LEARNING DISTRIBUTIONALLY ROBUST MODELS AT SCALE VIA COMPOSITE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

To train machine learning models that are robust to distribution shifts in the data, distributionally robust optimization (DRO) has been proven very effective. However, the existing approaches to learning a distributionally robust model either require solving complex optimization problems such as semidefinite programming or a first-order method whose convergence scales linearly with the number of data samples– which hinders their scalability to large datasets. In this paper, we show how different variants of DRO are simply instances of a finite-sum composite optimization for which we provide scalable methods. We also provide empirical results that demonstrate the effectiveness of our proposed algorithm with respect to the prior art in order to learn robust models from very large datasets.

1 INTRODUCTION

Conventional machine learning problem aims at learning a model based on the assumption that training data and test data come from same data distribution. However, this assumption may not hold in various practical learning problems where there is label shift (Zhang et al., 2020a), distribution shift (Sagawa et al., 2019), fairness constraints (Hashimoto et al., 2018), and adversarial examples (Sinha et al., 2017), to name a few.

Distributionally robust optimization (DRO), which has recently attracted remarkable attention from the machine learning community, is a common approach to deal with the aforementioned uncertainties (Duchi & Namkoong, 2016; Chen et al., 2017; Rahimian & Mehrotra, 2019). Defining the empirical distribution of the training data of size m by $\hat{\mathbb{P}}_m \triangleq \frac{1}{m}\sum_{i=1}^{m} \delta_{\hat{\xi}_i}$ where δ is the Dirac delta function, the goal of DRO formulation is to solve the following optimization problem

$$\inf_{\boldsymbol{x}} \left[\Psi(\boldsymbol{x}) \triangleq \sup_{\xi \in Q} \mathbb{E}_Q \left[\ell(\boldsymbol{x}; \xi) \right] \right], \tag{1}$$

where ξ is a data sample randomly drawn from distribution Q, $\ell(\boldsymbol{x}; \xi)$ is the corresponding loss function and $\mathbb{E}_Q \left[\ell(\boldsymbol{x}, \xi) \right]$ is the expected loss over distribution Q which belongs to uncertainty set $\mathcal{U}_m$. The uncertainty set $\mathcal{U}_m$ is defined as $\mathcal{U}_m \triangleq \{Q : d(Q, \hat{\mathbb{P}}_m) \leq \epsilon\}$ indicates the ball of a distribution with center $\hat{\mathbb{P}}_m$ and also $d(P, Q)$ is a distance measure between probability distribution P and Q. We note this uncertainty set captures the distribution shift hence Eq. (1) minimizes the worse data distribution. Prior studies (Ben-Tal et al., 2013; Bertsimas et al., 2018; Blanchet et al., 2019; Esfahani & Kuhn, 2018) considered different uncertainty sets (see Definition 3.1 in Esfahani & Kuhn (2018)) for which they proposed equivalent reformulations of Eq. (1) based on the specific choice of $\mathcal{U}_m$.

To solve the above min-max optimization problems, majority of prior studies heavily rely on either semidefinite programming (Esfahani & Kuhn, 2018) or stochastic primal-dual methods both for convex (Nemirovski et al., 2009; Juditsky et al., 2011; Yan et al., 2019; 2020; Namkoong & Duchi, 2016) and non-convex (deep learning) objectives (Yan et al., 2020). While primal-dual methods can be used as an approach to solve min-max optimization problems, it suffers from a few downsides. First and foremost, they need to store a probability distribution of constrained violation of dimension m corresponding to dual variables. Additionally, available primal-dual methods often demand data sampling that corresponds to the probability distribution over m data samples which introduces additional cost over uniform sampling. Finally, while majority of prior studies are limited to DRO problems with convex objectives, establishing tight convergence rate for DRO problems with penalty with non-convex objectives is still lacking.

To overcome these issues, in this paper, we consider three different reformulations of Eq. (1), corresponding to three different choices of uncertainty sets $\mathcal{U}_m$ namely, *(1) DRO with Wasserstein metrics*, *(2) DRO with χ^2 divergence metrics*, and *(3) DRO with regularized entropy metrics* (also known as KL) and show in Section 2 that all aforementioned DRO notions are indeed different instances of a *deterministic composite optimization* and can be solved by reducing to an instances of the following problem:

$$\min_{\boldsymbol{x}} \left[\Psi(\boldsymbol{x}) \triangleq r(\boldsymbol{x}) + \frac{1}{m} \sum\nolimits_{i=1}^{m} h_i(\boldsymbol{x}) + f\left(\frac{1}{m} \sum\nolimits_{i=1}^{m} g_i(\boldsymbol{x}) \right) \right], \tag{2}$$

where we suppose $r(\boldsymbol{x})$ is convex and a relatively simple function, $f(\boldsymbol{x}) : \mathbb{R}^p \to \mathbb{R}$ and $h_i(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ for $1 \leq i \leq m$ are scalar-valued functions, and $g_i(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^p$ for $1 \leq i \leq m$ are vector-valued functions. On the road to solve problem (2) at scale, we also develop a novel algorithm for heavily constrained optimization problems (Wang & Bertsekas, 2015; 2016; Narasimhan et al., 2020b) that rather surprisingly invokes a *single* projection through the course of optimization. This algorithm is of independent interest and addresses the scalability issues raised in applications such as fairness (Donini et al., 2018; Zafar et al., 2019).

