Agnostic Federated Learning

Mehryar Mohri [1 2] **Gary Sivek** [1] **Ananda Theertha Suresh** [1]

Abstract

A key learning scenario in large-scale applications is that of *federated learning*, where a centralized model is trained based on data originating from a large number of clients. We argue that, with the existing training and inference, federated models can be biased towards different clients. Instead, we propose a new framework of *agnostic federated learning*, where the centralized model is optimized for any target distribution formed by a mixture of the client distributions. We further show that this framework naturally yields a notion of fairness. We present data-dependent Rademacher complexity guarantees for learning with this objective, which guide the definition of an algorithm for agnostic federated learning. We also give a fast stochastic optimization algorithm for solving the corresponding optimization problem, for which we prove convergence bounds, assuming a convex loss function and a convex hypothesis set. We further empirically demonstrate the benefits of our approach in several datasets. Beyond federated learning, our framework and algorithm can be of interest to other learning scenarios such as cloud computing, domain adaptation, drifting, and other contexts where the training and test distributions do not coincide.

1. Motivation

A key learning scenario in large-scale applications is that of *federated learning*. In that scenario, a centralized model is trained based on data originating from a large number of clients, which may be mobile phones, other mobile devices, or sensors (Konečnỳ, McMahan, Yu, Richtárik, Suresh, and Bacon, 2016b; Konečnỳ, McMahan, Ramage, and Richtárik, 2016a). The training data typically remains distributed over the clients, each with possibly unreliable or relatively slow network connections.

Federated learning raises several types of issues and has been the topic of multiple research efforts. These include systems, networking and communication bottleneck problems due to frequent exchanges between the central server and the clients (McMahan et al., 2017). Other research efforts include the design of more efficient communication strategies (Konečnỳ, McMahan, Yu, Richtárik, Suresh, and Bacon, 2016b; Konečnỳ, McMahan, Ramage, and Richtárik, 2016a; Suresh, Yu, Kumar, and McMahan, 2017), devising efficient distributed optimization methods benefiting from differential privacy guarantees (Agarwal, Suresh, Yu, Kumar, and McMahan, 2018), as well as recent lower bound guarantees for parallel stochastic optimization with a dependency graph (Woodworth, Wang, Smith, McMahan, and Srebro, 2018).

Another important problem in federated learning, which appears more generally in distributed machine learning and other learning setups, is that of *fairness*. In many instances in practice, the resulting learning models may be biased or unfair: they may discriminate against some protected groups (Bickel, Hammel, and O'Connell, 1975; Hardt, Price, Srebro, et al., 2016). As a simple example, a regression algorithm predicting a person's salary could be using that person's gender. This is a central problem in modern machine learning that does not seem to have been specifically studied in the context of federated learning.

While many problems related to federated learning have been extensively studied, the key objective of learning in that context seems not to have been carefully examined. We are also not aware of statistical guarantees derived for learning in this scenario. A crucial reason for such questions to emerge in this context is that the target distribution for which the centralized model is learned is unspecified. Which expected loss is federated learning seeking to minimize? Most centralized models for standard federated learning are trained on the aggregate training sample obtained from the subsamples drawn from the clients. Thus, if we denote by $\mathcal{D}_k$ the distribution associated to client k, m_k the size of the sample available from that client and m the total sample size, intrinsically, the centralized model is trained to minimize the loss with respect to the *uniform distribution*

[1]Google Research, New York; [2]Courant Institute of Mathematical Sciences, New York, NY. Correspondence to: Ananda Theertha Suresh <theertha@google.com>.

Proceedings of the 36^{th} *International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

$\overline{\mathcal{U}} = \sum_{k=1}^{p} \frac{m_k}{m} \mathcal{D}_k$. But why should $\overline{\mathcal{U}}$ be the target distribution of the learning model? Is $\overline{\mathcal{U}}$ the distribution that we expect to observe at test time? What guarantees can be derived for the deployed system?

We argue that, in many common instances, the uniform distribution is not the natural objective distribution and that seeking to minimize the expected loss with respect to the specific distribution $\overline{\mathcal{U}}$ is *risky*. This is because the target distribution may be in general quite different from $\overline{\mathcal{U}}$. In many cases, that can result in a suboptimal or even a detrimental performance. For example, imagine a plausible scenario of federated learning where the learner has access to a large population of expensive mobile phones, which are most commonly adopted by software engineers or other technical users (say 70%) than other users (30%), and a small population of other mobile phones less used by non-technical users (5%) and significantly more often by other users (95%). The centralized model would then be essentially based on the uniform distribution based on the expensive clients. But, clearly, such a model would not be adapted to the wide general target domain formed by the majority of phones with a 5%–95% population of general versus technical users. Many other realistic examples of this type can help illustrate the learning problem resulting from a mismatch between the target distribution and $\overline{\mathcal{U}}$. In fact, it is not clear why minimizing the expected loss with respect to $\overline{\mathcal{U}}$ could be beneficial for the clients, whose distributions are $\mathcal{D}_k$s.

Thus, we put forward a new framework of *agnostic federated learning* (AFL), where the centralized model is optimized for any possible target distribution formed by a mixture of the client distributions. Instead of optimizing the centralized model for a specific distribution, with the high risk of a mismatch with the target, we define an agnostic and more risk-averse objective. We show that, for some target mixture distributions, the cross-entropy loss of the hypothesis obtained by minimization with respect to the uniform distribution $\overline{\mathcal{U}}$ can be worse than that of the hypothesis obtained in AFL by a constant additive term, even if the learner has access to infinite samples (Section 2.2).

We further show that our AFL framework naturally yields a notion of fairness, which we refer to as *good-intent fairness* (Section 2.3). Indeed, the predictor solution of the optimization problem for our AFL framework treats all protected categories similarly. Beyond federated learning, our framework and solution also cover related problems in cloud-based learning services, where customers may not have any training data at their disposal or may not be willing to share that data with the cloud due to privacy concerns. In that case too, the server needs to train a model without access to the training data. Our framework and algorithm can also be of interest to other learning scenarios such as domain adaptation, drifting, and other contexts where the training and test distributions do not coincide. In Appendix A, we give an extensive discussion of related work, including connections with the broad literature of domain adaptation.

The rest of the paper is organized as follows. In Section 2, we give a formal description of AFL. Next, we give a detailed theoretical analysis of learning within the AFL framework (Section 3), as well as a learning algorithm based on that theory (Section 4). We also present an efficient convex optimization algorithm for solving the optimization problem defining our algorithm (Section 4.2). In Section 5, we present a series of experiments comparing our solution with existing federated learning solutions. In Appendix B, we discuss several extensions of the AFL framework.

2. Learning scenario

In this section, we introduce the learning scenario of agnostic federated learning we consider. We then argue that the uniform solution commonly adopted in standard federated learning may not be an adequate solution, thereby further justifying our agnostic model. Next, we show the benefit of our model in fairness learning.

We start with some general notation and definitions used throughout the paper. Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space. We will primarily discuss a multi-class classification problem where $\mathcal{Y}$ is a finite set of classes, but much of our results can be extended straightforwardly to regression and other problems. The hypotheses we consider are of the form $h\colon \mathcal{X} \to \Delta_{\mathcal{Y}}$, where $\Delta_{\mathcal{Y}}$ stands for the simplex over $\mathcal{Y}$. Thus, $h(x)$ is a probability distribution over the classes or categories that can be assigned to $x \in \mathcal{X}$. We will denote by $\mathcal{H}$ a family of such hypotheses h. We also denote by ℓ a loss function defined over $\Delta_{\mathcal{Y}} \times \mathcal{Y}$ and taking non-negative values. The loss of $h \in \mathcal{H}$ for a labeled sample $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is given by $\ell(h(x), y)$. One key example in applications is the cross-entropy loss, which is defined as follows: $\ell(h(x), y) = -\log(\mathbb{P}_{y' \sim h(x)}[y' = y])$. We will denote by $\mathcal{L}_{\mathcal{D}}(h)$ the expected loss of a hypothesis h with respect to a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)],$$

and by $h_{\mathcal{D}}$ its minimizer: $h_{\mathcal{D}} = \operatorname{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h)$.

2.1. Agnostic federated learning

We consider a learning scenario where the learner receives p samples $S_1, \ldots, S_p$, with each $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,m_k}, y_{k,m_k})) \in (\mathcal{X} \times \mathcal{Y})^{m_k}$ of size m_k drawn i.i.d. from a possibly different domain or distribution $\mathcal{D}_k$. We will denote by $\widehat{\mathcal{D}}_k$ the empirical distribution associated to sample S_k of size m drawn from $\mathcal{D}^m$. The learner's objective is to determine a hypothesis $h \in \mathcal{H}$ that performs well on some target distribution. Let $m = \sum_{k=1}^{p} m_k$.

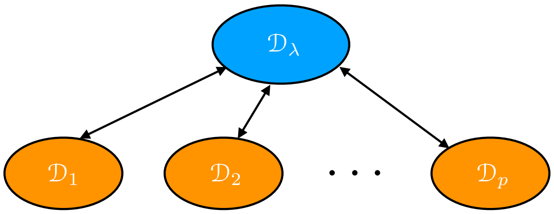

Figure 1. Illustration of the agnostic federated learning scenario.

This scenario coincides with that of *federated learning* where training is done with the *uniform distribution* over the union of all samples S_k, where all samples are uniformly weighted, that is $\widehat{\mathcal{U}} = \sum_{k=1}^{p} \frac{m_k}{m} \widehat{\mathcal{D}}_k$, and where the underlying assumption is that the target distribution is $\mathcal{U} = \sum_{k=1}^{p} \frac{m_k}{m} \mathcal{D}_k$. We will not adopt that assumption since it is rather restrictive and since, as discussed later, it can lead to solutions that are detrimental to domain users. Instead, we will consider an *agnostic federated learning* (AFL) scenario where the target distribution can be modeled as an unknown mixture of the distributions $\mathcal{D}_k$, $k = 1, \ldots, p$, that is $\mathcal{D}_\lambda = \sum_{k=1}^{p} \lambda_k \mathcal{D}_k$ for some $\lambda \in \Delta_p$. Since the mixture weight λ is unknown, here, the learner must come up with a solution that is favorable for any λ in the simplex, or any λ in a subset $\Lambda \subseteq \Delta_p$. Thus, we define the *agnostic loss* (or *agnostic risk*) $\mathcal{L}_{\mathcal{D}_\Lambda}(h)$ associated to a predictor $h \in \mathcal{H}$ as

$$\mathcal{L}_{\mathcal{D}_\Lambda}(h) = \max_{\lambda \in \Lambda} \mathcal{L}_{\mathcal{D}_\lambda}(h). \tag{1}$$

We will extend our previous definitions and denote by $h_{\mathcal{D}_\Lambda}$ the minimizer of this loss: $h_{\mathcal{D}_\Lambda} = \operatorname{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}_\Lambda}(h)$.

In practice, the learner has access to the distributions $\mathcal{D}_k$ only via the finite samples S_k. Thus, for any $\lambda \in \Delta_p$, instead of the mixture $\mathcal{D}_\lambda$, only the λ-mixture of empirical distributions, $\overline{\mathcal{D}}_\lambda = \sum_{k=1}^{p} \lambda_k \widehat{\mathcal{D}}_k$, is accessible.[1] This leads to the definition of $\mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h)$, the *agnostic empirical loss* of a hypothesis $h \in \mathcal{H}$ for a subset of the simplex, Λ:

$$\mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h) = \max_{\lambda \in \Lambda} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h).$$

We will denote by $h_{\overline{\mathcal{D}}_\Lambda}$ the minimizer of this loss: $h_{\overline{\mathcal{D}}_\Lambda} = \operatorname{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h)$. In the next section, we will present generalization bounds relating the expected and empirical agnostic losses $\mathcal{L}_{\mathcal{D}_\Lambda}(h)$ and $\mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h)$ for all $h \in \mathcal{H}$.

Notice that the domains $\mathcal{D}_k$ discussed thus far need not coincide with the clients. In fact, when the number of clients is very large and Λ is the full simplex, $\Lambda = \Delta_p$, it is typically preferable to consider instead domains defined by clusters of clients, as discussed in Appendix B. On the other hand, if p is small or Λ more restrictive, then the model may not perform well on certain domains of interest. We mitigate the effect of large p values using a suitable regularization term derived from our theory.

2.2. Comparison with federated learning

Here, we further argue that the uniform solution $h_{\overline{\mathcal{U}}}$ commonly adopted in federated learning may not provide a satisfactory performance compared with a solution of the agnostic problem. This further motivates our AFL model.

As already discussed, since the target distribution is unknown, the natural method for the learner is to select a hypothesis minimizing the agnostic loss $\mathcal{L}_{\mathcal{D}_\Lambda}$. Is the predictor minimizing the agnostic loss coinciding with the solution $h_{\widehat{\mathcal{U}}}$ of standard federated learning? How poor can the performance of the standard federated learning be? We first show that the loss of $h_{\widehat{\mathcal{U}}}$ can be higher than that of the optimal loss achieved by $h_{\mathcal{D}_\Lambda}$ by a constant loss, even if the number of samples tends to infinity, that is even if the learner has access to the distributions $\mathcal{D}_k$ and uses the predictor $h_{\overline{\mathcal{U}}}$.

