On-Demand Sampling: Learning Optimally from Multiple Distributions

Anonymous Author(s)
Affiliation
Address
email

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* [28], and *fair federated* learning [21]. 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. [21] by a multiplicative factor of n, the sample complexity of collaborative learning by Nguyen and Zakynthinou [23] 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. [28]. 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, 28, 21, 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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

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]	$\varepsilon^{-5}\log\left(\frac{1}{\varepsilon}\right)\log(\frac{n}{\delta})(\log\lvert\mathcal{H}\rvert + n)$ [23]
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]	$\varepsilon^{-2}\left(n\log\lvert\mathcal{H}\rvert + n\log(\frac{n}{\delta})\right)$ [21]
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) [28]

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 OPT $+\,\varepsilon$ or a deterministic classifier of worst-case error 2OPT $+\,\varepsilon$. For the GDRO setting, sample complexity refers to learning a deterministic model with worst-case error of 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.

data may be scarce to start with (as in fairness mechanisms in which samples are amended [25]); when the designer can actively perturb datasets towards rare or atypical instances (such as in robustness applications [16, 35]); or when sample sets represent agents' contributions to an interactive multi-agent system [21, 5].

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 $\widetilde{O}\left(\log(n)\right)$ times the sample complexity of learning a single realizable distribution [4, 6, 23], whereas relying on batched uniform convergence takes $\widetilde{\Omega}\left(n\right)$ 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 [23]. On the other hand, agnostic federated learning bounds [21] have been studied only on algorithms that sample in one large batch and thus require $\widetilde{\Omega}\left(n\right)$ 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 [28], 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., 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 OPT $+\,\varepsilon$ and 2OPT $+\,\varepsilon$, respectively. Our algorithms have an optimal sample complexity of $O\left(\frac{1}{\varepsilon^2}(\log(|H|) + n\log(\frac{n}{\delta}))\right)$. This improves upon the work of Nguyen and Zakynthinou [23] in two ways. First, it provides error bounds of OPT $+\,\varepsilon$ for randomized classifiers, where only 2OPT $+\,\varepsilon$ was previously established. Second, it improves the upper bound of Nguyen and Zakynthinou [23] 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. [28]:* For group DRO, we consider a convex and compact model space $\mathcal{H}$. Our Theorem 5.1 studies a model that achieves

an 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. [28] by a factor of n.

- *Agnostic federated learning of [21]:* 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. [21]. Our results also imply matching high-probability bounds to Mohri et al. [21] 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, 23, 6]. *Agnostic federated learning* interprets these distributions as *clients* in a federated learning system [21]. *Group distributionally robust optimization* interprets these distributions as data attributes or sources that a learner should avoid linking spuriously to labels [14, 28, 29]. 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, 19].

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 [27, 12, 24, 7, 8]. One such method, online mirror descent (OMD), can also approximate minima of convex functions with high probability using noisy first-order information [26, 22, 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 saddle points in robust optimization formulations. The primary difference between multi-distribution learning and traditional saddle-point optimization problems is that oracles give noisy re-usable datapoints instead of noisy local gradients.

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:

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

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 $\mathrm{loss}_q(p) \coloneqq \mathbb{E}_{D\sim q}\left[\mathbb{E}_{h\sim p}\left[\mathrm{loss}_D(h)\right]\right]$.

Collaborative Learning. We will use the *collaborative PAC learning model* of Blum et al. [4] and its agnostic extensions by Nguyen and Zakynthinou [23]. The overall goal of this setting is to guarantee small loss for *every* distribution in a collection of distributions. Formally, we consider a set of data distributions $\mathcal{D} := \{D_1, \ldots, D_n\}$. The goal of the learner is to learn a hypothesis h such that, with probability $1 - \delta$,

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

Group Distribution Robustness. We will also study the closely related setting of *group distributionally robust optimization (Group DRO)* of Sagawa et al. [28]. 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 $\mathcal{D} := \{D_1, \ldots, D_n\}$, the learner seeks a model $\theta \in \Theta$, such that, with probability $1 - \delta$,

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

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 [21], 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., $\text{Reg-Min}(p, q) := \phi(p, q) - \min_{i^* \in A_-} \phi(i^*, q) \leq \varepsilon$ and $\text{Reg-Max}(p, q) := \max_{j^* \in A_+} \phi(p, j^*) - \phi(p, q) \leq \varepsilon$. 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 $g^{(t)}, g^{(t+1)}, \ldots$ 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 $w^{(1)}, \ldots, w^{(t)} \in Z$ and operator sequence $g^{(1)}, \ldots, g^{(t)} \in \mathcal{E}^*$, the algorithm returns $w^{(t+1)} = \mathcal{Q}\left(\left\{w^{(1)}, g^{(1)}\right\}, \ldots, \left\{w^{(t)}, g^{(t)}\right\}\right)$. 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):

$$\mathcal{Q}_{\text{hedge}}\left(\left\{w^{(1)}, g^{(1)}\right\}, \ldots, \left\{w^{(t)}, g^{(t)}\right\}\right) := \frac{\widetilde{w}}{\|\widetilde{w}\|_1} \text{ where } \widetilde{w}_i := w_i^{(t)} \exp\left\{-\eta g_i^{(t)}\right\}, \widetilde{w} \in \mathbb{R}^n \quad (4)$$

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 ([31]). *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 outputs $w^{(1)}, \ldots, w^{(T)}$ satisfies,*

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

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 $A_- = \mathcal{H}, A_+ = \mathcal{D}, \phi(i, j) = \text{loss}_j(i)$, 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.,

$$\mathop{\mathbb{E}}_{h \sim p}\left[\max_{D \in \mathcal{D}} \text{loss}_D(h)\right] \leq \text{OPT} + 2\varepsilon. \quad (5)$$