We summarize the main **contributions** of our paper below:

- We provide a large-scale analysis of DRO with Wasserstein distance and heavily constrained reformulation when the objective function is strongly-convex. Our result relies on a novel mini-batch constraint sampling for handling heavily-constrained optimization problems. As summarized in Table. 1, our convergence analysis improves the state-of-the-art both in terms of the dependence on the convergence error ϵ as well as the number of constraints m.
- We represent a large-scale analysis of DRO with non-convex objectives and χ^2 or KL divergences.
- We verify our theoretical results through various extensive experiments on different datasets. In particular, we show that our proposed method outperforms recent methods in DRO for heavily constrained problems with a great reduction in time complexity over them.

The proofs of all the theorems are provided in the **appendix**.

1.1 Related Work

DRO and connections to heavily constrained optimization. As mentioned earlier, DRO has many different formulations, depending on the divergence metrics used (e.g., Wasserstein , χ^2 or KL). While Namkoong & Duchi (2016); Shapiro (2017); Duchi & Namkoong (2021) consider constrained or penalized DRO formulation, Sinha et al. (2017); Levy et al. (2020) formulate the underlying optimization problem as unconstrained. One of the contributions of our paper is to provide a unifying framework through the language of composite optimization and treat all these variants similarly.

In particular, when the objective function is convex, Levy et al. (2020) recently proposed scalable algorithms for different variants of the DRO problems with , e.g., χ^2 or KL divergence metrics. Our unifying approach readily extends those results to the more challenging non-convex setting for which we are unaware of any prior work with convergence guarantees (for instance, Hashimoto et al. (2018) studied DRO with χ^2-divergence but did not provide any convergence guarantee). Similarly, Kuhn et al. (2019); Esfahani & Kuhn (2018) formulated DRO with Wasserstein distance as an instance of constrained optimization. Notably, they require ti impose one constraint per training data point and to solve such a constrained problem they proposed a semi-definite program. Even though the formulation is very novel, it cannot scale. We, in contrast, consider such a heavily constrained optimization as an instance of a composite optimization for which we provide a scalable solution. What is rather surprising about our method is that it only checks a batch of constraints per iteration, inspired by Cotter et al. (2016), and performs a single projection at the final stage of the algorithm in order to provide an ϵ-optimal solution in the case of strongly convex objectives. Moreover, in contrast to Cotter et al. (2016), we do not keep a probability distribution over the set of constraints. We should also remark that our convergence guarantees achieve the known lower bounds in terms of accuracy ϵ and the number of constraints m. Finally, we should highlight the difference of our algorithm and Frank-Wolfe (FW) (Frank et al., 1956; Jaggi, 2013; Zhang et al., 2020b). While FW does not require a projection oracle, it performs a linear program over the set of constraints at each iteration. In contrast, our heavily- constrained optimization solution performs a single projection without the overhead of running a linear program at each iteration.

Reference	DRO type	# Constraint/Sample Checks to achieve ϵ error	objective
Midtouch Cotter et al. (2016)	Wasserstein	$O\left(\frac{\ln m}{\epsilon}+\frac{m^{1.5}(\ln m)^{1.5}}{\epsilon^{\frac{3}{2}}}+\frac{m(\ln m)^{2/3}}{\epsilon^{\frac{2}{3}}}+\frac{m^2\log m}{\sqrt{\epsilon}}\right)$	Strongly convex
SEVR Yu et al. (2021)	General Linearized Wasserstein	$O\left(m\ln\left(\frac{1}{\epsilon}\right)+\frac{D_L}{\epsilon}+\frac{(\ell+\kappa+1)^2D_u^4}{\epsilon^2}\right)$	Strongly convex-concave min-max
Theorem 4.1	**Wasserstein**	$\boldsymbol{O\left((m+\kappa\sqrt{m})\ln\frac{1}{\epsilon}\right)}$	**Optimally strongly convex**
Theorem 4.3	**χ^2 or KL**	$\boldsymbol{\tilde{O}\left(\min\{\frac{\sqrt{m}}{\epsilon},\frac{1}{\epsilon^{1.5}}\}\right)}$	**Non-convex**
Theorem D.1	**χ^2 or KL**	$\boldsymbol{O\left((m+\kappa\sqrt{m})\ln\frac{1}{\epsilon}\right)}$	**Optimally strongly convex**

Table 1: Comparison of our results with prior approached. All three approaches are using variance reduction techniques. D_u and D_L respectively denotes the upper bound on the distance of initial model from optimal model and initial optimality gap. Please see Yu et al. (2021) for more details. Finally, we note that while Midtouch approach in Cotter et al. (2016) requires additional storage of probability distribution of dimension m, our approach does not.