Proposition 1. *[Appendix C.1] Let ℓ be the cross-entropy loss. Then, there exist Λ, $\mathcal{H}$, and $\mathcal{D}_k$, $k \in [p]$, such that the following inequality holds:*

$$\mathcal{L}_{\mathcal{D}_\Lambda}(h_{\overline{\mathcal{U}}}) \geq \mathcal{L}_{\mathcal{D}_\Lambda}(h_{\mathcal{D}_\Lambda}) + \log \frac{2}{\sqrt{3}}.$$

2.3. Good-intent fairness in learning

Fairness in machine learning has received much attention in recent past (Bickel et al., 1975; Hardt et al., 2016). There is now a broad literature on the topic with a variety of definitions of the notion of fairness. In a typical scenario, there is a protected class c among p classes $c_1, c_2, \ldots, c_p$. While there are many definitions of fairness, the main objective of a fairness algorithm is to reduce bias and ensure that the model is fair to all the p protected categories, under some definition of fairness. The most common reasons for bias in machine learning algorithms are training data bias and overfitting bias. We first provide a brief explanation and illustration for both:

- biased training data: consider the regression task, where the goal is to predict the salary of a person based on features such as education, location, age, gender. Let gender be the protected class. If in the training data, there is a consistent discrimination against women irrespective of their education, e.g., their salary is lower, then we can conclude that the training data is inherently biased.
- biased training procedure: consider an image recognition task where the protected category is race. If the model is heavily trained on images based on certain races, then the resulting model will be biased because of over-fitting.

[1] Note, $\overline{\mathcal{D}}_\lambda$ is distinct from an empirical distribution $\widehat{\mathcal{D}}_\lambda$ which would be based on a sample drawn from $\mathcal{D}_\lambda$. $\overline{\mathcal{D}}_\lambda$ is based on samples drawn from $\mathcal{D}_k$s.

Our model of AFL can help define a notion of good-intent fairness, where we reduce the bias in the training procedure. Furthermore, if training procedure bias exists, it naturally highlights it.

Suppose we are interested in a classification problem and there is a protected feature class c, which can be one of p values $c_1, c_2, \ldots, c_p$. Then, we define $\mathcal{D}_k$ as the conditional distribution with the protected class being c_k. If $\mathcal{D}$ is the true underlying distribution, then

$$\mathcal{D}_k(x, y) = \mathcal{D}(x, y \mid c(x, y) = c_k).$$

Let $\Lambda = \{\delta_k \colon k \in [p]\}$ be the collection of Dirac measures over the indices k in $[p]$. With these definitions, a natural fairness principle consists of ensuring that the test loss is the same for all underlying protected classes, that is for all $\lambda \in \Lambda$. This is called the maxmin principle (Rawls, 2009), a special case of the CVar fairness risk (Williamson & Menon, 2019).

With the above intent in mind, we define a *good-intent fairness* algorithm as one seeking to minimize the agnostic loss $\mathcal{L}_{\mathcal{D}_\Lambda}$. Thus, the objective of the algorithm is to minimize the maximum loss incurred on any of the underlying protected classes and hence does not overfit the data to any particular model at the cost of others. Furthermore, it does not degrade the performance of the other classes so long as it does not affect the loss of the most-sensitive protected category. We further note that our approach does not reduce bias in the training data and is useful only for mitigating the training procedure bias.

3. Learning bounds

We now present learning guarantees for agnostic federated learning. Let $\mathcal{G}$ denote the family of the losses associated to a hypothesis set $\mathcal{H}$: $\mathcal{G} = \{(x, y) \mapsto \ell(h(x), y) \colon h \in \mathcal{H}\}$. Our learning bounds are based on the following notion of *weighted Rademacher complexity* which is defined for any hypothesis set $\mathcal{H}$, vector of sample sizes $\mathbf{m} = (m_1, \ldots, m_p)$ and mixture weight $\lambda \in \Delta_p$, by the following expression:

$$\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) = \mathop{\mathbb{E}}_{\substack{S_k \sim \mathcal{D}_k^{m_k} \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \sigma_{k,i}\, \ell(h(x_{k,i}), y_{k,i}) \right], \tag{2}$$

where $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,m_k}, y_{k,m_k}))$ is a sample of size m_k and $\boldsymbol{\sigma} = (\sigma_{k,i})_{k \in [p], i \in [m_k]}$ a collection of Rademacher variables, that is uniformly distributed random variables taking values in $\{-1, +1\}$. We also define the *minimax weighted Rademacher complexity* for a subset $\Lambda \subseteq \Delta_p$ by

$$\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \Lambda) = \max_{\lambda \in \Lambda} \mathfrak{R}_m(\mathcal{G}, \lambda). \tag{3}$$

Let $\overline{\mathbf{m}} = \frac{\mathbf{m}}{m} = \left(\frac{m_1}{m}, \ldots, \frac{m_p}{m}\right)$ denote the empirical distribution over Δ_p defined by the sample sizes m_k, where $m = \sum_{k=1}^{p} m_k$. We define the *skewness* of Λ with respect to $\overline{\mathbf{m}}$ by

$$\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}}) = \max_{\lambda \in \Lambda} \chi^2(\lambda \,\|\, \overline{\mathbf{m}}) + 1, \tag{4}$$

where, for any two distributions p and q in Δ_p, the chi-squared divergence $\chi^2(p \,\|\, q)$ is given by $\chi^2(p \,\|\, q) = \sum_{k=1}^{p} \frac{(p_k - q_k)^2}{q_k}$. We will also denote by Λ_ϵ a minimum ϵ-cover of Λ in ℓ_1 distance, that is, $\Lambda_\epsilon = \operatorname{argmin}_{\Lambda' \in C(\Lambda, \epsilon)} |\Lambda|$, where $C(\Lambda, \epsilon)$ is a set of distributions Λ' such that for every $\lambda \in \Lambda$, there exists $\lambda' \in \Lambda'$ such that $\sum_{k=1}^{p} |\lambda_k - \lambda'_k| \leq \epsilon$.

Our first learning guarantee is presented in terms of $\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \Lambda)$, the skewness parameter $\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}})$ and the ϵ-cover Λ_ϵ.

Theorem 1. *[Appendix C.2] Assume that the loss ℓ is bounded by $M > 0$. Fix $\epsilon > 0$ and $\mathbf{m} = (m_1, \ldots, m_p)$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of samples $S_k \sim \mathcal{D}_k^{m_k}$, for all $h \in \mathcal{H}$ and $\lambda \in \Lambda$, $\mathcal{L}_{\mathcal{D}_\lambda}(h)$ is upper bounded by*

$$\mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + 2\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) + M\epsilon + M\sqrt{\frac{\mathfrak{s}(\lambda \,\|\, \overline{\mathbf{m}})}{2m} \log \frac{|\Lambda_\epsilon|}{\delta}},$$

where $m = \sum_{k=1}^{p} m_k$.

It can be shown that for a given λ, the variance of the loss depends on the skewness parameter and hence it can be shown that generalization bound can also be lower bounded in terms of the skewness parameter (Theorem 9 in Cortes et al. (2010)). Note that the bound in Theorem 1 is *instance-specific*, i.e., it depends on the target distribution $\mathcal{D}_\lambda$ and increases monotonically as λ moves away from $\overline{\mathbf{m}}$. Thus, for target domains with $\lambda \approx \overline{\mathbf{m}}$, the bound is more favorable. The theorem supplies upper bounds for agnostic losses: they can be obtained simply by taking the maximum over $\lambda \in \Lambda$. The following result shows that, for a family of functions taking values in $\{-1, +1\}$, the Rademacher complexity $\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \Lambda)$ can be bounded in terms of the VC-dimension and the skewness of Λ.

Lemma 1. *[Appendix C.3] Let ℓ be a loss function taking values in $\{-1, +1\}$ and such that the family of losses $\mathcal{G}$ admits VC-dimension d. Then, the following upper bound holds for the weighted Rademacher complexity of $\mathcal{G}$:*

$$\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \Lambda) \leq \sqrt{2\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}}) \frac{d}{m} \log\left[\frac{em}{d}\right]}.$$

Both Lemma 1 and the generalization bound of Theorem 1 can thus be expressed in terms of the skewness parameter $\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}})$. Note that, when Λ contains only one distribution and is the uniform distribution, that is $\lambda_k = m_k/m$, then the skewness is equal to one and the results coincide with the standard guarantees in supervised learning.

Theorem 1 and Lemma 1 also provide guidelines for choosing the domains and Λ. When p is large and $\Lambda = \Delta_p$, then, the number of samples per domain could be small, the skewness parameter $\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}}) = \max_{1 \le k \le p} \frac{1}{m_k}$ would then be large and the generalization guarantees for the model would become weaker. We suggest some guidelines for choosing domains in Appendix B. We further note that, for a given p, if Λ contains distributions that are close to $\overline{\mathbf{m}}$, then the model generalizes well.

The corollary above can be straightforwardly extended to cover the case where the test samples are drawn from some distribution $\mathcal{D}$, instead of $\mathcal{D}_\lambda$. Define $\ell_1(\mathcal{D}, \mathcal{D}_\Lambda)$ by $\ell_1(\mathcal{D}, \mathcal{D}_\Lambda) = \min_{\lambda \in \Lambda} \ell_1(\mathcal{D}, \mathcal{D}_\lambda)$. Then, the following result holds.

Corollary 1. *Assume that the loss function ℓ is bounded by M. Then, for any $\epsilon > 0$ and $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds for all $h \in \mathcal{H}$:*

$$\mathcal{L}_{\mathcal{D}}(h) \le \mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h) + 2\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \Lambda) + M\ell_1(\mathcal{D}, \mathcal{D}_\Lambda) + M\epsilon + M\sqrt{\frac{\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}})}{2m} \log \frac{|\Lambda_\epsilon|}{\delta}}.$$

One straightforward choice of the parameter ϵ is $\epsilon = \frac{1}{\sqrt{m}}$, but, depending on $|\Lambda_\epsilon|$ and other parameters of the bound, more favorable choices may be possible. We conclude this section by adding that alternative learning bounds can be derived for this problem, as discussed in Appendix D.

4. Algorithm

4.1. Regularization

The learning guarantees of the previous section suggest minimizing the sum of the empirical AFL term $\mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h)$, a term controlling the complexity of $\mathcal{H}$ and a term depending on the skewness parameter. Observe that, since $\mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)$ is linear in λ, the following equality holds:

$$\mathcal{L}_{\overline{\mathcal{D}}_\Lambda}(h) = \mathcal{L}_{\overline{\mathcal{D}}_{\mathrm{conv}(\Lambda)}}(h), \tag{5}$$

where $\mathrm{conv}(\Lambda)$ is the convex hull of Λ. Assume that $\mathcal{H}$ is a vector space that can be equipped with a norm $\|\cdot\|$, as with most hypothesis sets used in learning applications. Then, given Λ and the regularization parameters $\mu \ge 0$ and $\gamma \ge 0$, our learning guarantees suggest the following minimization problem:

$$\min_{h \in \mathcal{H}} \max_{\lambda \in \mathrm{conv}(\Lambda)} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \gamma \|h\| + \mu\, \chi^2(\lambda \,\|\, \overline{\mathbf{m}}). \tag{6}$$

This defines our algorithm for AFL.

Assume that ℓ is a convex function of its first argument. Then, $\mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)$ is a convex function of h. Since $\|h\|$ is a convex function of h for any choice of the norm, for a fixed λ, the objective $\mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \gamma\|h\| + \mu\,\chi^2(\lambda \,\|\, \overline{\mathbf{m}})$ is a convex function of h. The maximum over λ (taken in any set) of a family of convex functions is convex. Thus, $\max_{\lambda \in \mathrm{conv}(\Lambda)} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \gamma\|h\| + \mu\,\chi^2(\lambda \,\|\, \overline{\mathbf{m}})$ is a convex function of h and, when the hypothesis set $\mathcal{H}$ is a convex, (6) is a convex optimization problem. In the next subsection, we present an efficient optimization solution for this problem in Euclidean norm, for which we prove convergence guarantees. In Appendix F.1, we generalize the results to other norms.

4.2. Optimization algorithm

When the loss function ℓ is convex, the AFL minmax optimization problem above can be solved using projected gradient descent or other instances of the generic mirror descent algorithm (Nemirovski & Yudin, 1983). However, for large datasets, that is p and m large, this can be computationally costly and typically slow in practice. Juditsky, Nemirovski, and Tauvel (2011) proposed a stochastic Mirror-Prox algorithm for solving stochastic variational inequalities, which would be applicable in our context. We present a simplified version of their algorithm for the AFL problem that admits a more straightforward analysis and that is also substantially easier to implement.

