac{r+1}{\delta}\right)}{T}} \\
\mathrm{Err}_{\mathrm{V}}(q^{(1:T)}) \leq& \sqrt{\frac{\gamma_+(T, A_+, C_+)}{T}} + \sqrt{\frac{\gamma_+(T, A_+, 2C_+)}{T}} + \sqrt{\frac{8r^2 R_+^2 C_+^2 \log\left(\frac{r+1}{\delta}\right)}{(r-1)T}}.
\end{aligned}$$

□

B.2 Proof of Theorem B.2 (Generalization of Theorems 4.1 and 5.1)

Fact B.5. *Let $(\Theta, \mathcal{D}, \ell)$ be a multi-distribution learning problem satisfying Definition A.12. Define a corresponding convex-concave game $(A_{\text{-}}, A_{+}, \phi)$ where:*

$$A_{\text{-}} = \Theta, \quad A_{+} = \Delta\mathcal{D}, \quad \phi(p, q) = \mathrm{Risk}_q(p).$$

The minimizing player's mixed strategy $\overline{p}$ in any ε-min-max equilibria (Definition A.3) constitutes an 2ε-error solution to $(\Theta, \mathcal{D}, \ell)$.

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

$$\mathrm{Risk}_{\overline{q}}(\overline{p}) \leq \min_{p \in \Theta} \mathrm{Risk}_{\overline{q}}(p) + \varepsilon \text{ and } \mathrm{Risk}_{\overline{q}}(\overline{p}) \geq \max_{q \in \Delta\mathcal{D}} \mathrm{Risk}_q(\overline{p}) - \varepsilon.$$

Equivalently, by the min-max theorem,

$$\begin{aligned} \max_{q \in \Delta\mathcal{D}} \mathrm{Risk}_q(\overline{p}) - \varepsilon &\leq \min_{p \in \Theta} \mathrm{Risk}_{\overline{q}}(p) + \varepsilon \\ &\leq \min_{p \in \Theta} \max_{q \in \Delta\mathcal{D}} \mathrm{Risk}_q(p) + \varepsilon. \end{aligned}$$

□

Algorithm 4 On-Demand Multi-Distribution Learning.

```
Input: Parameter space $\Theta$ with distance generating function $\omega$, distribution set $\mathcal{D}$ with $n := |\mathcal{D}|$, and loss function $\ell : \Theta \times \mathcal{Z} \to [0, L]$, all satisfying Definition A.13 and Assumptions 5 and 4;
Initialize: minimizing iterate $p^{(1)} = \Theta^o$ where $\theta^o$ is as defined in Definition A.9, maximizing iterate $q^{(1)} = [1/n]^n$, and iteration cap:
    $$T = \frac{36}{\varepsilon^2}\left(9C^2 D_\Theta + 8R_\Theta^2 C^2 \log\left(\frac{n+1}{\delta}\right) + 32L^2(n+2.1)\log\left(\frac{n+1}{\delta}\right)\right);$$
for $a = 1, 2, \ldots, \lceil T/n \rceil$ do
    For all $D \in \mathcal{D}$, sample datapoint $z_D^a$ from $\mathrm{EX}(D)$ ;
    for $t = an + 1 - n, \ldots, an$ do
        Sample datapoint $z^{(t)}$ from $\mathrm{EX}(q^{(t)})$;
        Define the estimates $\widehat{g}_{\text{-}}^{(t)} = \partial_\theta \ell(p^{(t)}, z^{(t)})$ and $\widehat{g}_{+}^{(t)} = [\ell(p^{(t)}, z_D^a)]_{D \in \mathcal{D}}$;
        Update iterates: $p^{(t+1)} = \mathcal{Q}_{\mathrm{omd},\omega}\left(p^{(t)}, \widehat{g}_{\text{-}}^{(t)}\right), q^{(t+1)} = \mathcal{Q}_{\mathrm{hedge}}\left(q^{(t)}, \widehat{g}_{+}^{(t)}\right)$;
    end for
end for
Return: parameter $\overline{\theta} := \frac{1}{T}\sum_{t=1}^{T} p^{(t)} \in \Theta$.
```

Theorem B.2 (Generalization of Theorems 4.1 and 5.1)**.** *Algorithm 4 is an (ε, δ) multi-distribution learning algorithm for any convex multi-distribution learning problem $(\Theta, \mathcal{D}, \ell)$ satisfying Definitions A.12 and A.13 and Assumptions 5 and 4. In other words, Algorithm 4 returns an $\overline{\theta} \in \Theta$ such that:*

$$\max_{D \in \mathcal{D}} \mathrm{Risk}_D(\overline{\theta}) \leq \inf_{\theta^* \in \Theta} \max_{D \in \mathcal{D}} \mathrm{Risk}_D(\theta^*) + \varepsilon.$$

Furthermore, the sample complexity of Algorithm 4 is in $\mathcal{O}\left(\frac{D_\Theta C^2 + (R_{\text{-}}^2 C^2 + nL^2)\log(n/\delta)}{\varepsilon^2}\right)$.

Proof. The sample complexity of Algorithm 4 is immediate from its construction. Every period a, Algorithm 4 samples n datapoints. Every iteration t, Algorithm 4 samples 1 datapoint. Thus, Algorithm 4 samples $2n\lceil T/n \rceil$ datapoints exactly.

We now prove that Algorithm 4 is an (ε, δ)-learning algorithm for any convex multi-distribution learning problem. We begin by constructing the following convex-concave game (A_-, A_+, ϕ) where:

$$A_- = \Theta, \quad A_+ = \Delta\mathcal{D}, \quad \phi(p, q) = \mathrm{Risk}_q(p).$$

We observe that this game satisfies Definition A.1 and Assumptions 1 and 2:

1. Definition A.13 defines $\mathrm{Risk}_q(\overline{p})$—and by extension $\phi(p, q)$—to be convex in p.
2. As $\mathrm{Risk}_q(\overline{p}) \coloneqq \sum_{D\in\mathcal{D}} q_D \mathrm{Risk}_D(p)$ by definition, $\phi(p, q)$ is linear and thus also concave in q.
3. In the l-1 norm, $\Delta\mathcal{D}$ satisfies Assumption 1 with diameter $R_+ = 2$.
4. Since Θ has finite Bregman radius of D_Θ by Assumption 5 and ω is strongly convex modulus 1 by definition, Θ satisfies Assumption 1 with a finite $R_\Theta \leq 2\sqrt{2D_\Theta}$.
5. Since $\partial_q\phi(p, q) = [\mathrm{Risk}_D(p)]_{D\in\mathcal{D}}$ and the range of ℓ is $[0, L]$, Assumption 2 is satisfied for $\partial_q\phi(p, q)$ by $C_+ \leq L$ in the l-infinity norm.
6. $\partial_p\phi(p, q)$ satisfies Assumption 2 for some finite $C_- = C$ directly by Assumption 4.