Stochastic composite optimization. The general stochastic composite optimization $\min_{\boldsymbol{x}}\left[\Psi(\boldsymbol{x})\triangleq r(\boldsymbol{x})+f\left(\mathbb{E}_\xi\left[g_\xi(\boldsymbol{x})\right]\right)\right]$ has recently received a lot of attentions (Qi et al., 2020b;a; Wang et al., 2017). Our reformulation of DRO variants is a finite-sum instance of this general problem. More concretely, Lian et al. (2017); Huo et al. (2018); Zhang & Xiao (2019a) aimed to solve the following finite-sum problem $\min_{\boldsymbol{x}}\left[\Psi(\boldsymbol{x})\triangleq r(\boldsymbol{x})+\frac{1}{n}\sum_{j=1}^n f_j\left(\frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x})\right)\right]$, using SVRG or SAGA (Defazio et al., 2014). In contrast, our proposed algorithm is inspired by Zhang & Xiao (2019a) and generalizes their method to the case where the extra terms $h_i(\boldsymbol{x})$ in Eq. (2) are non-zero. We should also note that Qi et al. (2020a) proposed a similar idea in the context of online learning for DRO problems with KL divergence. Our work in contrast provides guarantees for DRO with both constraints or penalty terms.

2 DRO via Finite-Sum Composite Optimization

In this section, we discuss in detail how a finite-sum composite optimization (2) can unify various notions of distributionally robust learning, where some of which rely on heavily constrained optimization subroutines. While much research effort has been devoted to develop a specialized algorithm for each notion, our reduction paves the way to developing a scalable algorithm, discussed in Section 3.

DRO with Wasserstein distance. Kuhn et al. (2019); Esfahani & Kuhn (2018) provide the equivalent and tractable reformulation of Eq. (1) which can be regarded as a heavily constrained optimization problem as follows:

$$\min_{\boldsymbol{x}} r(\boldsymbol{x})\triangleq\frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x})\qquad\text{subject to}\qquad\tilde{g}_i(\boldsymbol{x})\leq 0,\ \forall i\in[m].\tag{3}$$

where $\tilde{g}_i(\boldsymbol{x})$ are functions related to loss function as well as slack variables (please see Appendix A for more details). Naively solving optimization problem (3) suffers from the computational complexity due to the large number of constraints m. To efficiently solve the optimization problem (3), inspired by Mahdavi et al. (2012) and Cotter et al. (2016), we pursue the smoothed constrained reduction approach and introduce the augmented optimization problem (see Appendix B) of the form $\min_{\boldsymbol{x}}\Psi(\boldsymbol{x})\triangleq\left[r(\boldsymbol{x})+\gamma\ln\left(g(\boldsymbol{x})\right)\right]$ where $g_i(\boldsymbol{x})\triangleq\exp\left(\frac{\alpha\tilde{g}_i(\boldsymbol{x})}{\gamma}\right)$ and $g(\boldsymbol{x})=\frac{1}{m+1}\left[1+\sum_{i=1}^m g_i(\boldsymbol{x})\right]$. We can see that optimization problem (3) is a special case of the optimization problem Eq. (2) where $r(\boldsymbol{x})=f(\boldsymbol{x})$, $f(\frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x}))=\gamma\ln g(\boldsymbol{x})$, and $h(\boldsymbol{x})=0$. In contrast to Cotter et al. (2016) that requires an extra storage cost of probability distribution of dimension m, and relatively poor convergence rate in terms of m and accuracy ϵ, we propose an algorithm that simply checks a batch of constraints and achieves the optimum dependency in terms of m and ϵ.

DRO with χ^2-divergence. The second type of DRO problem we consider utilizes the χ^2-divergence metric as follows:

$$\min_{\boldsymbol{x}}\quad\max_{0\leq p_i\leq 1,\ \sum_{i=1}^m p_i=1}\sum_{i=1}^m p_i f_i(\boldsymbol{x}_i)-\gamma D_{\chi^2}(\boldsymbol{p}).\tag{4}$$

where the χ^2 divergence is defined as the distance between the uniform distribution and an arbitrary probability distribution $\boldsymbol{p}$, i.e., $D_{\chi^2}(\boldsymbol{p}) \triangleq \frac{m}{2}\sum_{i=1}^m \left(p_i - \frac{1}{m}\right)^2$. Levy et al. (2020) studied this problem only for the case of convex objectives. In this paper, we allow objective functions f_i for $1 \leq i \leq m$ to be both non-convex or strongly-convex. The following claim derives the equivalent finite-sum composite optimization.