Our optimization problem is over two sets of parameters, the hypothesis $h \in \mathcal{H}$ and the mixture weight $\lambda \in \Lambda$. In what follows, we will denote by $\mathcal{W}$ a non-empty subset of $\mathbb{R}^N$ and $w \in \mathcal{W}$ a vector of parameters defining a predictor h. Thus, we will rewrite losses and optimization solutions only in terms of w, instead of h. We will use the following notation:

$$\mathsf{L}(w, \lambda) = \sum_{k=1}^{p} \lambda_k \mathsf{L}_k(w), \tag{7}$$

where $\mathsf{L}_k(w)$ stands for $\mathcal{L}_{\widehat{\mathcal{D}}_k}(h)$, the empirical loss of hypothesis $h \in \mathcal{H}$ (corresponding to w) on domain k: $\mathsf{L}_k(w) = \frac{1}{m_k} \sum_{i=1}^{m_k} \ell(h(x_{k,i}), y_{k,i})$. We will consider the unregularized version of problem (6). We note that regularization with respect to w does not make the optimization harder. Thus, we will study the following problem given by the set of variables w:

$$\min_{w \in \mathcal{W}} \max_{\lambda \in \Lambda} \mathsf{L}(w, \lambda). \tag{8}$$

Observe that problem (8) admits a natural game-theoretic interpretation as a two-player game, where nature selects $\lambda \in \Lambda$ to maximize the objective, while the learner seeks $w \in \mathcal{W}$ minimizing the loss. We are interested in finding the equilibrium of this game, which is attained for some w^*, the minimizer of Equation 8 and $\lambda^* \in \Lambda$, the hardest domain mixture weights. At the equilibrium, moving w away from w^* or λ from λ^*, increases the objective function. Hence, λ^* can be viewed as the center of Λ in the manifold imposed

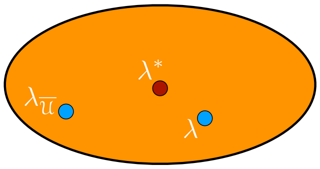

Figure 2. Illustration of the positions in Λ of λ^*, $\lambda_{\overline{\mathcal{U}}}$, the mixture weight corresponding to the distribution $\overline{\mathcal{U}}$, and an arbitrary λ. λ^* defines the least risky distribution $\overline{\mathcal{D}}_{\lambda^*}$ for which to optimize the expected loss.

by the loss function L, whereas $\overline{\mathcal{U}}$, the empirical distribution of samples, may lie elsewhere, as illustrated by Figure 2.

By Equation 5, using the set $\operatorname{conv}(\Lambda)$ instead of Λ does not affect the solution of the optimization problem. In view of that, in what follows, we will assume, without loss of generality, that Λ is a convex set. Observe that, since $\mathsf{L}_k(w)$ is not an average of functions, standard stochastic gradient descent algorithms cannot be used to minimize this objective. We will present instead a new stochastic gradient-type algorithm for this problem.

Let $\nabla_w \mathsf{L}(w, \lambda)$ denote the gradient of the loss function with respect to w and $\nabla_\lambda \mathsf{L}(w, \lambda)$ the gradient with respect to λ. Let $\delta_w \mathsf{L}(w, \lambda)$, and $\delta_\lambda \mathsf{L}(w, \lambda)$ be unbiased estimates of the gradient, that is,

$$\mathop{\mathbb{E}}_{\delta}[\delta_\lambda \mathsf{L}(w, \lambda)] = \nabla_\lambda \mathsf{L}(w, \lambda), \ \mathop{\mathbb{E}}_{\delta}[\delta_w \mathsf{L}(w, \lambda)] = \nabla_w \mathsf{L}(w, \lambda).$$

We first give an optimization algorithm STOCHASTIC-AFL for the AFL problem, assuming access to such unbiased estimates. The pseudocode of the algorithm is given in Figure 3. At each step, the algorithm computes a stochastic gradient with respect to λ and w and updates the model accordingly. It then projects λ to Λ by computing a value in Λ via convex minimization. If Λ is the full simplex, then there exist simple and efficient algorithms for this projection (Duchi et al., 2008). It then repeats the process for T steps and returns the average of the weights.

There are several natural candidates for the sampling method defining stochastic gradients. We highlight two techniques: PERDOMAIN GRADIENT and WEIGHTED GRADIENT. We analyze the time complexity and give bounds on the variance for both techniques in Lemmas 3 and 4 respectively.

4.3. Analysis

Throughout this section, for simplicity, we adopt the notation introduced for Equation 7. Our convergence guarantees hold under the following assumptions, which are similar to those adopted for the convergence proof of gradient descent-type algorithms.

Properties 1. *Assume that the following properties hold for the loss function* L *and sets* $\mathcal{W}$ *and* $\Lambda \subseteq \Delta_p$*:*

```
Algorithm STOCHASTIC-AFL
Initialization: $w_0 \in \mathcal{W}$ and $\lambda_0 \in \Lambda$.
Parameters: step size $\gamma_w > 0$ and $\gamma_\lambda > 0$.
For $t = 1$ to $T$:
    1. Obtain stochastic gradients: $\delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1})$ and $\delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1})$.
    2. $w_t = \text{PROJECT}(w_{t-1} - \gamma_w \delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1}), \mathcal{W})$
    3. $\lambda_t = \text{PROJECT}(\lambda_{t-1} + \gamma_\lambda \delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1}), \Lambda)$.
Output: $w^A = \frac{1}{T}\sum_{t=1}^T w_t$ and $\lambda^A = \frac{1}{T}\sum_{t=1}^T \lambda_t$.
Subroutine PROJECT
Input: $x', \mathcal{X}$. Output: $x = \operatorname{argmin}_{x \in \mathcal{X}} \|x - x'\|_2$.
```

Figure 3. Pseudocode of the STOCHASTIC-AFL algorithm.

1. Convexity*:* $w \mapsto \mathsf{L}(w, \lambda)$ *is convex for any* $\lambda \in \Lambda$.
2. Compactness*:* $\max_{\lambda \in \Lambda} \|\lambda\|_2 \leq R_\Lambda$, $\max_{w \in \mathcal{W}} \|w\|_2 \leq R_{\mathcal{W}}$.
3. Bounded gradients*:* $\|\nabla_w \mathsf{L}(w, \lambda)\|_2 \leq G_w$ *and* $\|\nabla_\lambda \mathsf{L}(w, \lambda)\|_2 \leq G_\lambda$ *for all* $w \in \mathcal{W}$ *and* $\lambda \in \Lambda$.
4. Stochastic variance*:* $\mathbb{E}[\|\delta_w \mathsf{L}(w, \lambda) - \nabla_w \mathsf{L}(w, \lambda)\|_2^2] \leq \sigma_w^2$ *and* $\mathbb{E}[\|\delta_\lambda \mathsf{L}(w, \lambda) - \nabla_\lambda \mathsf{L}(w, \lambda)\|_2^2] \leq \sigma_\lambda^2$ *for all* $w \in \mathcal{W}$ *and* $\lambda \in \lambda$.
5. Time complexity*:* U_w *denotes the time complexity of computing* $\delta_w \mathsf{L}(w, \lambda)$*,* U_λ *that of computing* $\delta_\lambda \mathsf{L}(w, \lambda)$*,* U_p *that of the projection, and* d *denotes the dimensionality of* $\mathcal{W}$.

Theorem 2. *[Appendix E.1] Assume that Properties 1 hold. Then, the following guarantee holds for* STOCHASTIC-AFL*:*

$$\mathbb{E}\left[\max_{\lambda \in \Lambda} \mathsf{L}(w^A, \lambda) - \min_{w \in \mathcal{W}} \max_{\lambda \in \Lambda} \mathsf{L}(w, \lambda)\right] \leq \frac{3R_{\mathcal{W}}\sqrt{(\sigma_w^2 + G_w^2)}}{\sqrt{T}} + \frac{3R_\Lambda\sqrt{(\sigma_\lambda^2 + G_\lambda^2)}}{\sqrt{T}},$$

for the step sizes $\gamma_w = \frac{2R_{\mathcal{W}}}{\sqrt{T(\sigma_w^2 + G_w^2)}}$ *and* $\gamma_\lambda = \frac{2R_\Lambda}{\sqrt{T(\sigma_\lambda^2 + G_\lambda^2)}}$*, and the time complexity of the algorithm is in* $\mathcal{O}((U_\lambda + U_w + U_p + d + p)T)$.

We note that similar algorithms have been proposed for solving minimax objectives (Namkoong & Duchi, 2016; Chen et al., 2017). Chen et al. (2017) assume the existence of an α-approximate Bayesian oracle, whereas our guarantees hold regardless of such assumptions. Namkoong & Duchi (2016) use importance sampling to obtain λ gradients, thus, their convergence guarantee for the Euclidean norm depends inversely on a lower bound on $\min_{\lambda \in \Lambda} \min_{k \in [p]} \lambda_k$.

Stochastic gradient for λ.

1. Sample $K \sim [p]$, according to the uniform distribution.
   Sample $I_K \sim [m_K]$, according to the uniform distribution.
2. **Output:** $\delta_\lambda \mathsf{L}(w,\lambda)$ such that $[\delta_\lambda \mathsf{L}(w,\lambda)]_K = p\mathsf{L}_{K,I_K}(w)$ and for all $k \neq K$, $[\delta_\lambda \mathsf{L}(w,\lambda)]_k = 0$.

PerDomain-stochastic gradient for w.

1. For $k \in [p]$, sample $J_k \sim [m_k]$, according to the uniform distribution.
2. **Output:** $\delta_w \mathsf{L}(w,\lambda) = \sum_{k=1}^p \lambda_k \nabla_w L_{k,J_k}(w,h)$.

Weighted-stochastic gradient for w

1. Sample $K \sim [p]$ according to the distribution λ.
   Sample $J_K \sim [m_k]$, according to the uniform distribution.
2. **Output:** $\delta_w \mathsf{L}(w,\lambda) = \nabla_w \mathsf{L}_{K,J_K}(w)$.

Figure 4. Definition of the stochastic gradients with respect to λ and w.

In contrast, our convergence guarantees are not affected by that.

4.4. Stochastic gradients

The convergence results of Theorem 4 depend on the variance of the stochastic gradients. We first discuss the stochastic gradients for λ. Notice that the gradient for λ is independent of λ. Thus, a natural choice for the stochastic gradient with respect to λ is based on uniformly sampling a domain $K \in [p]$ and then sampling $x_{K,i}$ from domain K. This leads to the definition of the stochastic gradient $\delta_\lambda \mathsf{L}(w,\lambda)$ shown in Figure 4. The following lemma bounds the variance for that definition of $\delta_\lambda \mathsf{L}(w,\lambda)$.

Lemma 2. *[Appendix E.2] The stochastic gradient $\delta_\lambda \mathsf{L}(w,\lambda)$ is unbiased. Further, if the loss function is bounded by M, then the following upper bound holds for the variance of $\delta_\lambda \mathsf{L}(w,\lambda)$:*

$$\sigma_\lambda^2 = \max_{w\in\mathcal{W},\lambda\in\Lambda} \mathrm{Var}(\delta_\lambda \mathsf{L}(w,\lambda)) \leq p^2 M^2.$$

If the above variance is too high, then we can sample one J_k for every domain k. This is the same as computing the gradient of a batch and reduces the variance by a factor of p.

The gradient with respect to w depends both on λ and w. There are two natural stochastic gradients: the PerDomain-stochastic gradient and the Weighted-stochastic gradient. For a PerDomain-stochastic gradient, we sample an element uniformly from $[m_k]$ for each $k \in [p]$. For the Weighted-stochastic gradient, we sample a domain according to λ and sample an element out of it. We can now bound the variance of both PerDomain and Weighted stochastic gradients. Let U denote the time complexity of computing the loss and gradient with respect to w for a single sample.

Lemma 3. *[Appendix E.3]* PerDomain *stochastic gradient is unbiased and runs in time $pU + \mathcal{O}(p\log m)$ and the variance satisfy, $\sigma_w^2 \leq R_\Lambda \sigma_I^2(w)$, where*

$$\sigma_I^2(w) = \max_{w\in\mathcal{W},k\in[p]} \frac{1}{m_k}\sum_{j=1}^{m_k}\left[\nabla_w L_{k,j}(w) - \nabla_w L_k(w)\right]^2.$$

Lemma 4. *[Appendix E.4]* Weighted *stochastic gradient is unbiased and runs in time $U + \mathcal{O}(p + \log m)$ and the variance satisfy the following inequality: $\sigma_w^2 \leq \sigma_I^2(w) + \sigma_O^2(w)$, where*

$$\sigma_O^2(w) = \max_{w\in\mathcal{W},\lambda\in\Lambda}\sum_{k=1}^{p}\lambda_k\left[\nabla_w \mathsf{L}_k(w) - \nabla_w \mathsf{L}(w,\lambda)\right]^2$$

and $\sigma_I^2(w)$ is defined in Lemma 3.

Since $R_\Lambda \leq 1$, at first glance, the above two lemmas may suggest that PerDomain stochastic is always better than Weighted stochastic gradient. Note, however, that the time complexity of the algorithms is dominated by U and thus, the time complexity of PerDomain-stochastic gradient is roughly p times larger than that of Weighted-stochastic gradient. Hence, if p is small, it is preferable to choose the PerDomain-stochastic gradient. For large values of p, we analyze the differences in Appendix E.5.

4.5. Related optimization algorithms

In Appendix F.1, we show that Stochastic-AFL can be extended to the case where arbitrary mirror maps are used, as in the standard mirror descent algorithm. In Appendix F.2, we give an algorithm with convergence rate $\mathcal{O}(\log T/T)$, when the loss function is strongly convex. Finally, in Appendix F.3, we present an optimistic version of Stochastic-AFL.

5. Experiments

To study the benefits of our AFL algorithm, we carried out experiments with three datasets. Even though our optimization convergence guarantees hold only for convex functions and stochastic gradient, we show that our domain-agnostic learning performs well for non-convex functions and variants of stochastic gradient descent such as Adagrad too.

In all the experiments, we compare the domain agnostic model with the model trained with $\widehat{\mathcal{U}}$, the uniform distribution over the union of samples, and the models trained on

Table 1. Adult dataset: test accuracy for various test domains of models trained with different loss functions.