We now define a stochastic setting for our game. Let the minimizing player's randomness source be given by the sequences $\boldsymbol{\xi}^q_- = \{\mathrm{EX}(q)_i\}_{i=1}^{\infty}$; recall that $\mathrm{EX}(D)_k$ refers to the kth call to an example oracle for a $D \in \Delta\mathcal{D}$. Let the maximizing player's randomness source be given by the sequence $\boldsymbol{\xi}^\perp_+ = \{[\mathrm{EX}(D)_i]_{D\in\mathcal{D}}\}_{i=1}^{\infty}$. Next, define the first-order oracle estimators:

$$\widehat{g}_-(\xi^q_{-;i}, p, q) = \partial_p\ell\left(p, \xi^q_{-;i}\right), \quad \widehat{g}_+(\xi^\perp_{+;i}, p, q) = [\ell\left(p, (\xi^\perp_{+;i})_D\right)]_{D\in\mathcal{D}}.$$

We can observe that $\widehat{g}_-$ is a locally unbiased first-order oracle (satisfying Definition A.7) and $\widehat{g}_+$ is a globally unbiased first-order oracle (satisfying Definition A.8), with both $\widehat{g}_-$ and $\widehat{g}_+$ satisfying Assumptions 3.

1. By the unbiasedness of empirical risk estimates, $\widehat{g}_+$ is globally unbiased as returns an empirical risk sample for each $D \in \mathcal{D}$. Similarly, by the unbiasedness of empirical risk estimates and linearity of derivatives, $\widehat{g}_-$ is locally unbiased.
2. As the range of loss function ℓ is in $[0, L]$, empirical loss is also bounded in $[0, L]$, $\widehat{g}_+$ satisfies Assumption 3 with $C_+ \leq L$ in the l-infinity norm.
3. By Assumption 4, empirical partial subgradients are norm-bounded by some finite C, so $\widehat{g}_-$ satisfies Assumption 3 with some finite $C_- = C$.

Finally, we observe that Algorithm 4 is equivalent to instantiating Algorithm 3 on our constructed game (A_-, A_+, ϕ) for our constructed stochastic setting.

We will now rewrite the iteration complexity requirement of Lemma B.1 given by Equation 10 (copied below):

$$T \geq \frac{9}{\varepsilon'^2}\left(\gamma_-(T, A_-, 2C_-) + 8R_-^2C_-^2\log\left(\frac{r+1}{\delta}\right) + \gamma_+(T, A_+, 2C_+) + \frac{8R_+^2C_+^2r^2}{r+1}\log\left(\frac{r+1}{\delta}\right)\right).$$

In particular, we aim to show that the default iteration setting of Algorithm 4 satisfies it for $\varepsilon' = \varepsilon/2$.

By Lemmas B.4 and B.9, we can bound the efficacy of our no-regret algorithms $\mathcal{Q}_{\mathrm{omd},,}\mathcal{Q}_{\mathrm{hedge}}$ by:

$$\gamma_-(T, A_-, C_-) \leq \frac{9C_-^2D_\Theta}{4}, \quad \gamma_+(T, A_+, C_+) \leq \frac{9C_+^2\log n}{4},$$

where $\gamma_-(T, A_-, C_-)$ and $\gamma_+(T, A_+, C_+)$ are as defined in Lemma B.1.

Accounting for our previous derivations of $C_-, C_+, \varepsilon', R_+$, to satisfy Equation 10, it suffices to set:

$$T \geq \frac{9\cdot 4}{\varepsilon^2}\left(\frac{36C^2D_\Theta}{4} + 8R_\Theta^2C^2\log\left(\frac{n+1}{\delta}\right) + \frac{9L^2\log n}{4} + \frac{8\cdot L^2 4n^2}{n+1}\log\left(\frac{n+1}{\delta}\right)\right),$$

or simplified further:

$$T \geq \frac{36}{\varepsilon^2}\left(9C^2 D_\Theta + 8R_\Theta^2 C^2 \log\left(\frac{n+1}{\delta}\right) + 32L^2(n+2.1)\log\left(\frac{n+1}{\delta}\right)\right).$$

Thus, by Lemma B.1, $\overline{q} \coloneqq \frac{1}{T}\sum_{t=1}^{T} q^{(t)}$ and $\overline{\theta}$, the output of Algorithm 4, form an $\frac{\varepsilon}{2}$-min-max equilibria of our game $(A_{\text{-}}, A_{+}, \phi)$ with probability at least $1-\delta$. The Theorem then follows by Fact B.5. □

The following theorems, which are restated from the main text, are immediate corollaries of Theorem B.2.

Theorem 4.1. *For any finite hypothesis class $\mathcal{H}$ and unknown set of distributions $\mathcal{D}$, with probability $1-\delta$, Algorithm 2 returns a distribution $\overline{p} \in \Delta\mathcal{H}$ such that*

$$\mathop{\mathbb{E}}_{h\sim\overline{p}}\left[\max_{D\in\mathcal{D}}(h)\right] \leq OPT + \varepsilon \quad \textit{and} \quad \max_{D\in\mathcal{D}}(h_{\overline{p}}^{Maj}) \leq 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{\log|\mathcal{H}|+n\log(n/\delta)}{\varepsilon^2}\right)$.