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 Variational Error 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

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

ensures that variational error provides an upper bound on regret: $\text{Err}_{\text{V}}(w^{(1:T)}) \geq \text{Reg-Min}(p, q) + \text{Reg-Max}(p, q)$, 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 the losses of each hypothesis or distribution under $q^{(t)}$ or $p^{(t)}$ respectively:

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

In the collaborative learning setting, we can only create noisy estimates $\widehat{g}$ for these gradients from samples. No-regret algorithms are advantageous in this setting as they choose their tth iterate $w^{(t)}$ before seeing the tth gradient $g^{(t)}$. This means that $w^{(t)}$ is independent of gradient noise, $\varepsilon^{(t)} := g^{(t)} - \widehat{g}^{(t)}$. We can thus linearize the noise and decompose variational error into the *training* and *generalization* errors 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 [23], 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 loss 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 loss 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. Lemma 3.1 proves this decoupling allows us to find an ε-min-max equilibrium with an additive $n + \log(\mathcal{H})$ sample complexity instead of a 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 step t. The total randomness cost of this algorithm is thus $2T$ because iteration of the outer loop incurs $2r$ cost.

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 $2T$.

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

Output: Mixed strategy profile $(p, q) \in \Delta A_- \times \Delta A_+$;
Input: Action sets A_-, A_+, cost $r \in \mathbb{Z}_+$, timesteps T, iterates $p^{(1)}, q^{(1)}$, gradient estimators $\widehat{g}_-, \widehat{g}_+$;
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: $\widehat{g}_+^{(t)} = \widehat{g}_+\left(\xi^{\perp^{(a)}}, p^{(t)}, q^{(t)}\right), \quad \widehat{g}_-^{(t)} = \widehat{g}_-\left(\xi^{q^{(t)}}, p^{(t)}, q^{(t)}\right)$;
    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 $\overline{q} = \frac{1}{T}\sum_{t=1}^{T} q^{(t)}$;

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 $\mathcal{H}$ 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 in $\mathcal{H}$. As stated below, the error of this deterministic classifier is approximately bounded by the expected error of $\overline{p}$.

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

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

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

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. We describe Algorithm 2, which attains near-optimal sample complexity by on-demand sampling: iteratively selecting distributions to sample from.

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 $\phi(h, D) = \text{loss}_D(h)$. Here, $\xi^{q^{(t)}}$ makes one call to $\text{EX}(q^{(t)})$ and $\xi^{\perp(a)}$ makes one call to $\text{EX}(D)$ 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 $t \in [(a-1)n + 1, \ldots, an]$.

Algorithm 2 On-Demand Agnostic Collaborative Learning.

Input: Hypothesis class $\mathcal{H}$, distribution set $\mathcal{D}$ with $n := |\mathcal{D}|$;
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 $a = 1, 2, \ldots, \lceil T/n \rceil$ **do**
  For all $D \in \mathcal{D}$, sample datapoint z_D^a from EX(D) .
  for $t = an + 1 - n, \ldots, an$ **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 $\frac{1}{T}\sum_{t=1}^{T} p^{(t)}$.

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*

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

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 OPT $+ \varepsilon$. By Lemma 3.2, the corresponding deterministic classifier $h_{\overline{p}}^{Maj}$ has worst-case error of 2OPT $+ \varepsilon$. 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 C.2, 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 $P_1, \ldots, P_n$ such that for all $x \in \mathcal{X}$, $d_i(A) \leq p_i(A)\text{poly}(1/\varepsilon, \text{VCD}(\mathcal{H}), n)$ 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 C.2 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,*

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

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 [23] in two ways. First, it provides OPT $+ \varepsilon$ for randomized classifiers whereas Nguyen and Zakynthinou [23] 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 $\mathcal{O}\left(\log(|\mathcal{H}|) + n\log n\right)$ appears to violate the lower bound of $\Omega\left(\log(|\mathcal{H}|)\log n + n\log\log|\mathcal{H}|\right)$ due to Chen, Zhang, and Zhou [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 $\mathcal{X} = \{1, \ldots, d\}$, $\mathcal{Y} = \{+, -\}$, 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 $n = d \cdot \eta$ for some $\eta \in \mathbb{Z}$. First, we use a hard construction for a d-armed multi-arm bandit exploration problem. This give us the $\Omega\left(d\log(1/\delta)/\varepsilon^2\right)$ part of the lower bound. Second, we construct η hard instances on η corresponding points. Since the learning algorithms has to solve each problem it has to incur a loss of $\eta \cdot d\log(d/\delta)/\varepsilon^2$. □

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 $\max_{D \in \mathcal{D}} \mathbb{E}_{z \sim D}\left[\ell(\theta, z)\right] \leq$ R-OPT $+ \varepsilon$, using a number of samples that is $\mathcal{O}\left(\frac{D_\Theta C^2 + nC^2\log(n/\delta)}{\varepsilon^2}\right)$.*

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 [28] and improves, by a factor of n, existing sample complexity bounds for agnostic federated learning [21]. This improvement is attained by sampling data on-demand, whereas [21] 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. [28]. 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. [28] 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 $\mathcal{O}\left(\frac{D_\Theta C^2 + nC^2\log(n/\delta)}{\varepsilon^2}\right)$.*

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] R. M. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 881–890, USA, Jan. 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-89871-624-5.

[18] 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.

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

[20] 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`.

[21] 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.

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

[23] 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.

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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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`.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]