Claim 2.1. *The optimization problem (4) is equivalent to the following problem:*

$$\min_{\boldsymbol{x}} \quad \left[\Psi(\boldsymbol{x}) \triangleq 1 - \frac{1}{2\gamma m}\sum_{i=1}^m [f_i(\boldsymbol{x})]^2 + \frac{1}{2\gamma}\left[\frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x})\right]^2\right] \tag{5}$$

We note that the optimization problem (5) fits into the formulation of finite-sum composite optimization (2) by choosing $r(\boldsymbol{x}) = 1$, $h(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^m -\frac{(f_i(\boldsymbol{x}))^2}{2\gamma}$ and $f(g(\boldsymbol{x})) = \frac{1}{2\gamma}\left[\frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x})\right]^2$ with $h_i(\boldsymbol{x}) = -\frac{f_i^2(\boldsymbol{x})}{2\gamma}$, $g_i(\boldsymbol{x}) = f_i(\boldsymbol{x})$ and $f(x) = \frac{x^2}{2\gamma}$.

DRO with KL divergence. Finally, for DRO with KL-divergence, usually considered in online settings (Qi et al., 2020a), we consider solving the following optimization problem:

$$\min_{\boldsymbol{x}} \quad \max_{0\leq p_i \leq 1,\, \sum_{i=1}^m p_i = 1} \left[\sum_{i=1}^m p_i f_i(\boldsymbol{x}_i) + \gamma H(p_1, \ldots, p_m)\right], \tag{6}$$

where $H(p_1, \ldots, p_m) = -\sum_{i=1}^m p_i \log p_i$ is the entropy function. To solve the min-max problem (6), it is straightforward to convert it to the equivalent stochastic composite optimization problem of the following form:

$$\min_{\boldsymbol{x}} \left[\Psi(\boldsymbol{x}) \triangleq \ln\left(\frac{1}{m}\sum_{i=1}^m \exp\left(\frac{f_i(\boldsymbol{x})}{\gamma}\right)\right)\right]. \tag{7}$$

As it can be seen, the optimization problem (7) fits into the composite optimization (2) by choosing $r(\boldsymbol{x}) = h(\boldsymbol{x}) = 0$ and $f(g(\boldsymbol{x})) = \ln\left(\frac{1}{m}\sum_{i=1}^m \exp\left(\frac{f_i(\boldsymbol{x})}{\gamma}\right)\right)$.

3 Our Proposed Algorithm

Having reduced the different notions of DRO to an instance of the composite optimization, in this section we describe our scalable approach for minimizing the objective $\Psi(\boldsymbol{x}) = r(\boldsymbol{x}) + \Phi(\boldsymbol{x})$ where $\Phi(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^m h_i(\boldsymbol{x}) + f(\frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x}))$. We note that the compositional structure in $\Phi(\cdot)$ leads to more challenges in optimization compared with the non-compositional finite-sum problem, since the stochastic gradient of the loss function is not an unbiased estimation of the full gradient. To overcome this issue, and by building on incremental variance reduction (Zhang & Xiao, 2019b), we propose a more general algorithm with a new ingredient in which we also employ variance reduction on the extra term $h(\boldsymbol{x}) = (1/m)\sum_{i=1}^m h_i(\boldsymbol{x})$. To handle $r(\cdot)$, similar to Zhang & Xiao (2019b), we assume r is convex and a relatively simple function. We follow the proximal gradient iterates (Nesterov, 2013; Beck, 2017) as follows:

$$\boldsymbol{x}^{(t+1)} = \Pi_r^\eta\left(\boldsymbol{x}^{(t)} - \eta\nabla\Phi(\boldsymbol{x}^{(t)})\right) \tag{8}$$

where we apply the proximal operator of $r(\boldsymbol{x})$ with the learning rate η as $\Pi_r^\eta(\boldsymbol{x}) \triangleq \arg\min_{\boldsymbol{y}}\left[r(\boldsymbol{y}) + \frac{1}{2\eta}\|\boldsymbol{y} - \boldsymbol{x}\|^2\right]$. By defining the proximal gradient mapping of Ψ as $\mathcal{G}_\eta(\boldsymbol{x}) \triangleq \frac{1}{\eta}\left[\boldsymbol{x} - \Pi_r^\eta\left(\boldsymbol{x} - \eta\nabla\Phi(\boldsymbol{x})\right)\right]$, the updating rule in Eq. (8) can be equivalently written as $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta\mathcal{G}_\eta(\boldsymbol{x}^{(t)})$. Given any $\boldsymbol{y}$ as an output of randomized algorithm, we say $\boldsymbol{y}$ achieves the stationary point of problem in Eq. (2) in expectation if $\mathbb{E}\left[\|\mathcal{G}_\eta(\boldsymbol{y})\|^2\right] \leq \epsilon$ holds. Our goal is to achieve an ϵ stationary point with the least number of calls to a (mini-batch) stochastic oracle.