Proof. Observe that this finite multi-distribution learning problem can be re-written as the convex multi-distribution learning problem $(\Delta\mathcal{H}, \mathcal{D}, \ell)$. Since $\Delta\mathcal{H}$ is a probability simplex of dimension $|\mathcal{H}|$, we know it is compact, convex, with $C = 1$ and $L = 1$ (as the range of ℓ is in $[0,1]$), and with $R_\Theta \leq 2$. We can then directly apply Theorem B.2, observing that Algorithm 2 is equivalent to Algorithm 4 in this setting. □

Theorem 5.1. *Consider a group distributionally robust problem $(\Theta, \mathcal{D})$ with convex compact unit-diameter parameter space Θ of Bregman radius D_Θ (Definition A.11), and convex loss $\ell : \Theta \times \mathcal{Z} \to [0, C]$. A variant of Algorithm 2 (in particular Algorithm 4 in Appendix 4.1), returns $\overline{\theta} \in \Theta$ such that $\max_{D\in\mathcal{D}} \mathbb{E}_{z\sim D}\left[\ell(\theta, z)\right] \leq$ R-$OPT + \varepsilon$, using a number of samples that is $\mathcal{O}\left(\frac{D_\Theta C^2 + nC^2\log(n/\delta)}{\varepsilon^2}\right)$.*

Proof. Similarly to Theorem 4.1, this claim follows immediately from Theorem B.2 for unit diameter $R_\Theta = 1$ and loss bound $L = C$. □

Corollary 5.2 follows in a similar fashion, running Algorithm 4 on empirical data distributions. The following proposition re-states this formally.

Proposition B.1 (Generalization of Corollary 5.2)**.** *Let $(\Theta, \mathcal{D}, \ell)$ be a convex multi-distribution learning problem satisfying Definitions A.12 and A.13 and Assumptions 5 and 4. For every $D \in \mathcal{D}$, let $\mathbf{B}_D \sim D$ be a non-empty batch of i.i.d. datapoint samples. Define $\mathcal{D}' = \{D'\}_{D\in\mathcal{D}}$, where D' is the empirical distribution of $\mathbf{B}_D$. It follows that $(\Theta, \mathcal{D}', \ell)$ also satisfies Definitions A.12 and A.13 and Assumptions 5 and 4 with identical parameters. Thus, Algorithm 4, when applied to $(\Theta, \mathcal{D}', \ell)$, with probability at least $1-\delta$ returns an $\overline{\theta}$ with a multi-distribution training error of at most ε. Furthermore, the number of iterations—and accordingly partial derivative operations—is in $\mathcal{O}\left(\frac{D_\Theta C^2 + (R_{-}^2 C^2 + nL^2)\log(n/\delta)}{\varepsilon^2}\right)$.*

B.3 Proof of Theorem 4.3

We now provide matching lower bounds for collaborative PAC learning.

We first define a notion of expected sample complexity. Take any multi-distribution learning problem $V = (\Theta, \mathcal{D}, \ell)$. Recall that, on this problem, the input to any multi-distribution learning algorithm is a random variable of form $\widehat{V} = (\Theta_i, \ell_i, \{\mathrm{EX}(D)\}_{D \in \Delta\mathcal{D}})$. Also recall that each example oracle $\mathrm{EX}(D)$ is an infinite sequence of i.i.d. samples from D. We will let X_V denote the probability distribution of the random variable tuple $(\Theta_i, \ell_i, \{\mathrm{EX}(D)\}_{D \in \Delta\mathcal{D}})$. Further let $N_A(\widehat{V})$ denote the expected sample complexity of A given inputs $\widehat{V}$, where expectation is taken over any randomness from the algorithm A itself. We can now define a general notion of expected sample complexity.

Definition B.1. *Let A be a multi-distribution learning algorithm and $\mathbb{P}$ a probability distribution over a set of multi-distribution learning problems $\mathbb{V} := \{(\Theta_i, \mathcal{D}_i, \ell_i)\}_i$. We define the expected sample complexity $N_A(\mathbb{P})$ as:*

$$N_A(\mathbb{P}) = \mathop{\mathbb{E}}_{V \sim \mathbb{P}} \left[\mathop{\mathbb{E}}_{\widehat{V} \sim X_V} \left[N_A(\widehat{V}) \right] \right].$$

The outer expectation is taken over the randomness of the problem selection, the inner expectation is taken over the randomness of datapoints, and $N_A(\widehat{V})$ takes an expectation over the internal randomness of the algorithm A.

Unless otherwise specified, we will use the shorthand: $\mathbb{E}_{V \sim \mathbb{P}} \left[\mathbb{E}_{\widehat{V} \sim X_V} [\cdot] \right] = \mathbb{E}_{\widehat{V}} [\cdot]$. We recall the following theorem from Section 4.2.

Theorem 4.3. *Take any $n, d \in \mathbb{Z}_+$, $\varepsilon, \delta \in (0, 1/8)$, and (ε, δ)-collaborative learning algorithm A. There exists a collaborative learning problem $(\mathcal{H}, \mathcal{D})$ with $|\mathcal{D}| = n$ and $|\mathcal{H}| = 2^d$, on which A takes at least $\Omega\left(\frac{1}{\varepsilon^2}\left(\log|\mathcal{H}| + |\mathcal{D}| \log(\min\{|\mathcal{D}|, \log|\mathcal{H}|\}/\delta)\right)\right)$ samples.*

We now prove two lemmas, Lemma B.3 and Lemma B.4, that directly imply Theorem 4.3.

Lemma B.3. *Take any $n, d \in \mathbb{Z}_+$, $\varepsilon, \delta \in (0, 1/8)$, and (ε, δ)-collaborative learning algorithm A. There exists a set of collaborative learning problems $\mathbb{V}$ on which A 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 lower bound on sample complexity of agnostic probably-approximately-correct (PAC) learning [29]. Take a set of PAC learning problems $\mathbb{V}'$ where every problem in $\mathbb{V}'$ shares a hypothesis set $\mathcal{H}'$ where $|\mathcal{H}'| = 2^d$, feature space $\mathcal{X}'$, and binary label space $\mathcal{Y}' = \{0, 1\}$. Note that $|\mathbb{V}'|$ is necessarily finite. We define the following class of collaborative learning problems. Let $\mathcal{X} = \mathcal{X}' \bigcup \{\perp\}$, where $\perp$ is some special symbol. Let $\mathcal{H} = \{g_{h'} \mid h' \in \mathcal{H}'\}$, where, for any $x \in \mathcal{X}$, we define $g_{h'}(x)$ as:

$$g_{h'}(x) = \begin{cases} 1 & x = \perp \\ h'(x) & \text{otherwise} \end{cases}$$

Define D^* as a degenerate probability distribution over $\mathcal{X} \times \mathcal{Y}$ where $\Pr_{D^*}(\perp, 1) = 1$. For any PAC learning problem $(\mathcal{H}', D') \in \mathbb{V}'$ we can thus define the collaborative learning problem $(\mathcal{H}, \mathcal{D}')$, where $\mathcal{D}' = \left\{\widehat{D'}\right\} \bigcup \{D^*\}_{i=1}^{n-1}$ and $\widehat{D'}$ is an adaptation of D' onto the support $\mathcal{X} \times \mathcal{Y}$. Accordingly, define $\mathbb{V} = \{(\mathcal{H}, \mathcal{D}') \mid (\mathcal{H}', D') \in \mathbb{V}'\}$.