Training loss function	$\mathcal{U}$	`doctorate`	`non-doctorate`	$\mathcal{D}_\Lambda$
$\mathcal{L}_{\texttt{doctorate}}$	53.35 ± 0.91	73.58 ± 0.48	53.12 ± 0.89	53.12 ± 0.89
$\mathcal{L}_{\texttt{non-doctorate}}$	82.15 ± 0.09	69.46 ± 0.29	82.29 ± 0.09	69.46 ± 0.29
$\mathcal{L}_{\widehat{\mathcal{U}}}$	82.10 ± 0.09	69.61 ± 0.35	82.24 ± 0.09	69.61 ± 0.35
$\mathcal{L}_{\mathcal{D}_\Lambda}$	80.10 ± 0.39	71.53 ± 0.88	80.20 ± 0.40	71.53 ± 0.88

Table 2. Fashion MNIST dataset: test accuracy for various test domains of models trained with different loss functions.

Training loss function	$\mathcal{U}$	`shirt`	`pullover`	`t-shirt/top`	$\mathcal{D}_\Lambda$
$\mathcal{L}_{\widehat{\mathcal{U}}}$	81.8 ± 1.3	71.2 ± 7.8	87.8 ± 6.0	86.2 ± 4.9	71.2 ± 7.8
$\mathcal{L}_{\mathcal{D}_\Lambda}$	82.3 ± 0.9	74.5 ± 6.0	87.6 ± 4.5	84.9 ± 4.4	74.5 ± 6.0

individual domains. In all the experiments, we used PERDOMAIN stochastic gradients and set $\Lambda = \Delta_p$. All algorithms were implemented in Tensorflow (Abadi et al., 2015).

5.1. Adult dataset

The Adult dataset is a census dataset from the UCI Machine Learning Repository (Blake, 1998). The task consists of predicting if the person's income exceeds \$50,000. We split this dataset into two domains depending on whether the person had a doctorate degree or not, resulting into domains: the `doctorate` domain and the `non-doctorate` domain. We trained a logistic regression model with just the categorical features and Adagrad optimizer. The performance of the models averaged over 50 runs is reported in Table 1. The performance on $\mathcal{D}_\Lambda$ of the model trained with $\widehat{\mathcal{U}}$, that is standard federated learning, is about 69.6%. In contrast, the performance of our AFL model is at least about 71.5% on *any* target distribution $\mathcal{D}_\lambda$. The uniform average over the domains of the test accuracy of the AFL model is slightly less than that of the uniform model, but the agnostic model is less biased and performs better on $\mathcal{D}_\Lambda$.

5.2. Fashion MNIST

The Fashion MNIST dataset (Xiao et al., 2017) is an MNIST-like dataset where images are classified into 10 categories of clothing, instead of handwritten digits. We extracted the subset of the data labeled with three categories `t-shirt/top`, `pullover`, and `shirt` and split this subset into three domains, each consisting of one class of clothing. We then trained a classifier for the three classes using logistic regression and the Adam optimizer. The results are shown in Table 2. Since here the domain uniquely identifies the label, in this experiment, we did not compare against models trained on specific domains. Of the three domains or classes, the `shirt` class is the hardest one to distinguish from others. The domain-agnostic model improves the performance for `shirt` more than it degrades it on `pullover` and `shirt`, leading to both `shirt`-specific and overall accuracy improvement when compared to the model trained with the uniform distribution $\widehat{\mathcal{U}}$. Furthermore, in this experiment, note that our agnostic learning solution not only improves the loss of the worst domain, but also generalizes better and hence improves the average test accuracy.

Table 3. Test perplexity for various test domains of models trained with different loss functions.

Training loss func.	$\mathcal{U}$	`doc.`	`con.`	$\mathcal{D}_\Lambda$
$\mathcal{L}_{\texttt{doc.}}$	414.96	83.97	615.75	615.75
$\mathcal{L}_{\texttt{con.}}$	108.97	1138.76	61.01	1138.76
$\mathcal{L}_{\widehat{\mathcal{U}}}$	68.18	96.98	62.50	96.98
$\mathcal{L}_{\mathcal{D}_\Lambda}$	79.98	86.33	78.48	86.33

5.3. Language models

Motivated by the keyboard application (Hard et al., 2018), where a single client uses a trained language model in multiple environments such as chat apps, email, and web input, we created a dataset that combines two very different types of language datasets: `conversation` and `document`. For `conversation`, we used the Cornell movie dataset that contain movie dialogues (Danescu-Niculescu-Mizil & Lee, 2011). For documents, we used the Penn TreeBank (PTB) dataset (Marcus et al., 1993). We created a single dataset by combining both of the above corpuses, with `conversation` and `document` as domains. We preprocessed the data to remove punctuations, capitalized the data uniformly, and computed a vocabulary of 10,000 most frequent words. We trained a two-layer LSTM model with momentum optimizer. The performance of the models are measured by their perplexity, that is the exponent of cross-entropy loss. The results are reported in Table 3. Of the two domains, the `document` domain is the one admitting the higher perplexity. For this domain, the test perplexity of the domain agnostic model is close to that of the model trained only on `document` data and is better than that of the model trained with the uniform distribution $\widehat{\mathcal{U}}$.

6. Conclusion

We introduced a new framework for federated learning, based on principled learning objectives, for which we presented a detailed theoretical analysis, a learning algorithm motivated by our theory, a new stochastic optimization solution for large-scale problems and several extensions. Our experimental results suggest that our solution can lead to significant benefits in practice. In addition, our framework and algorithms benefit from favorable fairness properties. This constitutes a global solution that we hope will be generally adopted in federated learning, and other related learning tasks such as domain adaptation.

7. Acknowledgements

We thank Corinna Cortes, Satyen Kale, Shankar Kumar, Rajiv Mathews, and Brendan McMahan for helpful comments and discussions.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Agarwal, N., Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, B. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Proceedings of NeurIPS*, pp. 7575–7586, 2018.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. Clustering with Bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *NIPS*, pp. 137–144, 2006.

Bickel, P. J., Hammel, E. A., and O'Connell, J. W. Sex bias in graduate admissions: Data from Berkeley. *Science*, 187(4175):398–404, 1975. ISSN 0036-8075.

Blake, C. L. UCI repository of machine learning databases, Irvine, University of California. *http://www.ics.uci.edu/~mlearn/MLRepository*, 1998.

Blitzer, J., Dredze, M., and Pereira, F. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of ACL 2007*, Prague, Czech Republic, 2007.

Chen, R. S., Lucier, B., Singer, Y., and Syrgkanis, V. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, pp. 4705–4714, 2017.

Cortes, C. and Mohri, M. Domain adaptation and sample bias correction theory and algorithm for regression. *Theor. Comput. Sci.*, 519:103–126, 2014.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pp. 442–450, 2010.

Cortes, C., Mohri, M., and Muñoz Medina, A. Adaptation algorithm and theory based on generalized discrepancy. In *KDD*, pp. 169–178, 2015.

Danescu-Niculescu-Mizil, C. and Lee, L. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pp. 76–87. Association for Computational Linguistics, 2011.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training GANs with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Dredze, M., Blitzer, J., Talukdar, P. P., Ganchev, K., Graca, J., and Pereira, F. Frustratingly Hard Domain Adaptation for Parsing. In *Proceedings of CoNLL 2007*, Prague, Czech Republic, 2007.

Duchi, J. C., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *ICML*, pp. 272–279, 2008.

Farnia, F. and Tse, D. A minimax approach to supervised learning. In *Proceedings of NIPS*, pp. 4240–4248, 2016.

Ganin, Y. and Lempitsky, V. S. Unsupervised domain adaptation by backpropagation. In *ICML*, volume 37, pp. 1180–1189, 2015.

Gauvain, J.-L. and Chin-Hui. Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pp. 580–587, 2014.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pp. 2066–2073, 2012.

Gong, B., Grauman, K., and Sha, F. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, volume 28, pp. 222–230, 2013a.

Gong, B., Grauman, K., and Sha, F. Reshaping visual datasets for domain adaptation. In *NIPS*, pp. 1286–1294, 2013b.

Hard, A., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Hardt, M., Price, E., Srebro, N., et al. Equality of opportunity in supervised learning. In *Proceedings of NIPS*, pp. 3315–3323, 2016.

Hoffman, J., Kulis, B., Darrell, T., and Saenko, K. Discovering latent domains for multisource domain adaptation. In *ECCV*, volume 7573, pp. 702–715, 2012.

Hoffman, J., Rodner, E., Donahue, J., Saenko, K., and Darrell, T. Efficient learning of domain-invariant image representations. In *ICLR*, 2013.

Hoffman, J., Mohri, M., and Zhang, N. Algorithms and theory for multiple-source adaptation. In *Proceedings of NeurIPS*, pp. 8256–8266, 2018.

Jelinek, F. *Statistical Methods for Speech Recognition*. The MIT Press, 1998.

Jiang, J. and Zhai, C. Instance Weighting for Domain Adaptation in NLP. In *Proceedings of ACL 2007*, pp. 264–271, Prague, Czech Republic, 2007. Association for Computational Linguistics.

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

Koltchinskii, V. and Panchenko, D. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30, 2002.

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

Lee, J. and Raginsky, M. Minimax statistical learning and domain adaptation with Wasserstein distances. *arXiv preprint arXiv:1705.07815*, 2017.

Legetter, C. J. and Woodland, P. C. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, pp. 171–185, 1995.

Liu, J., Zhou, J., and Luo, X. Multiple source domain adaptation: A sharper bound using weighted Rademacher complexity. In *Technologies and Applications of Artificial Intelligence (TAAI), 2015 Conference on*, pp. 546–553. IEEE, 2015.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In *ICML*, volume 37, pp. 97–105, 2015.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Multiple source adaptation and the Rényi divergence. In *UAI*, pp. 367–374, 2009a.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009b.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In *NIPS*, pp. 1041–1048, 2009c.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of english: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

Martínez, A. M. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):748–763, 2002.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, pp. 1273–1282, 2017.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, volume 28, pp. 10–18, 2013.

Namkoong, H. and Duchi, J. C. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pp. 2208–2216, 2016.

Nemirovski, A. S. and Yudin, D. B. *Problem complexity and Method Efficiency in Optimization*. Wiley, 1983.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

Pietra, S. D., Pietra, V. D., Mercer, R. L., and Roukos, S. Adaptive language modeling using minimum discriminant estimation. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pp. 103–106, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

Raju, A., Hedayatnia, B., Liu, L., Gandhe, A., Khatri, C., Metallinou, A., Venkatesh, A., and Rastrow, A. Contextual language model adaptation for conversational agents. *arXiv preprint arXiv:1806.10215*, 2018.

Rakhlin, S. and Sridharan, K. Optimization, learning, and games with predictable sequences. In *Proceedings of NIPS*, pp. 3066–3074, 2013.

Rawls, J. *A theory of justice*. Harvard University Press, 2009.

Roark, B. and Bacchiani, M. Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of HLT-NAACL*, 2003.

Rockafellar, R. T. *Convex analysis*. Princeton University Press, 1997.

Rosenfeld, R. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer Speech and Language*, 10:187–228, 1996.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *ECCV*, volume 6314, pp. 213–226, 2010.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3329–3337. JMLR. org, 2017.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *ICCV*, pp. 4068–4076, 2015.

Williamson, R. C. and Menon, A. K. Fairness risk measures. *arXiv preprint arXiv:1901.08665*, 2019.

Woodworth, B. E., Wang, J., Smith, A. D., McMahan, B., and Srebro, N. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Proceedings of NeurIPS*, pp. 8505–8515, 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Xu, Z., Li, W., Niu, L., and Xu, D. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*, volume 8691, pp. 628–643, 2014.

Yang, J., Yan, R., and Hauptmann, A. G. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, pp. 188–197, 2007.

Zhang, K., Gong, M., and Schölkopf, B. Multi-source domain adaptation: A causal view. In *AAAI*, pp. 3150–3157, 2015.

A. Related work

Here, we briefly discuss several learning scenarios and work related to our study of federated learning.

The problem of federated learning is closely related to other learning scenarios where there is a mismatch between the source distribution and the target distribution. This includes the problem of *transfer learning* or *domain adaptation* from a single source to a known target domain (Ben-David, Blitzer, Crammer, and Pereira, 2006; Mansour, Mohri, and Rostamizadeh, 2009b; Cortes and Mohri, 2014; Cortes, Mohri, and Muñoz Medina, 2015), either through unsupervised adaptation techniques (Gong et al., 2012; Long et al., 2015; Ganin & Lempitsky, 2015; Tzeng et al., 2015), or via lightly supervised ones (some amount of labeled data from the target domain) (Saenko et al., 2010; Yang et al., 2007; Hoffman et al., 2013; Girshick et al., 2014). This also includes previous applications in natural language processing (Dredze et al., 2007; Blitzer et al., 2007; Jiang & Zhai, 2007; Raju et al., 2018), speech recognition (Legetter & Woodland, 1995; Gauvain & Chin-Hui, 1994; Pietra et al., 1992; Rosenfeld, 1996; Jelinek, 1998; Roark & Bacchiani, 2003), and computer vision (Martínez, 2002)

A problem more closely related to that of federated learning is that of *multiple-source adaptation*, first formalized and analyzed theoretically by Mansour, Mohri, and Rostamizadeh (2009c;a) and later studied for various applications such as object recognition (Hoffman et al., 2012; Gong et al., 2013a;b). Recently, (Zhang et al., 2015) studied a causal formulation of this problem for a classification scenario, using the same combination rules as Mansour et al. (2009c;a). The problem of *domain generalization* (Pan & Yang, 2010; Muandet et al., 2013; Xu et al., 2014), where knowledge from an arbitrary number of related domains is combined to perform well on a previously unseen domain is very closely related to that of federated learning, though the assumptions about the information available to the learner and the availability of unlabeled data may differ.