Focusing on $\Phi(\cdot)$, as detailed in Algorithm 1, we apply three time-scale variance-reduced estimators for $g_i(\boldsymbol{x})$ and its gradient $\nabla g_i(\boldsymbol{x})$, as well as $h_i(\boldsymbol{x})$. For DRO with Wasserstein divergence metric with

Algorithm 1: Generalized Composite Incremental Variance Reduction (**GCIVR** ($\boldsymbol{x}^{(0)}$))

Inputs: Number of iterations $t = 1, \ldots, T$, learning rate η, initial global model $\boldsymbol{x}^{(0)}$, the size of epoch length τ_t, and mini-batch sizes of B_t and S_t at time t.

for $t = 1, \ldots, T$ **do**

Sample a mini-batch $\mathcal{B}^{(t)}$ with size B_t uniformly over $[m]$ and compute

$$\boldsymbol{y}_0^{(t)} = \frac{1}{B_t}\sum\nolimits_{\xi\in\mathcal{B}^{(t)}} g(\boldsymbol{x}_{\tau_t}^{(t)};\xi), \boldsymbol{z}_0^{(t)} = \frac{1}{B_t}\sum\nolimits_{\xi\in\mathcal{B}^{(t)}} \nabla g(\boldsymbol{x}_{\tau_t}^{(t)};\xi), \boldsymbol{w}_0^{(t)} = \frac{1}{B_t}\sum\nolimits_{\xi\in\mathcal{B}^{(t)}} \nabla h(\boldsymbol{x}_{\tau_t}^{(t)};\xi)$$

Compute $\tilde{\nabla}\Phi(\boldsymbol{x}_0^{(t)}) = \left(\boldsymbol{z}_0^{(t)}\right)^\top \left(f'(\boldsymbol{y}_0^{(t)})\right) + \boldsymbol{w}_0^{(t)}$

Update the model as follows: $\boldsymbol{x}_1^{(t)} = \Pi_r^\eta\left(\boldsymbol{x}_0^{(t)} - \tilde{\nabla}\Phi(\boldsymbol{x}_0^{(t)}))\right)$

for $j = 1, \ldots, \tau_t - 1$ **do in parallel**

Sample a mini-batch $\mathcal{S}_j^{(t)}$ with size S_t uniformly over $[m]$, and form the estimates

$$\boldsymbol{y}_j^{(t)} = \boldsymbol{y}_{j-1}^{(t)} + \frac{1}{S_t}\sum\nolimits_{\xi\in\mathcal{S}_j^{(t)}}\left[g(\boldsymbol{x}_j^{(t)};\xi) - g(\boldsymbol{x}_{j-1}^{(t)};\xi)\right] \tag{9}$$

$$\boldsymbol{z}_j^{(t)} = \boldsymbol{z}_{j-1}^{(t)} + \frac{1}{S_t}\sum\nolimits_{\xi\in\mathcal{S}_j^{(t)}}\left[\nabla g(\boldsymbol{x}_j^{(t)};\xi) - \nabla g(\boldsymbol{x}_{j-1}^{(t)};\xi)\right] \tag{10}$$

$$\boldsymbol{w}_j^{(t)} = \boldsymbol{w}_{j-1}^{(t)} + \frac{1}{S_t}\sum\nolimits_{\xi\in\mathcal{S}_j^{(t)}}\left[\nabla h(\boldsymbol{x}_j^{(t)};\xi) - \nabla h(\boldsymbol{x}_{j-1}^{(t)};\xi)\right] \tag{11}$$

Compute $\tilde{\nabla}\Phi(\boldsymbol{x}_j^{(t)}) = \left(\boldsymbol{z}_j^{(t)}\right)^\top \left(f'(\boldsymbol{y}_j^{(t)})\right) + \boldsymbol{w}_j^{(t)}$