It is not hard to see that any (ε, δ) collaborative learning algorithm A for $\mathbb{V}$ is also an (ε, δ) PAC learning algorithm A' for $\mathbb{V}'$ with an identical sample complexity. We can construct A' using A as follows. Suppose we are given a $(\mathcal{H}', D') \in \mathbb{V}'$. Construct $(\mathcal{H}, \left\{\widehat{D'}\right\} \bigcup \{D^*\}_{i=1}^{n-1})$ as described. Run A on $(\mathcal{H}, \mathcal{D}')$, simulating data draws from $\widehat{D'}$ by drawing from D'. Letting $h \in \mathcal{H}$ denote the output of A (if it terminates), return $h' \in \mathcal{H}'$ where $h'(x) = h(x)$ for every $x \in \mathcal{X}'$. Since with probability at least $1 - \delta$, A successfully outputs a hypothesis h where:

$$\mathrm{Risk}_{\widehat{D'}}(h) = \max_{D \in \mathcal{D}'} \mathrm{Risk}_D(h) \leq \min_{h^* \in \mathcal{H}} \max_{D \in \mathcal{D}'} \mathrm{Risk}_D(h^*) + \varepsilon = \min_{h^* \in \mathcal{H}} \mathrm{Risk}_{\widehat{D'}}(h^*) + \varepsilon,$$

meaning our output h' also satisfies:

$$\text{Risk}_{D'}(h') \leq \min_{h^* \in \mathcal{H}'} \text{Risk}_{D'}(h^*) + \varepsilon.$$

Finally, we can invoke the well-known lower bound of agnostic PAC learning to observe that there exists an agnostic PAC learning problem $\mathbb{V}'$ such that any (ε, δ)-learning algorithm has a sample complexity of $\Omega\left(\frac{\log|\mathcal{H}|}{\varepsilon^2}\right)$ [11]; we defer interested readers to Zhang [35] for a constructive proof of the existence of $\mathbb{V}'$. Thus, there exists a $\mathbb{V}$ satisfying the assumptions of Lemma B.3 where any (ε, δ) collaborative learning algorithm has a sample complexity of $\Omega\left(\frac{\log|\mathcal{H}|}{\varepsilon^2}\right)$. □

Lemma B.4. *Take any $n, d \in \mathbb{Z}_+$, $\varepsilon, \delta \in (0, 1/8)$, and (ε, δ)-collaborative learning algorithm A. There exists a set of collaborative learning problems $\mathbb{V}$ on which A takes at least $\Omega\left(\frac{1}{\varepsilon^2}\left(|\mathcal{D}| \log(k/\delta)\right)\right)$ samples and where, for every $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}$, $|\mathcal{D}| = n$ and $|\mathcal{H}| = 2^k$ with $k := \min\{n, d\}$.*

Proof. We prove this constructively. Fix a choice of $n, d \in \mathbb{Z}_+$ and $\varepsilon, \delta \in (0, 1/8)$. We begin by defining collaborative learning problem sets $\mathbb{V}_{u,w}$ for all $u, w \in \mathbb{N}$ with $u \geq w$. A problems in $\mathbb{V}_{u,w}$ share an feature space $\mathcal{X} = \{1, \ldots, w\}$, label space $\mathcal{Y} = \{+, -\}$, and hypothesis class $\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y}\}$ (the set of all deterministic binary labeling functions). For every $i \in [u]$, we define distributions D_i^- and D_i^+ as:

$$\Pr_{D_i^-}(i, -) = \Pr_{D_i^+}(i, +) = \frac{1}{2} + \varepsilon \text{ and } \Pr_{D_i^-}(i, +) = \Pr_{D_i^+}(i, -) = \frac{1}{2} - \varepsilon.$$

Now define $\mathbb{Q}$ as a distribution over w-length strings where each character is independently uniformly sampled from $\{+, -\}$. We then define $\mathbb{P}_{u,w}$ as a distribution over collaborative learning problems, where for any w-length string b, $\mathbb{P}_n\left(\left(\mathcal{H}, \left\{D_{i \pmod{w}}^{b_i \pmod{w}}\right\}_{i=1}^{u}\right)\right) = \mathbb{Q}(b) = 2^{-u}$. Finally, we define $\mathbb{V}_{u,w}$ to be the support of $\mathbb{P}_{u,w}$.

Claim B.1. *Take any (ε, δ)-learning algorithm A on $\mathbb{V}_{1,1}$. Then, the expected sample complexity (see Definition B.1) of A on $\mathbb{P}_{1,1}$ is at least $\frac{1}{8\varepsilon^2} \log \frac{1}{\delta}$.*

Proof. When $u = w = 1$, the distributions $\Pr_{D_1^+}, \Pr_{D_1^-}$ are simply Bernoulli distributions with parameters $p_1 = \frac{1}{2} + \varepsilon$, $p_2 = \frac{1}{2} - \varepsilon$ respectively. We will first prove that any algorithm given only N datapoints will result in an at least ε-error output with probability at least $\frac{1}{4} \exp\left(-8\varepsilon^2 N\right)$. We will then relax these qualifiers on A to recover our claim.

Let the random variables $X_1^N, X_2^N, X_3^N, X_0^N$ each represent N i.i.d. samples from Bernoulli distributions with parameter $p = \frac{1}{2} + \varepsilon$, $p = \frac{1}{2} - \varepsilon$, $p = \frac{1}{2} + 2\varepsilon$, $p = \frac{1}{2}$ respectively. We will show that there is no function $f : \{0, 1\}^N \to \{0, 1\}$ that can reliably output 1 when given a realization of X_1^N and reliably output 0 when given a realization of X_2^N. Let $\alpha = \mathbb{E}_{D_1}\left[f(X_1^N)\right]$ and $\beta = \mathbb{E}_{D_2}\left[f(X_2^N)\right]$. We can write the KL divergence between the Bernoulli random variables $f(X_1^N)$ and $f(X_2^N)$ as:

$$\text{KL}\left(f(X_1^N) || f(X_2^N)\right) = \alpha \log \frac{\alpha}{\beta} + (1 - \alpha) \log \frac{1 - \alpha}{1 - \beta}. \tag{17}$$