In the multiple-source adaptation problem studied by Mansour, Mohri, and Rostamizadeh (2009c;a) and Hoffman, Mohri, and Zhang (2018), each domain k is defined by the corresponding distribution $\mathcal{D}_k$ and the learner has only access to a predictor h_k for each domain and no access to labeled training data drawn from these domains. The authors show that it is possible to define a predictor h whose expected loss $\mathcal{L}_{\mathcal{D}}(h)$ with respect to any distribution $\mathcal{D}$ that is a mixture of the source domains $\mathcal{D}_k$ is at most the maximum expected loss of the source predictors: $\max_k L_{\mathcal{D}_k}(h_{\mathcal{D}_k})$. They also provide an algorithm for determining h.

Our learning scenario differs from the one adopted in that work since we assume access to labeled training data from each domain $\mathcal{D}_k$. Furthermore, the predictor determined by the algorithm of Hoffman, Mohri, and Zhang (2018) belongs to a specific hypothesis set $\mathcal{H}'$, which is that of distribution weighted combinations of the domain predictors h_k, while, in our setup, the objective is to determine the best predictor in some global hypothesis set $\mathcal{H}$, which may include $\mathcal{H}'$ as a subset, and which is not depending on some domain-specific predictors.

Our optimization solution also differs from the work of Farnia & Tse (2016) and Lee & Raginsky (2017) on local minimax results, where samples are drawn from a single source $\mathcal{D}$, and where the generalization error is minimized over a set of locally ambiguous distributions $\widehat{\mathcal{D}}$, where $\widehat{\mathcal{D}}$ is the empirical distribution. The authors propose this metric for statistical robustness. In our work, we obtain samples from p unknown distributions, and the set of distributions D_λ over which we optimize the expected loss is fixed and independent of samples. Furthermore, the source distributions can differ arbitrarily and need not be close to each other. In reverse, we note that our stochastic algorithm can be used to minimize the loss functions proposed in (Farnia & Tse, 2016; Lee & Raginsky, 2017).

B. Extensions

In this section, we briefly discuss several extensions of the framework, theory and algorithms that we presented.

B.1. Domain definitions

The choice of the domains can significantly impact learnability in federated learning. In view of our learning bounds, if the number of domains, p, is large and Λ is the full simplex, $\Lambda = \Delta_p$, then the models may not generalize well. Thus, if the number of clients is very large, using each client as a domain may be a poor choice for better generalization. Ideally, each domain is represented with a sufficiently large number of samples and is relatively homogeneous or pure. This suggests using a clustering algorithm for defining the domains based on the similarity of the client distributions. Different Bregman divergences could be used to define the divergence or similarity between distributions. Thus, techniques such as those of

Banerjee, Merugu, Dhillon, and Ghosh (2005) could be used to determine clusters of clients using a suitable Bregman divergence.

Client clusters can also be determined based on domain expertise. For example, in federated keyboard next word prediction (Hard et al., 2018), domains can be chosen to be the native language of the clients. If the model is used in variety of applications, domains can also be based on the application of interest. For example, the keyboard in (Hard et al., 2018) is used in chat apps, long form text input apps, and web inputs. Here, domains can be the app that was used. Training models agnostically ensures that the user experience is favorable in all apps.

B.2. Incorporating a prior on Λ

Agnostic federated learning as defined in (1) treats all domains equally and does not incorporate any prior knowledge of λ. Suppose we have a prior distribution $p_\Lambda(\lambda)$ over $\lambda \in \Lambda$ at our disposal, then, we can modify (1) to incorporate that prior. If the loss function ℓ is the cross-entropy loss, then the agnostic loss can be modified as follows:

$$\max_{\lambda \in \Lambda} \left(\mathcal{L}_{D_\lambda}(h) + \log p_\Lambda(\lambda) \right). \tag{9}$$

In this formulation, larger weights are assigned to more likely domains. The generalization guarantees of Theorem 1 can be appropriately modified to include these changes. Furthermore, if the prior $p_\Lambda(\lambda)$ is a log-concave function of λ, then the new objective is convex in h and concave in λ and a slight modification of our proposed algorithm can be used to determine the global minima. We note that we could also adopt a multiplicative formulation with the prior multiplying the loss, instead of the additive one with the negative log of the probability in Equation 9.

B.3. Domain features and personalization

We studied agnostic federated learning, where we learn a model that performs well on all domains. First, notice that we do not make any assumption on the hypothesis set $\mathcal{H}$ and the hypotheses can use the domain k as a feature. Such models could be useful for applications where the target domain is known at inference time. Second, while this paper deals with learning a centralized model, the resulting model $h_{\mathcal{D}_\Lambda}$ can be combined with a personalized model, on the client's machine, to design better client-specific models. This can be done for example by learning an appropriate mixture weight $\alpha_k \in [0,1]$ to use a mixture $\alpha_k h_{\mathcal{D}_\Lambda} + (1 - \alpha_k) h_k$ of the domain agnostic centralized model $h_{\mathcal{D}_\Lambda}$ and a client- or domain-specific model h_k.

C. Learning-theoretical guarantees

C.1. Proof of Proposition 1

Consider the following two distributions with support reduced to a single element $x \in \mathcal{X}$ and two classes $\mathcal{Y} = \{0,1\}$: $\mathcal{D}_1(x,0) = 0$, $\mathcal{D}_1(x,1) = 1$, $\mathcal{D}_2(x,0) = \frac{1}{2}$, and $\mathcal{D}_2(x,1) = \frac{1}{2}$. Let $\Lambda = \{\delta_1, \delta_2\}$, where δ_k, $k = 1,2$, denotes the Dirac measure on index k. We will consider the case where the sample sizes m_k are all equal, that is $h_{\overline{\mathcal{U}}} = \frac{1}{2}(\mathcal{D}_1 + \mathcal{D}_2)$. Let p_0 denote the probability that h assigns to class 0 and p_1 the one it assigns to class 1. Then, the cross-entropy loss of a predictor h can be expressed as follows:

$$\begin{aligned}
\mathcal{L}_{\overline{\mathcal{U}}}(h) = \mathop{\mathbb{E}}_{(x,y)\sim\overline{\mathcal{U}}}\left[-\log p_y\right] &= \frac{1}{4}\log\frac{1}{p_0} + \frac{1}{2}\log\frac{1}{p_1} + \frac{1}{4}\log\frac{1}{p_1} \\
&= \frac{1}{4}\log\frac{1}{p_0} + \frac{3}{4}\log\frac{1}{p_1} \\
&= \mathsf{D}\big(\left(\tfrac{1}{4}, \tfrac{3}{4}\right) \,\|\, (p_0, p_1)\big) + \frac{1}{4}\log\frac{4}{1} + \frac{3}{4}\log\frac{4}{3} \\
&\geq \frac{1}{4}\log\frac{4}{1} + \frac{3}{4}\log\frac{4}{3},
\end{aligned}$$

where the last inequality follows the non-negativity of the relative entropy. Furthermore, equality is achieved when $p_0 = 1 - p_1 = \frac{1}{4}$, which defines $h_{\overline{\mathcal{U}}}$, the minimizer of $\mathcal{L}_{\overline{\mathcal{U}}}(h)$. In view of that, $\mathcal{L}_{\mathcal{D}_\Lambda}(h_{\overline{\mathcal{U}}})$ is given by the following:

$$\begin{aligned}\mathcal{L}_{\mathcal{D}_\Lambda}(h_{\overline{\mathcal{U}}}) &= \max\left(\mathcal{L}_{\delta_1}(\overline{\mathcal{U}}), \mathcal{L}_{\delta_2}(\overline{\mathcal{U}})\right)\\ &= \max\left\{\log\frac{4}{3}, \frac{1}{2}\log\frac{4}{1} + \frac{1}{2}\log\frac{4}{3}\right\}\\ &= \log\frac{4}{\sqrt{3}}.\end{aligned}$$

We now compute the loss of $h_{\mathcal{D}_\Lambda}$:

$$\begin{aligned}\min_{h\in\mathcal{H}}\mathcal{L}_{\mathcal{D}_\Lambda}(h) &= \min_{h\in\mathcal{H}}\max_{k\in[p]}\mathcal{L}_{\mathcal{D}_k}(h)\\ &= \min_{(p_0,p_1)\in\Delta_2}\max\left\{\log\frac{1}{p_1}, \frac{1}{2}\log\frac{1}{p_0} + \frac{1}{2}\log\frac{1}{p_1}\right\}\\ &= \min_{p_1\in[0,1]}\max\left\{\log\frac{1}{p_1}, \log\frac{1}{\sqrt{p_1(1-p_1)}}\right\}\\ &= \log 2,\end{aligned}$$

since $\frac{1}{2}$ is the solution of the convex optimization in p_1, in view of $\max\left\{\frac{1}{p_1}, \frac{1}{\sqrt{p_1(1-p_1)}}\right\} = \frac{1}{\sqrt{p_1(1-p_1)}} \leq \frac{1}{2}$ for $p_1 > \frac{1}{2}$.

C.2. Proof of Theorem 1

The proof is an extension of the standard proofs for Rademacher complexity generalization bounds (Koltchinskii & Panchenko, 2002; Mohri et al., 2018). Fix $\lambda \in \Lambda$. For any sample $S = S_1, \ldots, S_p$, define $\Psi(S_1, \ldots, S_p)$ by

$$\Psi(S_1, \ldots, S_p) = \sup_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right).$$

Let $S' = (S'_1, \ldots, S'_p)$ be a sample differing from $S = (S_1, \ldots, S_p)$ only by point $x'_{k,i}$ in S'_k and $x_{k,i}$ in S_k. Then, since the difference of suprema over the same set is bounded by the supremum of the differences, we can write

$$\begin{aligned}\Psi(S') - \Psi(S) &= \sup_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}'_\lambda}(h)\right) - \sup_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right)\\ &\leq \sup_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}'_\lambda}(h)\right) - \left(\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right)\\ &\leq \sup_{h\in\mathcal{H}}\mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}'_\lambda}(h)\\ &= \sup_{h\in\mathcal{H}}\sum_{k=1}^{p}\frac{\lambda_k}{m_k}\sum_{i=1}^{m_k}\ell(h(x'_{k,i}), y'_{k,i}) - \sum_{k=1}^{p}\frac{\lambda_k}{m_k}\sum_{i=1}^{m_k}\ell(h(x_{k,i}), y_{k,i})\\ &= \sup_{h\in\mathcal{H}}\frac{\lambda_k}{m_k}\left[\ell(h(x'_{k,i}), y'_{k,i}) - \ell(h(x_{k,i}), y_{k,i})\right]\\ &\leq \frac{\lambda_k M}{m_k}.\end{aligned}$$

Thus, by McDiarmid's inequality, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$ for any $h \in \mathcal{H}$:

$$\mathcal{L}_{\mathcal{D}_\lambda}(h) \leq \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \mathbb{E}\left[\max_{h\in\mathcal{H}}\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right] + M\sqrt{\sum_{k=1}^{p}\frac{\lambda_k^2}{2m_k}\log\frac{1}{\delta}}.$$

Therefore, by the union over Λ_ϵ, with probability at least $1 - \delta$, for any $h \in \mathcal{H}$ and $\lambda \in \Lambda_\epsilon$ the following holds:

$$\mathcal{L}_{\mathcal{D}_\lambda}(h) \leq \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \mathbb{E}\left[\max_{h\in\mathcal{H}}\mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right] + M\sqrt{\sum_{k=1}^{p}\frac{\lambda_k^2}{2m_k}\log\frac{|\Lambda_\epsilon|}{\delta}}.$$

By definition of Λ_ϵ, for any $\lambda \in \Lambda$, there exists $\lambda' \in \Lambda_\epsilon$ such that $\mathcal{L}_{\mathcal{D}_\lambda}(h) \le \mathcal{L}_{\mathcal{D}'_\lambda}(h) + M\epsilon$. In view of that, with probability at least $1 - \delta$, for any $h \in \mathcal{H}$ and $\lambda \in \Lambda$ the following holds:

$$\mathcal{L}_{\mathcal{D}_\lambda}(h) \le \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \mathbb{E}\left[\max_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right] + M\epsilon + M\sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{2m_k} \log\frac{|\Lambda_\epsilon|}{\delta}}.$$

The expectation appearing on the right-hand side can be bounded following standard proofs for Rademacher complexity upper bounds (see for example (Mohri et al., 2018)), leading to

$$\mathbb{E}\left[\max_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}_\lambda}(h) - \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h)\right] \le \mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda).$$

The sum $\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}$ can be expressed in terms of the skewness of Λ, using the following equalities:

$$m\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k} = \sum_{k=1}^{p} \frac{\lambda_k^2}{\frac{m_k}{m}} = \sum_{k=1}^{p} \frac{\lambda_k^2}{\frac{m_k}{m}} + \sum_{k=1}^{p} \frac{m_k}{m} - 2\sum_{k=1}^{p} \lambda_k + 1 = \sum_{k=1}^{p} \frac{(\lambda_k - \frac{m_k}{m})^2}{\frac{m_k}{m}} + 1 = \chi^2(\lambda \,\|\, \overline{\mathbf{m}}) + 1.$$

This completes the proof.