Update the model as follows: $\boldsymbol{x}_{j+1}^{(t)} = \Pi_r^\eta\left(\boldsymbol{x}_j^{(t)} - \tilde{\nabla}\Phi(\boldsymbol{x}_j^{(t)}))\right)$

end

end

Output: Return a randomly selected solution from $\{\boldsymbol{x}_j^t\}_{j=0,\ldots,\tau_t}^{t=1,\ldots,T}$

optimally strongly objective, at the beginning of each epoch t we compute a full-batch gradient over the entire data samples $B_t = m$, i.e., $\boldsymbol{y}_0^{(t)} = \frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x}_{\tau_t}^{(t)}), \boldsymbol{z}_0^{(t)} = \frac{1}{m}\sum_{i=1}^m \nabla g_i(\boldsymbol{x}_{\tau_t}^{(t)}), \boldsymbol{w}_0^{(t)} = \frac{1}{m}\sum_{i=1}^m \nabla h_i(\boldsymbol{x}_{\tau_t}^{(t)})$. In contrast, for the case of DRO with χ^2 or KL divergence metrics and non-convex objectives we incrementally increase the size of the mini-batch $B_t \leq m$ in the beginning of each epoch until it reaches the point where we need to compute the full-batch of samples. We should also highlight that a mini-batch in case of Wasserstein DRO indicates a batch of constraints and in DRO with χ^2 or KL divergence metrics represents the number of data sample accessed. We denote the length of each epoch and the mini-batch size within each epoch with τ_t and S_t, respectively. In each epoch t and iteration j, we estimate mini-batch gradients from Eqs. (9), (10), and (11) in Algorithm 1, with some corrections term applied from the previous iteration. We note that the variance-reduced term corresponding to the correction is inspired by SARAH (Nguyen et al., 2017) and SPIDER (Fang et al., 2018). Finally, the algorithm returns a randomly selected solution from the iterates.

Distributed variant of Algorithm 1. As mentioned before, efficient training of the stochastic composite optimization problem has attracted increasing attention in recent years. Despite much progress, all of existing methods including Algorithm 1 only focus on the single-machine setting. To employ Algorithm 1 in a distributed setting with p machines, in Appendix E we propose a distributed variant of proposed algorithm and establish its convergence rate for convex and non-convex objectives which enjoys a speedup in terms of number of machines.

4 Convergence Analysis

In this section we establish the convergence of proposed algorithm for different DRO notions discussed in Section 2. We start by stating the general assumptions and then discuss the obtained rates. Due to lack of space we only include the rates on the convergence of DRO with Wasserstein metric for strongly convex, and χ^2 and KL divergence metrics for non-convex objectives. We defer

the analysis of χ^2 and KL divergence metrics with strongly objectives to Appendix D. To establish the convergence rates, we first introduce some standard assumptions.

Assumption 1. *We make the following assumptions on the components of objective* $\Psi(\boldsymbol{x}) = r(\boldsymbol{x}) + \frac{1}{m}\sum_{i=1}^m h_i(\boldsymbol{x}) + f(\frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x}))$*:*

1) $\|\nabla h(\boldsymbol{x}_1) - \nabla h(\boldsymbol{x}_2)\| \leq L_h \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ *where* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$. *We also assume that* $\|h(\boldsymbol{x}_1) - h(\boldsymbol{x}_2)\| \leq \ell_h \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$.
2) $\|f'(\boldsymbol{x}_1) - f'(\boldsymbol{x}_2)\| \leq L_f \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ *where* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}$. *We also assume that* $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq \ell_f \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$.
3) $\|\nabla g(\boldsymbol{x}_1;\xi) - \nabla g(\boldsymbol{x}_2;\xi)\| \leq L_g \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ *and* $\|g(\boldsymbol{x}_1;\xi) - g(\boldsymbol{x}_2;\xi)\| \leq \ell_g \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ *for* $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$.
4) *We suppose that* $\Psi(\boldsymbol{x})$ *in Eq. (2) is bounded from below that* $\Psi^* = \inf_{\boldsymbol{x}} \Psi(\boldsymbol{x}) > -\infty$.
5) *We assume* $r(\boldsymbol{x}) \in \mathbb{R} \cup \{\infty\}$ *is a convex and lower-semicontinues function.*

An immediate implication of above assumption is that $f(g(\boldsymbol{x})) \triangleq f(\frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x}))$ is smooth with module $\left(\ell_g^2 L_f + \ell_f L_g\right)$, hence $\Psi(\boldsymbol{x})$ is smooth with module $L_\Phi \triangleq \left[\left(\ell_g^2 L_f + \ell_f L_g\right) + L_h\right]$ (Wang et al., 2016; Zhang & Xiao, 2019b).

Assumption 2 (Unbiased estimation)**.** *We assume that the mini-batch of samples* ξ *over functions* g_i *for* $i = 1, 2, \ldots, m$ *is unbiased, that is* $\mathbb{E}\left[g(\boldsymbol{x};\xi)\right] = g(\boldsymbol{x})$ *and* $\mathbb{E}\left[\nabla g(\boldsymbol{x};\xi)\right] = \nabla g(\boldsymbol{x})$.

4.1 Optimally Strongly Convex Objectives

We here establish the convergence for optimally strongly convex objectives under Wasserstein metric as defined below.