Since $\alpha \log \alpha + (1 - \alpha) \log(1 - \alpha) \geq -\log 2$, we have that,

$$\text{KL}\left(f(X_1^N) || f(X_2^N)\right) \geq (\alpha \log \frac{1}{\beta} + (1 - \alpha) \log \frac{1}{1 - \beta} - \log 2).$$

Applying in order the data processing inequality, the tensorization property of KL divergence, and the fact that $\text{KL}\left(X_1^1 || X_2^1\right) \leq \text{KL}\left(X_3^1 || X_0^1\right)$, we have that:

$$\text{KL}\left(f(X_1^N) || f(X_2^N)\right) \leq \text{KL}\left(X_1^N || X_2^N\right) = N \cdot \text{KL}\left(X_1^1 || X_2^1\right) = \frac{-N}{2} \log(1 - \varepsilon^2) \leq 16N\varepsilon^2. \tag{18}$$

Recall that α is the probability that f is correct given $p = \frac{1}{2} + \varepsilon$ and β is the probability that f is wrong given $p = \frac{1}{2}$. Combining Equations 17 and 18, we see that either $\alpha < 1/2$ or $\beta \geq \frac{1}{4} \exp\left(-8\varepsilon^2 N\right)$. Thus, for any algorithm A outputting deterministic classifiers, there exists a problem $V \in \mathbb{V}_{1,1}$ such that when A is given only N samples, its output classifier, with probability at least $\frac{1}{4}\exp\left(-8\varepsilon^2 N\right)$, has an risk of at least ε. By the minimax risk inequality, this also statement extends to algorithms returning random classifiers.

Now consider an (ε, δ)-learning algorithm A for $\mathbb{V}_{1,1}$. Recalling notation from Definition B.1, $N_A(\widehat{V})$ is a random variable denoting the number of samples taken by A. Let E denote the event that A returns a classifier with error at least ε. By linearity of expectation, we can lower bound our failure probability as:

$$\begin{aligned}
\delta &\geq \mathop{\mathbb{E}}_{\widehat{V}}\left[\sum_{N=0}^{\infty} \Pr\left(N_A(\widehat{V}) = N \mid V\right) \Pr\left(E \mid N_A(\widehat{V}) = N, V\right)\right] \\
&\geq \frac{1}{2}\sum_{N=0}^{\infty} \Pr(N_A(\widehat{V}) = N \mid V) \cdot \frac{1}{4}\exp\left(-8\varepsilon^2 N\right) \\
&= \mathop{\mathbb{E}}_{\widehat{V}}\left[\frac{1}{4}\exp\left(-8\varepsilon^2 N_A(\widehat{V})\right)\right]
\end{aligned}$$

Observing that $h(x) = \log_{\frac{1}{4}\exp(-8\varepsilon^2)}(x)$ is convex for $\varepsilon \in (0, 1/8)$, by Jensen's inequality:

$$\mathop{\mathbb{E}}_{\widehat{V}}\left[N_A(\widehat{V})\right] \geq h\left(\mathop{\mathbb{E}}_{\widehat{V}}\left[\frac{1}{4}\exp\left(-8\varepsilon^2 N_A(\widehat{V})\right)\right]\right) \geq h\left(\delta\right) \geq \frac{\log(1/\delta)}{\log(4\exp(8\varepsilon^2))} \geq \frac{\log(1/\delta)}{8\varepsilon^2}.$$

□

Claim B.2. *Suppose there exists an (ε, δ) learning algorithm A for $\mathbb{V}_{n,k}$. Further suppose A has an expected sample complexity on $\mathbb{P}_{n,k}$ of m. Then there exists an $(\varepsilon, \frac{10\delta}{9k})$ learning algorithm A' for $\mathbb{V}_{1,1}$ with an expected sample complexity on $\mathbb{P}_{1,1}$ of $\frac{10}{9n}m$.*

Proof. This claim closely follows the proofs of Claims 4.3 and 4.4 in Blum et al. [4], and is included for completeness. We construct A' as follows. Suppose a problem $V' \in \mathbb{V}_{1,1}$ is drawn.

1. A' draws a problem $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}_{n,k}$ and chooses an index $i \in [n]$ uniformly at random.

2. A' simulates the algorithm A on $(\mathcal{H}, \mathcal{D})$; when A tries to sample a datapoint from the ith distribution D_i, return a sampled datapoint from the data distribution of V'.

3. When A terminates and returns a classifier h, A' checks whether, for every $j \neq i$: $\text{Risk}_{D_j}(h) < \frac{1}{2}$. If this condition is satisfied, A' returns $h'(1) = h(\ell)$. If not, we repeat from Step 1. We denote the number of total iterations with T.

Consider the probability p_i that, in the third step, for every $j \neq i$ we have $\text{Risk}_{D_j}(h) < \frac{1}{2}$ but $\text{Risk}_{D_i}(h) \geq \frac{1}{2}$. Let E_t denote the event that A' returns an at least ε-error hypothesis after t iterations of our procedure. Noting that E_t can only occur if A failed all $t-1$ iterations before and at the tth iteration, Step 3 fails to catch the bad hypothesis for D_i. By assumption, $\delta \geq \sum_{i=1}^{k} p_i$. By symmetry of our construction $\mathbb{V}$:

$$\sum_{t=1}^{\infty} \Pr\left(E_t\right) \leq \sum_{t=1}^{\infty} \delta^{t-1}\frac{1}{k}\sum_{i=1}^{k} p_i \leq \sum_{t=1}^{\infty} \delta^t / k \leq \frac{10\delta}{9k}$$

Thus, A' is an $(\varepsilon, \frac{10\delta}{9k})$-algorithm for $\mathbb{P}_{1,1}$.