C.3. Proof of Lemma 1

For any $\lambda \in \Lambda$, define the set of vectors A_λ in $\mathbb{R}^m$ by

$$A_\lambda = \left\{ \left[\frac{\lambda_k}{m_k} \ell(h(x_{k,i}), y_{k,i}) \right]_{(k,i)\in[p]\times[m_k]} : \mathbf{x} \in \mathcal{X}^m, \mathbf{y} \in \mathcal{Y}^m \right\}.$$

For any $\mathbf{a} \in A_\lambda$, $\|\mathbf{a}\|_2 = \sqrt{\sum_{k=1}^{p} m_k \frac{\lambda_k^2}{m_k^2}} = \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}} \le \sqrt{\frac{\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}})}{m}}$. Then, by Massart's lemma, for any $\lambda \in \Lambda$, the following inequalities hold:

$$\begin{aligned}
\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) &= \mathop{\mathbb{E}}_{\substack{S_k \sim \mathcal{D}_k^{m_k} \\ \boldsymbol{\sigma}}} \left[\sup_{h\in\mathcal{H}} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \sigma_{k,i}\, \ell(h(x_{k,i}), y_{k,i}) \right] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{a}\in A} \sum_{k=1}^{p} \sum_{i=1}^{m_k} \sigma_{k,i} a_{k,i} \right] \\
&\le \sqrt{\frac{\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}})}{m}}\, \frac{\sqrt{2\log|A_\lambda|}}{m} \\
&= \frac{\sqrt{2\mathfrak{s}(\Lambda \,\|\, \overline{\mathbf{m}}) \log|A_\lambda|}}{m}.
\end{aligned}$$

By Sauer's lemma, the following holds for $m \ge d$: $|A_\lambda| \le \left(\frac{em}{d}\right)^d$. Plugging in the right-hand side in the inequality above completes the proof.

D. Alternative learning guarantees

An objective similar to that of AFL was considered in the context of multiple-source domain adaptation by Liu et al. (2015). The authors presented generalization bounds for a scenario where the target is based on some specific mixture λ of the source domains. Our theoretical results differ from those of this work in two ways. First, our generalization bounds do not hold for a single mixture weight λ but for any subset Λ of the simplex. Second, the complexity terms in the bounds presented by these authors are proportional to $\sqrt{m}\max_{k\in[p]} \frac{\lambda_k}{m_k}$, while our guarantees are in terms of $\sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}}$, which is strictly tighter. In particular, in the special case where $k = 2$, $\lambda_1 = \frac{1}{\sqrt{m}}$, $\lambda_2 = 1 - \lambda_1$ and $m_1 = 1$ and $m_2 = m - 1$, the bounds of Liu et al. (2015) are proportional to a constant and thus not informative, $\sqrt{m}\max_{k\in[p]} \frac{\lambda_k}{m_k} = 1$, while our guarantees are in terms of $\frac{1}{\sqrt{m}}$.

Our generalization error in Theorem 1 is particularly useful when Λ is a strict subset of the simplex, $\Lambda \subset \Delta_p$. If $\Lambda = \Delta_p$, we can give the following alternative learning guarantee based.

Theorem 3. *For any $\delta > 0$, with probability at least $1 - \delta$ over the draw of samples $S_k \sim \mathcal{D}_k^{m_k}$, the following inequality holds for all $h \in \mathcal{H}$ and $\lambda \in \Lambda$:*

$$L_{\mathcal{D}_\lambda}(h) \leq L_{\overline{\mathcal{D}}_\lambda}(h) + \sum_{k=1}^{p} \left(2\lambda_k \mathfrak{R}^k_{m_k}(\mathcal{G}) + \lambda_k M \sqrt{\frac{1}{2m_k} \log \frac{p}{\delta}} \right),$$

where $\mathfrak{R}^k_{m_k}(\mathcal{G})$ is the Rademacher complexity over domain $\mathcal{D}_k$ with m_k samples.

Proof. For a fixed $k \in [p]$, by a standard Rademacher complexity bound, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds for all $h \in \mathcal{H}$:

$$L_{\mathcal{D}_k}(h) \leq L_{\overline{\mathcal{D}}_k}(h) + 2\mathfrak{R}^k_{m_k}(\mathcal{G}) + M \sqrt{\frac{1}{2m_k} \log \frac{1}{\delta}}.$$

Summing up the inequalities for each $k \in [p]$ after multiplication by λ_k and using the union bound complete the proof. □

We will now compare the generalization bounds of Theorem 1 and Theorem 3. The Rademacher complexity term of the bound of Theorem 1, $\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda)$, is more favorable than that of Theorem 3, since by the sub-additivity of $\sup$ and the linearity of expectation, we can write

$$\begin{aligned}
\mathfrak{R}_{\mathbf{m}}(\mathcal{G}, \lambda) = \mathop{\mathbb{E}}_{\substack{S_k \sim \mathcal{D}_k^{m_k} \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}} \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sum_{i=1}^{m_k} \sigma_{k,i}\, \ell(h(x_{k,i}), y_{k,i}) \right] &\leq \mathop{\mathbb{E}}_{\substack{S_k \sim \mathcal{D}_k^{m_k} \\ \boldsymbol{\sigma}}} \left[\sum_{k=1}^{p} \frac{\lambda_k}{m_k} \sup_{h \in \mathcal{H}} \sum_{i=1}^{m_k} \sigma_{k,i}\, \ell(h(x_{k,i}), y_{k,i}) \right] \\
&= \sum_{k=1}^{p} \frac{\lambda_k}{m_k} \mathop{\mathbb{E}}_{\substack{S_k \sim \mathcal{D}_k^{m_k} \\ \boldsymbol{\sigma}}} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{m_k} \sigma_{k,i}\, \ell(h(x_{k,i}), y_{k,i}) \right] \\
&= \sum_{k=1}^{p} \lambda_k \mathfrak{R}^k_{m_k}(\mathcal{G}).
\end{aligned}$$

The comparison of the last terms of the two bounds, $M\sqrt{\frac{\mathfrak{s}(\lambda \,\|\, \overline{\mathbf{m}})}{2m} \log \frac{|\Lambda_\epsilon|}{\delta}}$ versus $M \sum_{k=1}^{p} \sqrt{\frac{1}{2m_k} \log \frac{p}{\delta}}$, depends on the covering number $|\Lambda_\epsilon|$. When $|\Lambda_\epsilon|$ is small, as in the case where Λ is a finite discrete set (in the extreme case reduced to a single element), then, the last term of the bound of Theorem 1 is more favorable. This is because $|\Lambda_\epsilon|$ is then smaller or in the same order of magnitude as p, while, by the sub-additivity of $\sqrt{\cdot}$, the following inequality holds:

$$\sqrt{\frac{\mathfrak{s}(\lambda \,\|\, \overline{\mathbf{m}})}{m}} = \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}} \leq \sum_{k=1}^{p} \sqrt{\frac{\lambda_k^2}{m_k}} = \sum_{k=1}^{p} \lambda_k \sqrt{\frac{1}{m_k}}.$$

On the other hand, when $|\Lambda_\epsilon| = O((\frac{1}{\epsilon})^p)$ as in the case where Λ is the full simplex, then $\log |\Lambda_\epsilon| = p\, O(\log \frac{1}{\epsilon})$ can be substantially larger than $\log p$ and the last term of the bound of Theorem 3 seems more favorable since, by the Cauchy-Schwarz inequality, the following inequality holds:

$$\sum_{k=1}^{p} \lambda_k \sqrt{\frac{1}{m_k}} \leq \sqrt{p} \sqrt{\sum_{k=1}^{p} \frac{\lambda_k^2}{m_k}} = \sqrt{p} \sqrt{\frac{\mathfrak{s}(\lambda \,\|\, \overline{\mathbf{m}})}{m}}.$$

In general, it is not clear which of the two bounds is more favorable. This depends on $\overline{\mathbf{m}}$, λ, and Λ_ϵ. Learning bounds improving upon both may be based on a careful interpolation, which we leave to future work.

E. Analysis of the optimization algorithm

E.1. Proof of Theorem 4

The time complexity of the algorithm follows the definitions of the complexity terms U_λ, U_w, and U_p the dimension d in Properties 1. To prove the convergence guarantee, we first state the following lemma.

Lemma 5. *Assume that the Property 1.1 holds. Then,*

$$\max_{\lambda\in\Lambda} \mathsf{L}(w^A, \lambda) - \min_{w\in\mathcal{W}} \max_{\lambda\in\Lambda} \mathsf{L}(w, \lambda) \leq \frac{1}{T} \max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \left\{ \sum_{t=1}^{T} \mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t) \right\}.$$

Proof. Recall that (w^A, λ^A) is the solution returned by the algorithm. First observe that L is convex in w and linear and thus concave in λ. Thus, by the generalized von Neumann's theorem, the following holds:

$$\begin{aligned}
\max_{\lambda\in\Lambda} \mathsf{L}(w^A, \lambda) - \min_{w\in\mathcal{W}} \max_{\lambda\in\Lambda} \mathsf{L}(w, \lambda) &= \max_{\lambda\in\Lambda} \mathsf{L}(w^A, \lambda) - \max_{\lambda\in\Lambda} \min_{w\in\mathcal{W}} \mathsf{L}(w, \lambda) && \text{(von Neumann's minimax)}\\
&\leq \max_{\lambda\in\Lambda} \left\{ \mathsf{L}(w^A, \lambda) - \min_{w\in\mathcal{W}} \mathsf{L}(w, \lambda^A) \right\} && \text{(subadd. of } \max\text{)}\\
&= \max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \left\{ \mathsf{L}(w^A, \lambda) - \mathsf{L}(w, \lambda^A) \right\}\\
&\leq \frac{1}{T} \max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \left\{ \sum_{t=1}^{T} \mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t) \right\}. && \text{(convexity in } w \text{ and lin. in } \lambda\text{)}
\end{aligned}$$

This completes the proof. □

In view of the lemma, to derive convergence guarantees for the algorithm, it suffices to bound $\mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t)$. Since L is linear in λ and convex in w, we have

$$\begin{aligned}
\mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t) &= \mathsf{L}(w_t, \lambda) - \mathsf{L}(w_t, \lambda_t) + \mathsf{L}(w_t, \lambda_t) - \mathsf{L}(w, \lambda_t)\\
&\leq (\lambda - \lambda_t)\nabla_\lambda \mathsf{L}(w_t, \lambda_t) + (w_t - w)\nabla_w \mathsf{L}(w_t, \lambda_t)\\
&\leq (\lambda - \lambda_t)\delta_\lambda \mathsf{L}(w_t, \lambda_t) + (w_t - w)\delta_w \mathsf{L}(w_t, \lambda_t)\\
&+ (\lambda - \lambda_t)(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)) + (w_t - w)(\nabla_w \mathsf{L}(w_t, \lambda_t) - \delta_w \mathsf{L}(w_t, \lambda_t)).
\end{aligned}$$

In view of these inequalities, by the subadditivity of $\max$, the following inequality holds:

$$\begin{aligned}
&\max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \left\{ \sum_{t=1}^{T} \mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t) \right\}\\
&\leq \max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \sum_{t=1}^{T} (\lambda - \lambda_t)\delta_\lambda \mathsf{L}(w_t, \lambda_t) + (w_t - w)\delta_w \mathsf{L}(w_t, \lambda_t)\\
&+ \max_{\substack{\lambda\in\Lambda\\ w\in\mathcal{W}}} \sum_{t=1}^{T} \lambda(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)) - w(\nabla_w \mathsf{L}(w_t, \lambda_t) - \delta_w \mathsf{L}(w_t, \lambda_t))\\
&+ \sum_{t=1}^{T} \lambda_t(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)) - w_t(\nabla_w \mathsf{L}(w_t, \lambda_t) - \delta_w \mathsf{L}(w_t, \lambda_t)).
\end{aligned} \tag{10}$$

We now bound each of the three terms above separately. For the first term, observe that for any $w \in \mathcal{W}$,

$$\begin{aligned}
&\sum_{t=1}^{T} (w_t - w)\delta_w \mathsf{L}(w_t, \lambda_t) \\
&= \frac{1}{2\gamma_w} \sum_{t=1}^{T} \|(w_t - w)\|_2^2 + \gamma_w^2 \|\delta_w \mathsf{L}(w_t, \lambda_t)\|_2^2 - \|(w_t - \gamma_w \delta_w \mathsf{L}(w_t, \lambda_t) - w)\|_2^2 \\
&\le \frac{1}{2\gamma_w} \sum_{t=1}^{T} \|(w_t - w)\|_2^2 + \gamma_w^2 \|\delta_w \mathsf{L}(w_t, \lambda_t)\|_2^2 - \|(w_{t+1} - w)\|_2^2 && \text{(property of projection)} \\
&= \frac{1}{2\gamma_w} \|(w_1 - w)\|_2^2 - \|(w_{T+1} - w)\|_2^2 + \frac{\gamma_w}{2} \sum_{t=1}^{T} \|\delta_w \mathsf{L}(w_t, \lambda_t)\|_2^2 && \text{(telescoping sum)} \\
&\le \frac{1}{2\gamma_w} \|(w_1 - w)\|_2^2 + \frac{\gamma_w}{2} \sum_{t=1}^{T} \|\delta_w \mathsf{L}(w_t, \lambda_t)\|_2^2 \\
&\le \frac{2R_{\mathcal{W}}^2}{\gamma_w} + \frac{\gamma_w}{2} \sum_{t=1}^{T} \|\delta_w \mathsf{L}(w_t, \lambda_t)\|_2^2 \\
&\le \frac{2R_{\mathcal{W}}^2}{\gamma_w} + \frac{\gamma_w}{2} \sum_{t=1}^{T} \|\delta_w \mathsf{L}(w_t, \lambda_t) - \nabla_w \mathsf{L}(w_t, \lambda_t) + \nabla_w \mathsf{L}(w_t, \lambda_t)\|_2^2.
\end{aligned}$$