Definition 1 (Optimally strongly convex)**.** *We say that the objective function* $\Psi(\boldsymbol{x})$ *is optimally strongly convex objective with module* μ *if* $\Psi(\boldsymbol{x}) - \Psi(\mathcal{P}_{\boldsymbol{x}^*}(\boldsymbol{x})) \geq \mu \left\|\boldsymbol{x} - \mathcal{P}_{\boldsymbol{x}^*}(\boldsymbol{x})\right\|^2$.

Following Cotter et al. (2016) and Mahdavi et al. (2012), we make the following assumption.

Assumption 3. *There exists a constant* ρ *such that if we define* $g(\boldsymbol{x}) = \max_{1\leq i\leq m} \tilde{g}_i(\boldsymbol{x})$ *we have* $\min_{g(\boldsymbol{x})=0} \|\nabla g(\boldsymbol{x})\|_2 \geq \rho$.

Assumption 4. *Function* $r(\boldsymbol{x})$ *is smooth with module* G.

Theorem 4.1. *Let us assume that* Ψ *is* μ*-optimally strongly convex. Let us set* $\tau_t = S_t = \sqrt{B_t} = \sqrt{m}$, $T = \frac{5}{\sqrt{m}\mu\eta}$, $\eta < \frac{2}{L_\Phi + \sqrt{L_\Phi^2 + 36G_0}}$ *where* $G_0 \triangleq 3(\ell_g^4 L_f^2 + \ell_f^2 L_g^2 + \ell_h^2)$, $\alpha > \frac{G}{\rho}$, *and* $\gamma = \frac{\exp(-K)}{\ln(m+1)}$. *Let us denote* $\boldsymbol{x}^{(k+1)} = \texttt{GCIVR}\left(\boldsymbol{x}^{(k)}\right)$ *for* $k = 0, \ldots, K-1$ *(using Algorithm 1). Under assumptions 1-4 and by letting* $K = \ln(1/\epsilon)$, *the solution of DRO with Wasserstein divergence is obtained by projecting* $\boldsymbol{x}^{(K)}$ *onto the constraint set* $\mathcal{K} = \{\boldsymbol{x} | g_i(\boldsymbol{x}) \leq 0, i = 1, 2, \ldots, m\}$, *i.e.,* $\bar{\boldsymbol{x}}^{(K)} = \Pi_{\mathcal{K}}(\boldsymbol{x}^{(K)})$. *In order to achieve* $r(\bar{\boldsymbol{x}}^{(K)}) - r(\boldsymbol{x}^{(*)}) \leq \epsilon$ *we require* $O\left((m + \kappa\sqrt{m})\ln\frac{1}{\epsilon}\right)$ *number of calls to the stochastic oracle.*

Comparison with previous results. Compared to the MidTouch approach by Cotter et al. (2016), our obtained rate improves both on the dependency on the number of constraints m, as well as convergence error ϵ. To better understand the intuition behind achieving such a double folded improvement in terms of m and ϵ, we note that unlike Cotter et al. (2016) which utilizes a primal-dual approach in order to obtain an approximate to the optimal distribution over the constraints, we directly find the optimal distribution exactly. Second, while Cotter et al. (2016) does not apply any variance reduction technique to primal variable $\boldsymbol{x}$, our algorithm benefits from variance reduction over $\boldsymbol{x}$ too. Furthermore, Theorem 4.1 entails tighter rate in terms of final accuracy compared to Yu et al. (2021). These results are summarized in Table. 1.

In Algorithm 1 for optimally strongly convex objectives, we need to do a single projection at the end. The following corollary bounds the error between the solution before and after the projection.

Corollary 4.2. *Under the assumptions made in Theorem 4.1, the error between the solution with and without projection is bounded by*

$$r(\bar{\boldsymbol{x}}^{(K)}) - r(\boldsymbol{x}^{(K)}) \leq G\left[\frac{\gamma\ln(m+1)}{\alpha\rho - G} + \frac{1}{\alpha\rho - G}O\left(\exp(-K)\right)\right] + \gamma\ln(m+1). \tag{12}$$

Therefore, with a proper choice of γ we can make the right hand side matches the convergence error of Theorem 4.1.

Remark 1. *It is worthy to highlight that through reduction of heavily-constrained optimization to composite optimization,* ***GCIVR*** *algorithm can be considered as an alternative method to solving constrained optimization problems via mini-batch sampling of constraints. In particular, Theorem 4.1 shows that under certain conditions, we can solve any heavily-constrained optimization problem with sampling a mini-batch of constraints and achieve similar guarantees compared to projection-based counterparts while avoiding the heavy dependency on the number of constraints. Another implication of Theorem 4.1 is that we can solve heavily constrained optimization with the sample complexity similar to that of an unconstrained optimization problem.*