We now bound the sample complexity of A'. Let $N_{A'}(t)$ denote the number of samples that A' takes from V' on the tth iteration. Note that $N_{A'}(1), N_{A'}(2), \ldots$ are i.i.d. In addition, by the symmetry of $\mathbb{V}$ and linearity of expectation, $\mathbb{E}_{V' \in \mathbb{P}_{1,1}}\left[N_{A'}(t)\right] = m/n$. Thus, we can write:

$$\mathop{\mathbb{E}}_{\widehat{V'}}\left[\sum_{t=1}^{T} N_{A'}(t)\right] = \mathop{\mathbb{E}}_{\widehat{V'}}\left[T\right] \mathop{\mathbb{E}}_{\widehat{V'}}\left[N_{A'}(1)\right] = \mathop{\mathbb{E}}_{\widehat{V'}}\left[T\right] m/n.$$

We can upper bound T by observing that our procedure only repeats if A fails. Thus,

$$\mathop{\mathbb{E}}_{\widehat{V'}}\left[T\right] = \sum_{t=1}^{\infty} \Pr(T \geq t) \leq \sum_{t=0}^{\infty} \delta^t \leq \frac{1}{1-\delta} \leq \frac{10}{9}.$$

Thus, A' has an expected sample complexity of at most $\frac{10m}{9n}$. □

Combining claims B.1 and B.2, we see that any (ε, δ) collaborative learning algorithm A for $\mathbb{V}_{n,k}$ has an expected sample complexity on $\mathbb{P}_{n,k}$ of at least $m \geq \frac{9n}{80\varepsilon^2} \log\left(\frac{9k}{10\delta}\right)$. By the probabilistic method, for at least some collaborative learning problem $V \in \mathbb{V}_{n,k}$, A must have a sample complexity in $\Omega\left(\frac{9n}{80\varepsilon^2} \log\left(\frac{9k}{10\delta}\right)\right)$. Recall that each collaborative learning problem $(\mathcal{H}, \mathcal{D}) \in \mathbb{V}_{n,k}$ has $|\mathcal{D}| = n$ and $|\mathcal{H}| = 2^k$, satisfying our Lemma B.4 requirements. □

The following is a more general restatement of Theorem 4.3 in terms of the terminology of Section A. It follows by observing the difficult cases described in Theorem 4.3 constitute challenging cases for both convex multi-distribution learning (Definition A.13) and binary classifier multi-distribution learning (Definition A.15).

Corollary B.5. *Take any $n, m \in \mathbb{N}$ and $\varepsilon, \delta \in (0, 1/8)$. There exists a finite set $\mathbb{V}$ of multi-distribution learning problems where:*

1. *Every $(\Theta, \mathcal{D}, \ell) \in \mathbb{V}$ satisfies Definitions A.13 and A.15, with $|\mathcal{D}| = n$ and $D_\Theta = \log(m)$.*
2. *Every (ε, δ)-algorithm A has a sample complexity in $\left(\frac{D_\Theta + n \log(\min\{n, D_\Theta\}/\delta)}{\varepsilon^2}\right)$.*

B.4 Proof of Lemmas B.4 and B.9

For completeness, this section includes standard results on exponentiated gradient descent and mirror descent more generally, proofs of which can be found in [30].

Lemma B.6. ***Pinsker's inequality*** *For any two vectors from the same probability simplex $w, w' \in \Delta^n$, we can bound their generalized Kullback-Leibler divergence as,*

$$KL\left(w||w'\right) \geq \frac{1}{2}\left\|w - w'\right\|_1^2.$$

Lemma B.7. ***Law of Cosines*** *Define $x, y, z \in Z$ where Z is a convex set, and let $H : Z \to \mathbb{R}$ be a strictly convex differentiable distance generating function and D_H the Bregman divergence of H. Then,*

$$\langle \nabla H(y) - \nabla H(z), y - x\rangle = D_H(x, y) + D_H(y, z) - D_H(x, z).$$

Lemma B.8. ***Pythagorean theorem for Bregman Divergence*** *Define $x, y \in Z^0$ where Z^0 is a closed subset of a convex set Z, $z \in Z$. Let $H : Z \to \mathbb{R}$ be a strictly convex differentiable distance generating function, and let D_H be the Bregman divergence of H. If $y = \arg\min_{u \in Z^0} D_H(u, z)$, then $D_H(x, y) + D_H(y, z) \leq D_H(x, z)$.*

We now turn to proving Lemma (restated below), which concerns exponentiated gradient descent with bounded gradients.

Lemma 2.1 ([30])**.** *Let $g^{(1)}, \ldots, g^{(T)} \in \mathbb{R}^n$ and $Z = \Delta^n$. Further assume $\left\|g^{(t)}\right\|_\infty \leq C$ for all timesteps $t = 1, \ldots, T$. Choosing $\eta = \sqrt{\log n / T}$, after T iterations of exponential gradient descent, the output $\{w\}_{t=1}^T$ satisfies,*

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \leq \frac{3C}{2}\sqrt{\frac{KL\left(w^{(T)}||w^{(1)}\right)}{T}}.$$

Proof. This proof closely follows that of Theorem 7.5 in [30]. Fix $t \in 1, \ldots, T$. First, we provide an expression for $g^{(t)}$ in terms of $\widetilde{w}^{(t+1)}$ and $w^{(t)}$, where $\widetilde{w}$ is as defined in Equation 4. For all $i \in 1, \ldots, n$, we have by Equation 4:

$$\widetilde{w}_i^{(t+1)} = w_i^{(t)} \exp\left(-\eta g_i^{(t)}\right).$$

Equivalently,

$$g_i^{(t)} = \frac{1}{\eta}\left(\log w_i^{(t)} - \log \widetilde{w}_i^{(t+1)}\right).$$

Letting $H(x) = \sum_{i=1}^n x_i \log x_i - x_i$ denote our distance generating function, generalized negative entropy, we can also write,

$$g_i^{(t)} = \frac{1}{\eta}\left(\nabla H\left(w^{(t)}\right) - \nabla H\left(\widetilde{w}^{(t+1)}\right)\right),$$

where logs are applied coordinate-wise. Defining $\mathrm{KL}\left(\cdot||\cdot\right)$ as generalized Kullback–Leibler divergence: the Bregman divergence with respect to our choice of H as a distance generating function. Since Z is already closed, by Lemma B.8, for any $w^* \in Z$,

$$\begin{aligned}\left\langle g_i^{(t)}, w^{(t)} - w^*\right\rangle &= \frac{1}{\eta}\left\langle \nabla H\left(w^{(t)}\right) - \nabla H\left(\widetilde{w}^{(t+1)}\right), w^{(t)} - w^*\right\rangle \\ &= \frac{1}{\eta}\left(\mathrm{KL}\left(w^*||w^{(t)}\right) + \mathrm{KL}\left(w^{(t)}||\widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^*||\widetilde{w}^{(t+1)}\right)\right).\end{aligned}$$