Since the right-hand side does not depend on w, taking the maximum of both sides over $w \in \mathcal{W}$ and the expectation yields

$$\mathbb{E}\left[\max_{w \in \mathcal{W}} \sum_{t=1}^{T} (w_t - w)\delta_w \mathsf{L}(w_t, \lambda_t)\right] \le \frac{2R_{\mathcal{W}}^2}{\gamma_w} + \frac{\gamma_w T \sigma_w^2}{2} + \frac{T\gamma_w G_w^2}{2},$$

using the following identity:

$$\begin{aligned}
&\mathbb{E}\left[\|\delta_w \mathsf{L}(w_t, \lambda_t) - \nabla_w \mathsf{L}(w_t, \lambda_t) + \nabla_w \mathsf{L}(w_t, \lambda_t)\|_2^2\right] \\
&= \mathbb{E}\left[\|\delta_w \mathsf{L}(w_t, \lambda_t) - \nabla_w \mathsf{L}(w_t, \lambda_t)\|^2\right] - 2\,\mathbb{E}\left[\delta_w \mathsf{L}(w_t, \lambda_t) - \nabla_w \mathsf{L}(w_t, \lambda_t)\right] \cdot \nabla_w \mathsf{L}(w_t, \lambda_t) + \|\nabla_w \mathsf{L}(w_t, \lambda_t)\|_2^2 \\
&= \mathbb{E}\left[\|\delta_w \mathsf{L}(w_t, \lambda_t) - \nabla_w \mathsf{L}(w_t, \lambda_t)\|^2\right] + \|\nabla_w \mathsf{L}(w_t, \lambda_t)\|_2^2.
\end{aligned}$$

Similarly, using the projection property, the following inequality can be shown:

$$\mathbb{E}\left[\max_{\lambda \in \Lambda} \sum_{t=1}^{T} (\lambda - \lambda_t)\delta_\lambda \mathsf{L}(w_t, \lambda_t)\right] \le \frac{2R_\Lambda^2}{\gamma_\lambda} + \frac{\gamma_\lambda T \sigma_\lambda^2}{2} + \frac{T\gamma_\lambda G_\lambda^2}{2}.$$

For the second term, by the Cauchy-Schwarz inequality, we can write

$$\max_{\lambda \in \Lambda} \sum_{t=1}^{T} \lambda(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)) \le R_\Lambda \|\sum_{t=1}^{T} \nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)\|_2.$$

Taking the expectation of both sides and using Jensen's inequality yields

$$\mathbb{E}\left[\max_{\lambda \in \Lambda} \sum_{t=1}^{T} \lambda(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t))\right] \le R_\Lambda \sqrt{T} \sigma_\lambda.$$

Similarly, we obtain the following:

$$\mathbb{E}\left[\max_{w \in \mathcal{W}} w \nabla_w \mathsf{L}(w_t, \lambda_t) - \delta_w \mathsf{L}(w_t, \lambda_t)\right] \le R_{\mathcal{W}} \sqrt{T} \sigma_w.$$

For the third term, observe that the stochastic gradients at time t are unbiased, conditioned on λ_t, and w_t, hence,

$$\mathbb{E}\left[\sum_{t=1}^{T} \lambda_t(\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \delta_\lambda \mathsf{L}(w_t, \lambda_t)) - w_t(\nabla_w \mathsf{L}(w_t, \lambda_t) - \delta_w \mathsf{L}(w_t, \lambda_t))\right] = 0.$$

Combining the upper bounds just derived gives:

$$\mathbb{E}\left[\max_{\lambda \in \Lambda} \mathsf{L}(w^A, \lambda) - \min_{w \in \mathcal{W}} \max_{\lambda \in \Lambda} \mathsf{L}(w, \lambda)\right] \le \frac{2R_{\mathcal{W}}^2}{T\gamma_w} + \frac{\gamma_w(\sigma_w^2 + G_w^2)}{2} + \frac{2R_\Lambda^2}{T\gamma_\lambda} + \frac{\gamma_\lambda(\sigma_\lambda^2 + G_\lambda^2)}{2} + \frac{R_{\mathcal{W}}\sigma_w}{\sqrt{T}} + \frac{R_\Lambda \sigma_\lambda}{\sqrt{T}}.$$

Setting $\gamma_w = \frac{2R_{\mathcal{W}}}{\sqrt{T((\sigma_w^2 + G_w^2))}}$ and $\gamma_\lambda = \frac{2R_\Lambda}{\sqrt{T((\sigma_\lambda^2 + G_\lambda^2))}}$ to minimize this upper bound and using Lemma 5 completes the proof.

E.2. Proof of Lemma 2

The unbiasedness of $\delta_\lambda \mathsf{L}(w, \lambda)$ follows directly its definition. For the variance, observe that, for index $k \in [p]$, since the probability of not drawing domain k is $(1 - \frac{1}{p})$, the variance is given by the following

$$\begin{aligned}\operatorname*{Var}_k[\delta_\lambda \mathsf{L}(w, \lambda)] &= \left[1 - \frac{1}{p}\right][0 - \mathsf{L}_k(w)]^2 + \frac{1}{p}\sum_{k=1}^{p}\frac{1}{m_k}\sum_{i=1}^{m_k}[p\mathsf{L}_{k,i}(w) - \mathsf{L}_k(w)]^2 \\ &\leq \left[1 - \frac{1}{p}\right]M^2 + \frac{1}{p}\sum_{k=1}^{p}\frac{1}{m_k}\sum_{i=1}^{m_k}[pM]^2 = pM^2.\end{aligned}$$

Summing over all indices from $k \in [p]$ completes the proof.

E.3. Proof of Lemma 3

The time complexity and the unbiasedness follow from the definitions. We now bound the variance. Since $\nabla_w \mathsf{L}_{k,J_k}$ is an unbiased estimate of $\nabla_w \mathsf{L}_k(w)$ and we have:

$$\operatorname{Var}[\delta_w] = \sum_{k=1}^{p} \lambda_k^2 \operatorname{Var}\left[\nabla_w \mathsf{L}_{k,J_k}(w) - \nabla_w \mathsf{L}_k(w)\right] \leq \sum_{k=1}^{p} \lambda_k^2 \sigma^2(w, I) \leq R_\Lambda \sigma_I^2(w).$$

This completes the proof.

E.4. Proof of Lemma 4

The time complexity and the unbiasedness follow from the definitions. We now bound the variance. By definition for any w, λ,

$$\begin{aligned}\operatorname{Var}(\delta_w) &= \sum_{k=1}^{p}\frac{\lambda_k}{m_k}\sum_{j=1}^{m_k}\left(\nabla_w \mathsf{L}_{k,j}(w) - \mathsf{L}(w, \lambda)\right)^2 \\ &= \sum_{k=1}^{p}\frac{\lambda_k}{m_k}\sum_{j=1}^{m_k}\left(\nabla_w \mathsf{L}_{k,j}(w) - \mathsf{L}_k(w)\right)^2 + \sum_{k=1}^{p}\lambda_k(\mathsf{L}_k(w) - \mathsf{L}(w, h))^2 \\ &\leq \sigma_I^2(w) + \sigma_O^2(w),\end{aligned}$$

where the second equality follows from the unbiasedness of the stochastic gradients.

E.5. Comparison of PerDomain and Weighted stochastic gradients

For large values of p, to do a fair comparison, we need to average p independent copies of the Weighted-stochastic gradient, which we refer to as p-Weighted, and compare it with the PerDomain-stochastic gradient. Since the variance of the average of p i.i.d. random variables is $1/p$ times the individual variance, by Lemma 4, the following holds:

$$\operatorname{Var}(p\text{-}\textsc{Weighted}) = \frac{\sigma_I^2(w) + \sigma_O^2(w)}{p}.$$

Further, observe that $R_\Lambda = \max_{\lambda \in \Lambda} \sum_{k=1}^{p} \lambda_k^2 \geq \frac{1}{p}$. Thus,

$$\operatorname{Var}(\textsc{PerDomain}) \geq \frac{\sigma_I^2(w)}{p}.$$

Hence, the right choice of the stochastic variance of w depends on the application. If all domains are roughly equally weighted, then we have $R(\Lambda) \approx \frac{1}{p}$ and the PerDomain-variance is a more favorable choice. Otherwise, if $\sigma_O^2(w)$ is small, then the Weighted-stochastic gradient is more favorable.

F. Alternative optimization algorithms for AFL

F.1. Stochastic mirror descent

In this section, we extend our Stochastic-AFL algorithm to the case where a general mirror map is used, as in the mirror descent algorithm. The pseudocode of our general algorithm Stochastic-MD-AFL is given in Figure 5.

```
Algorithm STOCHASTIC-MD-AFL
Initialization: $w_0 \in \mathcal{W}$ and $\lambda_0 \in \Lambda$.
Parameters: step size $\gamma_w > 0$ and $\gamma_\lambda > 0$.
For $t = 1$ to $T$:
    1. Obtain stochastic gradients: $\delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1})$ and $\delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1})$.
    2. $w_t = \operatorname{argmin}_{w \in \mathcal{W}} \delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1}) \cdot w + \frac{1}{\gamma_w} B_w(w \| w_{t-1})$.
    3. $\lambda_t = \operatorname{argmin}_{\lambda \in \Lambda} \delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1}) \cdot \lambda + \frac{1}{\gamma_\lambda} B_\lambda(\lambda \| \lambda_{t-1})$.
Output: $w^A = \frac{1}{T} \sum_{t=1}^T w_t$ and $\lambda^A = \frac{1}{T} \sum_{t=1}^T \lambda_t$.
```

Figure 5. Pseudocode of the STOCHASTIC-MD-AFL algorithm.

Let Φ_w be a function defined over $\operatorname{Int}(\mathcal{W})$ that is of the Legendre type (Rockafellar, 1997), that is a proper convex and differential function such that $\nabla \Phi_w$ is a one-to-one mapping from $\operatorname{Int}(\mathcal{W})$ to $\nabla \Phi_w(\operatorname{Int}(\mathcal{W}))$. Let B_w denote the Bregman divergence associated to Φ_w. For all $w, w' \in \mathcal{W}$, we have

$$B_w(w \| w') = \Phi_w(w) - \Phi_w(w') - \nabla \Phi_w(w') \cdot (w - w').$$

Similarly let Φ_λ be a Legendre-type function defined over an open set whose closure contains Λ and let B_λ denote the corresponding Bregman divergence. To simplify the notation, we use $\| \cdot \|$ to denote the norm over both w and λ, where the usage becomes clear in the context. Let $\| \cdot \|_*$ denote the corresponding dual norms. We will assume that the following properties hold.

Properties 2. *Assume that the following properties hold for the loss function* L *and sets* $\mathcal{W}$ *and* $\Lambda \subseteq \Delta_p$*:*

1. Convexity*:* $w \mapsto \mathsf{L}(w, \lambda)$ *is convex for any* $\lambda \in \Lambda$.
2. Compactness*:* $\max_{\lambda \in \Lambda} \|\lambda\| \leq R_\Lambda$ *and* $\max_{w \in \mathcal{W}} \|w\| \leq R_\mathcal{W}$*, for some* $R_\Lambda > 0$ *and* $R_\mathcal{W} > 0$.
3. Bounded gradients*:* $\|\nabla_w \mathsf{L}(w, \lambda)\|_* \leq G_w$ *and* $\|\nabla_\lambda \mathsf{L}(w, \lambda)\|_* \leq G_\lambda$ *for all* $w \in \mathcal{W}$ *and* $\lambda \in \Lambda$.
4. Stochastic variance*:* $\mathbb{E}[\|\delta_w \mathsf{L}(w, \lambda) - \nabla_w \mathsf{L}(w, \lambda)\|_*^2] \leq (\sigma_w^*)^2$ *and* $\mathbb{E}[\|\delta_\lambda \mathsf{L}(w, \lambda) - \nabla_\lambda \mathsf{L}(w, \lambda)\|_*^2] \leq (\sigma_\lambda^*)^2$ *for all* $w \in \mathcal{W}$ *and* $\lambda \in \Lambda$.
5. Strong convexity of Φ*: assume that* Φ_w *is* α_w*-strongly convex and* Φ_λ *is* α_λ*-strongly convex with respect to the norm* $\| \cdot \|$*. Further, assume that both* Φ_w *and* Φ_λ *are Legendre-type functions.*

With these definitions, we can now prove convergence guarantees for STOCHASTIC-MD-AFL.