Generalized Pythagorean Theorem, e.g. Theorem 7.7 in [30] gives,

$$\mathrm{KL}\left(w^*||\widetilde{w}^{(t+1)}\right) \geq \mathrm{KL}\left(w^*||w^{(t+1)}\right) + \mathrm{KL}\left(w^{(t+1)}||\widetilde{w}^{(t+1)}\right).$$

Then we can bound,

$$\begin{aligned}
\eta \sum_{t=1}^{T} \left\langle g_i^{(t)}, w^{(t)} - w^* \right\rangle &= \sum_{t=1}^{T} \mathrm{KL}\left(w^* || w^{(t)}\right) + \mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^* || \widetilde{w}^{(t+1)}\right) \\
\text{(By Pythagorean)} \quad &\leq \sum_{t=1}^{T} \mathrm{KL}\left(w^* || w^{(t)}\right) + \mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) \\
&\qquad - \left(\mathrm{KL}\left(w^* || w^{(t+1)}\right) + \mathrm{KL}\left(w^{(t+1)} || \widetilde{w}^{(t+1)}\right)\right) \\
&= \sum_{t=1}^{T} \mathrm{KL}\left(w^* || w^{(t)}\right) - \mathrm{KL}\left(w^* || w^{(t+1)}\right) \\
&\qquad + \left(\mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^{(t+1)} || \widetilde{w}^{(t+1)}\right)\right) \\
\text{(By telescoping)} \quad &\leq \mathrm{KL}\left(w^* || w^{(0)}\right) + \sum_{t=1}^{T} \mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^{(t+1)} || \widetilde{w}^{(t+1)}\right). \qquad (19)
\end{aligned}$$

To bound the second term, we again apply the law of cosines, this time in reverse order, recovering,

$$\mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^{(t+1)} || \widetilde{w}^{(t+1)}\right) = \eta \left\langle g^{(t)}, w^{(t)} - w^{(t+1)} \right\rangle - \mathrm{KL}\left(w^{(t+1)} || w^{(t)}\right).$$

As $w^{(t+1)}, w^{(t)} \in Z$, by Pinsker's inequality (Lemma B.6),

$$\begin{aligned}
\mathrm{KL}\left(w^{(t)} || \widetilde{w}^{(t+1)}\right) - \mathrm{KL}\left(w^{(t+1)} || \widetilde{w}^{(t+1)}\right) &\leq \eta \left\langle g^{(t)}, w^{(t)} - w^{(t+1)} \right\rangle - \frac{1}{2} \left\| w^{(t+1)} - w^{(t)} \right\|_1^2 \\
&\leq \eta \left\| g^{(t)} \right\|_\infty \left\| w^{(t)} - w_1^{(t+1)} \right\| - \frac{1}{2} \left\| w^{(t+1)} - w^{(t)} \right\|_1^2 \\
&\leq \eta C \left\| w^{(t)} - w_1^{(t+1)} \right\| - \frac{1}{2} \left\| w^{(t+1)} - w^{(t)} \right\|_1^2 \\
&\leq \frac{\eta^2 C^2}{2}, \qquad (20)
\end{aligned}$$

where the final inequality follows from maximizing the quadratic $\eta C z - \frac{z^2}{2}$, attained at $z = \left\| w^{(t)} - w^{(t+1)} \right\|_1 = C\eta$. The claim follows by plugging Equation 20 into Equation 19. □

Exponentiated gradient descent is a special case of mirror descent in the Euclidian space $\mathcal{E} = \mathbb{R}^n$ equipped with an L1-norm $\|\cdot\|_1$, over the probability simplex $Z = \Delta^n$, and using entropy as a distance generating function. The following lemma generalizes Lemma B.4 to more general Euclidian spaces, choices of convex compact subsets, and strongly-convex distance generating functions. As the proof closely mirrors that of Lemma B.4, we defer interested readers to Beck and Teboulle [2].

Lemma B.9 (Generalization of Lemma B.4 [2])**.** *Let Z be a convex compact subset of a Euclidean space $\mathcal{E}$ with distance generating function ω satisfying Definition A.9. Let $g^{(1)}, \ldots, g^{(T)} \in \mathcal{E}^*$. Further assume $\left\| g^{(t)} \right\|_{\mathcal{E}^*} \leq C$ for all timesteps $t = 1, \ldots, T$. Choose step size $\eta = \sqrt{\frac{D_Z}{T}}$ where D_Z is the Bregman radius of Z. After T iterations of online mirror descent [2], the output $\{w\}_{t=1}^{T}$ satisfies,*

$$\mathrm{Err}_{\mathrm{V}}(w^{(1:T)}) \leq \frac{3C}{2} \sqrt{\frac{D_Z}{T}}.$$

B.5 Proof of Theorem 4.2

This section discusses the implications of our results for finite VC dimension problems.

First, we will use $D_{\mathcal{X}}$ to denote the marginal distribution of a data distribution D, i.e. the distribution of D over its feature space. We also introduce the following definitions.

Definition B.2. *The Renyi divergence $D_\alpha(P||Q)$ between discrete distributions P, Q is defined by:*

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log_2 \sum_{x \in \mathcal{X}} P(x) \left(\frac{P(x)}{Q(x)} \right)^{\alpha - 1}$$

and between continuous distributions P, Q as:

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log_2 \int_{\mathcal{X}} P(x) \left(\frac{P(x)}{Q(x)} \right)^{\alpha - 1} dx.$$

We will write $d_\alpha(P||Q) \coloneqq 2^{D_\alpha(P||Q)}$.

Remark B.1. *Denoting the support of Q as $\mathcal{X}_Q$, we can write $\sup_{x \in \mathcal{X}_Q} \frac{P(x)}{Q(x)} = d_\infty(P||Q)$.*

Recall that in Theorem B.2 we describe a multi-distribution learning algorithm (Algorithm 4) with provably tight sample complexity upper bounds for convex multi-distribution learning problems (Definition A.13). We note that there is one class of multi-distribution learning problems, *non-convex finite VC multi-distribution learning*, that has been previously studied by [4, 22, 6] but does not satisfy the assumptions of convex multi-distribution learning. A *non-convex finite VC multi-distribution learning* problem is a binary-classification multi-distribution learning problem (Definition A.15) that satisfies three criteria: the hypothesis space $\mathcal{H}$ is non-convex, of infinite size, and of finite VC dimension $\text{VCD}(\mathcal{H}) < \infty$. [4, 22, 6] provide upper bounds for non-convex finite VC multi-distribution learning that are identical to their upper bounds in Table 1 but replacing $\log |\mathcal{H}|$ with $\text{VCD}(\mathcal{H})$.

In contrast, our Theorem B.2 upper bounds do not directly apply to non-convex finite VC multi-distribution learning. However, a similar result can be obtained by running our Algorithm 4 on a probability simplex $\Delta\mathcal{H}'$ over some ε-covering $\mathcal{H}'$ of $\mathcal{H}$. Such ε-nets of size $n\varepsilon^{-2\text{VCD}(\mathcal{H})}$ necessarily exist (see, e.g., [1]). For example, given an ε-net for each distribution $D \in \mathcal{D}$, we may take their union as the covering $\mathcal{H}'$ and run Algorithm 2. This directly inherits a favorable upper bound from Theorem B.2.

Corollary B.10 (VC Dimension Corollary of Theorem B.2)**.** *Consider any binary classification multi-distribution learning problem $(\mathcal{H}, \mathcal{D})$ where the hypothesis set $\mathcal{H}$ is of finite VC dimension d and the unknown distribution set is of size $|\mathcal{D}| = n$. There is an algorithm that, given an ε-net of size* $\text{poly}\left(\varepsilon^d, \varepsilon, d, n\right)$ *for each distribution, with probability $1 - \delta$, returns a distribution $\overline{p} \in \Delta\mathcal{H}$ with,*

$$\mathop{\mathbb{E}}_{h \sim \overline{p}} \left[\max_{D \in \mathcal{D}}(h) \right] \leq OPT + \varepsilon \quad \text{and} \quad \max_{D \in \mathcal{D}}(h_{\overline{p}}^{Maj}) \leq 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{d \log(1/\varepsilon) + n \log(n/\delta)}{\varepsilon^2}\right)$.

It is also 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 $\mathcal{D}$. Theorem 4.2, restated below, formalizes this claim.

Theorem 4.2. *For any $\mathcal{H}$ of VC dimension d and unknown set of distributions $\mathcal{D}$ for which Assumption 1 or 2 is met, there is an algorithm that, with probability $1 - \delta$, returns a distribution $\overline{p} \in \Delta\mathcal{H}$ with,*

$$\mathop{\mathbb{E}}_{h \sim \overline{p}} \left[\max_{D \in \mathcal{D}}(h) \right] \leq OPT + \varepsilon \quad \text{and} \quad \max_{D \in \mathcal{D}}(h_{\overline{p}}^{Maj}) \leq 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{d \log(dn/\varepsilon) + n \log(n/\delta)}{\varepsilon^2}\right)$.

The following lemmas directly imply Theorem 4.2.

Lemma B.11 (Assumption 1)**.** *Consider any binary classification multi-distribution learning problem* $(\mathcal{H}, \mathcal{D})$ *where the hypothesis set* $\mathcal{H}$ *is of finite VC dimension* d *and the unknown distribution set is of size* $|\mathcal{D}| = n$*. 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 a distribution* $\overline{p} \in \Delta\mathcal{H}$ *with,*

$$\mathop{\mathbb{E}}_{h\sim\overline{p}}\left[\max_{D\in\mathcal{D}}(h)\right] \le OPT + \varepsilon \quad \textit{and} \quad \max_{D\in\mathcal{D}}(h_{\overline{p}}^{Maj}) \le 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{d\log(1/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.

Proof. By uniform convergence, sampling $\Theta\left(\frac{d\log(d/\varepsilon)+\log(1/\delta)}{\varepsilon^2}\right)$ i.i.d. samples from a distribution $D_\mathcal{X}$, with probability at least $1-\delta$, yields an ε-covering on D. By Sauer-Shelah's lemma, the resulting covering $\mathcal{H}'_D$ is of size $\mathcal{O}\left((\log(d/\varepsilon) + \frac{1}{d}\log(1/\delta)/\varepsilon^2)^d\right)$. Repeating this procedure for each $D \in \mathcal{D}$, with probability at least $1-n\delta$, we have an ε-covering of $\mathcal{D}$ of size $\mathcal{O}\left(n(\log(d/\varepsilon) + \frac{1}{d}\log(1/\delta)/\varepsilon^2)^d\right) \in \text{poly}\left(\varepsilon^d, \varepsilon, d, n\right)$ and can appeal to Corollary B.10. □

Lemma B.12 (Assumption 2)**.** *Consider any binary classification multi-distribution learning problem* $(\mathcal{H}, \mathcal{D})$ *where the hypothesis set* $\mathcal{H}$ *is of finite VC dimension* d *and the unknown distribution set is of size* $|\mathcal{D}| = n$*. 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 unlabeled access, with probability* $1-\delta$*, returns a distribution* $\overline{p} \in \Delta\mathcal{H}$ *with,*

$$\mathop{\mathbb{E}}_{h\sim\overline{p}}\left[\max_{D\in\mathcal{D}}(h)\right] \le OPT + \varepsilon \quad \textit{and} \quad \max_{D\in\mathcal{D}}(h_{\overline{p}}^{Maj}) \le 2OPT + \varepsilon,$$

using a number of samples that is $\mathcal{O}\left(\frac{d\log(dn/\varepsilon)+n\log(n/\delta)}{\varepsilon^2}\right)$.

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 Sauer-Shelah's lemma, the resulting covering $\mathcal{H}'_D$ is of size $\mathcal{O}\left((\text{poly}(1/\varepsilon, d, n))^d\right)$. 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 $|\mathcal{H}'| \in \mathcal{O}\left(n(\text{poly}(1/\varepsilon, d, n))^d\right)$. We can then appeal directly to Theorem 4.1 for a sample complexity bound on learning $(\mathcal{H}', \mathcal{D})$. □

C Experiment Details

Datasets Our experiments were performed on three datasets: Multi-NLI, CelebA, and Waterbirds [27]. We use identical preprocessing settings and dataset splits as Sagawa et al. [27]. Our experiments, unless otherwise specified, replicate the exact hyperparameter settings adopted by Sagawa et al. [27] 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. [27] or to our public source code for replication details.

The **Multi-NLI dataset** [32] 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 [32] while negation labels were annotated by Sagawa et al. [27]. 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 [17]. 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. [27], 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. [17] limited the "gender" attribute label to binary options of male and not male. There are 162,770 datapoints available in 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. [27] curated from a larger Caltech-UCSD Birds-200-2011 (CUB) dataset [31]. 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 [15] and BERT [9]. We use the *torchvision* [19] 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* [33] 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 (resampling-based GDRO). 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-case 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.