Theorem 4. *[Appendix E.1] Assume that the Properties 2 hold. Then, for the step sizes* $\gamma_w = \frac{R_\mathcal{W} \sqrt{\alpha_w}}{\sqrt{T((\sigma_w^*)^2 + G_w^2)}}$ *and* $\gamma_\lambda = \frac{R_\Lambda \sqrt{\alpha_\lambda}}{\sqrt{T((\sigma_\lambda^*)^2 + G_\lambda^2)}}$*, the following guarantee holds for* STOCHASTIC-MD-AFL*:*

$$\mathbb{E}\left[\max_{\lambda \in \Lambda} \mathsf{L}(w^A, \lambda) - \min_{w \in \mathcal{W}} \max_{\lambda \in \Lambda} \mathsf{L}(w, \lambda)\right] \leq \frac{2R_\mathcal{W} \sqrt{\alpha_w((\sigma_w^*)^2 + G_w^2)}}{\sqrt{T}} + \frac{2R_\Lambda \sqrt{\alpha_\lambda((\sigma_\lambda^*)^2 + G_\lambda^2)}}{\sqrt{T}} + \frac{R_\mathcal{W} \sigma_w^*}{\sqrt{T}} + \frac{R_\Lambda \sigma_\lambda^*}{\sqrt{T}}.$$

Proof. By Lemma 5, it suffices to bound $\sum_{t=1}^T \mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t)$. By (10), we can decompose this sum into three terms. The expectation of third term is zero (see proof of Theorem 4). We now bound $\sum_{t=1}^T (w_t - w) \nabla_w \mathsf{L}(w_t, \lambda_t)$. To do so, following (Mohri et al., 2018), we break step (2) of the algorithm into two equivalent steps:

$$\begin{aligned} v_t &= [\nabla \Phi_w]^{-1} (\nabla \Phi_w(w_{t-1}) - \gamma_w \delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1})). \\ w_t &= \operatorname*{argmin}_{w \in \mathcal{W}} B(w \| v_t). \end{aligned}$$

We can write

$$\begin{aligned}
&\sum_{t=1}^{T}(w_t - w)\delta_w \mathsf{L}(w_t, \lambda_t) \\
&= \frac{1}{\gamma_w}\sum_{t=1}^{T}\left(\nabla\Phi_w(w_t) - \nabla\Phi_w(v_{t+1})\right)\cdot(w_t - w) && \text{(def. of } v_t\text{)} \\
&= \frac{1}{\gamma_w}\sum_{t=1}^{T}\left(B(w\|w_t) - B(w\|v_{t+1}) + B(w_t\|v_{t+1})\right) && \text{(Breg. div. Identity)} \\
&\le \frac{1}{\gamma_w}\sum_{t=1}^{T}\left(B(w\|w_t) - B(w\|w_{t+1}) - B_w(w_{t+1}\|v_{t+1}) + B(w_t\|v_{t+1})\right) && \text{(Pythagorean ineq.)} \\
&= \frac{1}{\gamma_w}\left(B(w\|w_1) - B(w\|w_{T+1})\right) + \frac{1}{\gamma_w}\sum_{t=1}^{T}\left(-B_w(w_{t+1}\|v_{t+1}) + B(w_t\|v_{t+1})\right) && \text{(telescoping sum)} \\
&\le \frac{B(w\|w_1)}{\gamma_w} + \frac{1}{\gamma_w}\sum_{t=1}^{T}\left(B(w_t\|v_{t+1}) - B_w(w_{t+1}\|v_{t+1})\right).
\end{aligned}$$

The second sum can be analyzed as follows:

$$\begin{aligned}
&B(w_t\|v_{t+1}) - B_w(w_{t+1}\|v_{t+1}) \\
&= \Phi_w(w_t) - \Phi_w(w_{t+1}) - \nabla\Phi_w(v_{t+1})(w_t - w_{t+1}) \\
&\le \left(\nabla\Phi_w(w_t) - \nabla\Phi_w(v_{t+1})\right)(w_t - w_{t+1}) - \frac{\alpha_w}{2}\|w_t - w_{t+1}\|^2 && (\alpha\text{-strong convexity}) \\
&= \gamma_w\delta_w\mathsf{L}(w_t, \lambda_t)(w_t - w_{t+1}) - \frac{\alpha_w}{2}\|w_t - w_{t+1}\|^2 && \text{(def. of } v_{t+1}\text{)} \\
&\le \gamma_w\|\delta_w\mathsf{L}(w_t, \lambda_t)\|_*\|w_t - w_{t+1}\| - \frac{\alpha_w}{2}\|w_t - w_{t+1}\|^2 && \text{(def. of dual norm)} \\
&\le \frac{\gamma_w^2\|\delta_w\mathsf{L}(w_t, \lambda_t)\|_*^2}{2\alpha_w} && \text{(max. of 2nd deg. eq.)} \\
&\le \frac{\gamma_w^2(\|\delta_w\mathsf{L}(w_t, \lambda_t) - \nabla_w\mathsf{L}(w_t, \lambda_t)\|_*^2 + \|\nabla_w\mathsf{L}(w_t, \lambda_t)\|_*^2)}{\alpha_w}. && \text{(triangle ineq. and Cauchy-Schwarz)}
\end{aligned}$$

Summing the inequalities above and taking expectation yields

$$\mathbb{E}\left[\sum_{t=1}^{T}(w_t - w)\delta_w\mathsf{L}(w_t, \lambda_t)\right] \le \frac{R_w^2}{\gamma_w} + \frac{\gamma_w((\sigma_w^*)^2 + G_w^2)}{\alpha_w}.$$

Similarly it can be shown that

$$\mathbb{E}\left[\sum_{t=1}^{T}(\lambda - \lambda_t)\delta_\lambda\mathsf{L}(w_t, \lambda_t)\right] \le \frac{R_\lambda^2}{\gamma_\lambda} + \frac{\gamma_\lambda((\sigma_\lambda^*)^2 + G_\lambda^2)}{\alpha_w}.$$

For the second term, by the Cauchy-Schwarz inequality and Jensen's inequality, we have

$$\mathbb{E}\left[\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\lambda(\nabla_\lambda\mathsf{L}(w_t, \lambda_t) - \delta_\lambda\mathsf{L}(w_t, \lambda_t))\right] \le R_\Lambda\,\mathbb{E}\left[\|\sum_{t=1}^{T}\nabla_\lambda\mathsf{L}(w_t, \lambda_t) - \delta_\lambda\mathsf{L}(w_t, \lambda_t)\|_*\right] \le R_\Lambda\sqrt{T}\sigma_\lambda^*.$$

Similarly, we can show that the following inequality holds:

$$\mathbb{E}\left[\max_{w\in\mathcal{W}}\sum_{t=1}^{T}w(\nabla_w\mathsf{L}(w_t, \lambda_t) - \delta_w\mathsf{L}(w_t, \lambda_t))\right] \le R_{\mathcal{W}}\sqrt{T}\sigma_w^*.$$

Combining these inequalities, we obtain the following:

$$\begin{aligned}
&\mathbb{E}\left[\max_{\lambda\in\Lambda}\mathsf{L}(w^A, \lambda) - \min_{w\in\mathcal{W}}\max_{\lambda\in\Lambda}\mathsf{L}(w, \lambda)\right] \\
&\qquad\qquad \le \frac{1}{T}\left[\frac{R_w^2}{\gamma_w} + \frac{\gamma_w((\sigma_w^*)^2 + G_w^2)}{\alpha_w} + \frac{R_\lambda^2}{\gamma_\lambda} + \frac{\gamma_\lambda((\sigma_\lambda^*)^2 + G_\lambda^2)}{\alpha_\lambda} + \sqrt{T}\left(R_{\mathcal{W}}\sigma_w^* + R_\Lambda\sigma_\lambda^*\right)\right].
\end{aligned}$$

Plugging in the expressions of γ_w and γ_λ completes the proof. □

```
Algorithm NON-STOCHASTIC-AFL
Initialization: $w_0$ and $\lambda_0 \in \Lambda$.
Parameters: step size $\gamma^w_{t+1} = \frac{1}{\beta_w t}$ and $\gamma^\lambda_{t+1} = \frac{1}{\beta_\lambda t}$.
For $t = 1$ to $T$:
    1. Obtain gradients: $\nabla_w \mathsf{L}(w_{t-1}, \lambda_{t-1})$ and $\nabla_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1})$.
    2. $w_t = \textsc{Project}(w_{t-1} - \gamma^w_t \nabla_w \mathsf{L}(w_{t-1}, \lambda_{t-1}), \mathcal{W})$.
    3. $\lambda_t = \textsc{Project}(\lambda_{t-1} + \gamma^\lambda_t \nabla_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1}), \Lambda)$.
Output: $w^A = \frac{1}{T}\sum_{t=1}^T w_t$ and $\lambda^A = \frac{1}{T}\sum_{t=1}^T \lambda_t$.
```

Figure 6. Pseudocode of the NON-STOCHASTIC-AFL algorithm.

F.2. Algorithm for strongly convex objectives

When the loss function is strongly convex with respect to w and strongly concave with respect to λ, conditions which often hold in the presence of regularization terms, a more favorable convergence rate of $\mathcal{O}((\log T)/T)$ can be proven for the non-stochastic algorithm NON-STOCHASTIC-AFL whose pseudocode is given in Figure 6.

Theorem 5. *Assume that the objective function is β_w-strongly convex with respect to w and β_λ-strongly concave with respect to λ, and that Properties* 1.1 *and* 1.3 *hold. Then, the following guarantee holds for* NON-STOCHASTIC AFL*:*

$$\mathbb{E}\left[\max_{\lambda\in\Lambda} \mathsf{L}(w^A, \lambda) - \min_{w\in\mathcal{W}} \max_{\lambda\in\Lambda} \mathsf{L}(w, \lambda)\right] \le \frac{G_w^2 + G_\lambda^2}{2} \cdot \frac{1 + \log T}{T}.$$

Proof. By Lemma 5, it suffices to consider $\mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t)$. Since the function is strongly convex with respect to w and strongly concave with respect to λ,

$$\mathsf{L}(w_t, \lambda) - \mathsf{L}(w, \lambda_t) \le (\lambda - \lambda_t)\nabla_\lambda \mathsf{L}(w_t, \lambda_t) - \beta_\lambda \|\lambda - \lambda_t\|_2^2 + (w_t - w)\nabla_w \mathsf{L}(w_t, \lambda_t) - \beta_w \|w - w_t\|_2^2.$$

We bound the term corresponding to λ,

$$\begin{aligned}
&\sum_{t=1}^T \nabla_\lambda \mathsf{L}(\lambda_t, \lambda_t) - \frac{\beta_\lambda}{2}\|\lambda - \lambda_t\|_2^2 \\
&= \sum_{t=1}^T \frac{1}{2\gamma^\lambda_{t+1}} \left(\|\lambda_t - \lambda\|_2^2 + (\gamma^\lambda_{t+1})^2 \|\nabla_\lambda \mathsf{L}(\lambda_t, \lambda_t)\|_2^2 - \|\lambda_t - \gamma^\lambda_{t+1}\nabla_\lambda \mathsf{L}(\lambda_t, \lambda_t) - \lambda\|_2^2\right) - \frac{\beta_\lambda}{2}\|\lambda - \lambda_t\|_2^2 \\
&\overset{(a)}{\le} \sum_{t=1}^T \frac{1}{2\gamma^\lambda_{t+1}} \left(\|\lambda_t - \lambda\|_2^2 + (\gamma^\lambda_{t+1})^2 \|\nabla_\lambda \mathsf{L}(\lambda_t, \lambda_t)\|_2^2 - \|\lambda_{t+1} - \lambda\|_2^2\right) - \frac{\beta_\lambda}{2}\|\lambda - \lambda_t\|_2^2 \\
&\overset{(b)}{\le} \frac{\beta_\lambda}{2} \sum_{t=1}^T \left((t-1)\|\lambda_t - \lambda\|_2^2 - t\|\lambda_{t+1} - \lambda\|_2^2\right) + \frac{G_\lambda^2}{2\beta_\lambda} \sum_{t=1}^T \frac{1}{t} \\
&\le \frac{G_\lambda^2}{2\beta_\lambda}(1 + \log T),
\end{aligned}$$

where (a) follows from the property of projection and (b) follows from the definition of γ^λ_{t+1}. The following inequality can be shown in a similar way:

$$\sum_{t=1}^T (w_t - w)\nabla_w \mathsf{L}(w_t, \lambda_t) - \beta_w \|w - w_t\|_2^2 \le \frac{G_w^2}{2\beta_w}(1 + \log T).$$

Summing up the two inequalities above and using Lemma 5 completes the proof. □

```
Algorithm OPTIMISTIC STOCHASTIC-AFL
Initialization: $w_0$ and $\lambda_0 \in \Lambda$.
Parameters: step size $\gamma_w > 0$ and $\gamma_\lambda > 0$.
For $t = 1$ to $T$:
    1. Obtain stochastic gradients: $\delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1})$ and $\delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1})$.
    2. $w_t = \text{PROJECT}(w_{t-1} - 2\gamma_w \delta_w \mathsf{L}(w_{t-1}, \lambda_{t-1}) + \gamma_w \delta_w \mathsf{L}(w_{\max(t-2,0)}, \lambda_{\max(t-2,0)}), \mathcal{W})$.
    3. $\lambda_t = \text{PROJECT}(\lambda_{t-1} + 2\gamma_\lambda \delta_\lambda \mathsf{L}(w_{t-1}, \lambda_{t-1}) - \gamma_\lambda \delta_\lambda \mathsf{L}(w_{\max(t-2,0)}, \lambda_{\max(t-2,0)}), \Lambda)$.
Output: $w_T, \lambda_T$.
```

Figure 7. Pseudocode of the OPTIMISTIC STOCHASTIC-AFL algorithm.

F.3. Optimistic stochastic algorithm

Recently, Rakhlin & Sridharan (2013) and Daskalakis et al. (2017) gave an optimistic gradient descent algorithm for minimax optimizations. Our algorithm can also be modified to derive a stochastic optimistic algorithm, which we refer to as OPTIMISTIC-STOCHASTIC-AFL. The pseudocode of this algorithm is also given in Figure 7. However, the convergence analysis we have presented so far does not cover this algorithm.