Remark 2. *To understand the tightness of our obtained rate, consider* $\min_{\boldsymbol{x}} \left[\Psi(\boldsymbol{x}) \triangleq r(\boldsymbol{x}) + \frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x})\right]$ *which is an instance of our general optimization problem (2). Clearly, any lower bound to solve this instance also holds for the original optimization problem (2). Xie et al. (2019) provides a lower-bound of* $O\left((m+\sqrt{\kappa m})\ln(1/\epsilon)\right)$ *for above instance, matching our upper bound in terms of the dependency on the number of data samples, while the dependency on κ can still be improved. Furthermore, for the DRO problem with Wasserstein divergence metric, rather than solving constrained optimization problem we solve unconstrained compositional optimization problem. Thus, since solving constrained optimization is more complex than unconstrained optimization problem, we do not expect to obtain any better bound regarding m even for DRO with Wasserstein metric.*

Remark 3. *In distributed setting with p devices, we are able to improve the sample complexity to* $O\left(\left(m + \frac{m}{p} + \kappa\sqrt{m}\right)\ln\frac{1}{\epsilon}\right)$. *The proof can be found in Appendix F.3.*

4.2 Non-Convex Objectives

In this section, we estbalish the convergence of DRO problems with χ^2 and KL divergence metrics for non-convex objectives. To this end, we make an additional assumption on the variance of stochastic mini-batches.

Assumption 5 (Bounded variance)**.** *The mini-batch sampling has bounded variance that is* $\mathbb{E}\left[\|\nabla g(\boldsymbol{x};\xi) - \nabla g(\boldsymbol{x})\|^2\right] \leq \frac{\sigma^2}{B}$ *where B indicates the batch size.*

Theorem 4.3. *Under Assumptions 1,2, and 5 using Algorithm 1 for some positive constants β and $0 \leq \zeta < \sqrt{m}$, denote $T_0 \triangleq \frac{\sqrt{m}-\zeta}{\beta} = O(\sqrt{m})$. For $t \leq T_0$, let us set $\tau_t = S_t = \sqrt{B_t} = \beta t + \zeta$ and for $t > T_0$ set $\tau_t = S_t = \sqrt{m}$. Then, if $\eta < \frac{4}{L_\Phi + \sqrt{L_\Phi^2 + 12G_0}}$, after $T = \tilde{O}\left(\min(1/\sqrt{\epsilon}, 1/\sqrt{m}\epsilon)\right)$ iterations, with $\tilde{O}\left(\min\{\sqrt{m}/\epsilon, 1/\epsilon^{1.5}\}\right)$ number of calls to the stochastic oracle it holds that $\mathbb{E}\left[\left\|\mathcal{G}_\eta(\boldsymbol{x}^{(T)})\right\|^2\right] \leq \epsilon$, where we used $\tilde{O}(.)$ notation to hide terms with logarithmic dependency.*

We also remark that Assumption 5 is only needed for convergence until $T_0 = \frac{\sqrt{m}-\zeta}{\beta}$ and after $t > T_0$ we will not need this assumption. We also note that this assumption is not required in optimally strongly convex setting as we utilize a full batch at the beginning of each epoch.

Discussion on lower bound. As we discussed in Remark 2, any impossibility result for optimizing $\min_{\boldsymbol{x}} \left[\Psi(\boldsymbol{x}) \triangleq r(\boldsymbol{x}) + \frac{1}{m}\sum_{i=1}^m g_i(\boldsymbol{x})\right]$ also holds for general compositional optimization problem in Eq. (2). For former problem, Fang et al. (2018) shows that to achieve the stationary point with error ϵ we require at least $\tilde{O}\left(\min\{\frac{\sqrt{m}}{\epsilon}, \frac{1}{\epsilon^{1.5}}\}\right)$ (almost optimal) gradient oracle calls for strongly optimal objectives. As a consequence, for the DRO with χ^2 or KL divergence metrics we do not expect less number of constraint checks or sample complexity, respectively.

Remark 4. *In distributed setting with p devices, we can improve the convergence rate of DRO with χ^2 and KL metrics to* $\tilde{O}\left(\min\{\frac{\sqrt{m}}{p\epsilon} + \frac{\sqrt{m}}{\epsilon}, \frac{1}{p\epsilon^{1.5}} + \frac{1}{\epsilon^{1.5}}\}\right)$. *For details, please refer to Appendix F.3.*

5 EXPERIMENTS

In this section, we empirically examine the efficacy of the proposed algorithm in different use cases. The main algorithm that we compare against is Heavily-Constrained algorithm that uses a parametric multiplier model proposed in Narasimhan et al. (2020b). We call this algorithm Heavily-constrained, where they learn the Lagrange multiplier values using a parametric model such as a neural network. The tasks we apply the algorithms are mainly focused on fairness constraints since they fit greatly to the problem definition in this paper due to the large number of constraints they add to the main learning problem. In addition, similar to distribution robust optimization approaches, the main goal of fairness constraint is to find a solution that is agnostic to the distribution of protected groups. The code for the experiments is available at this link.

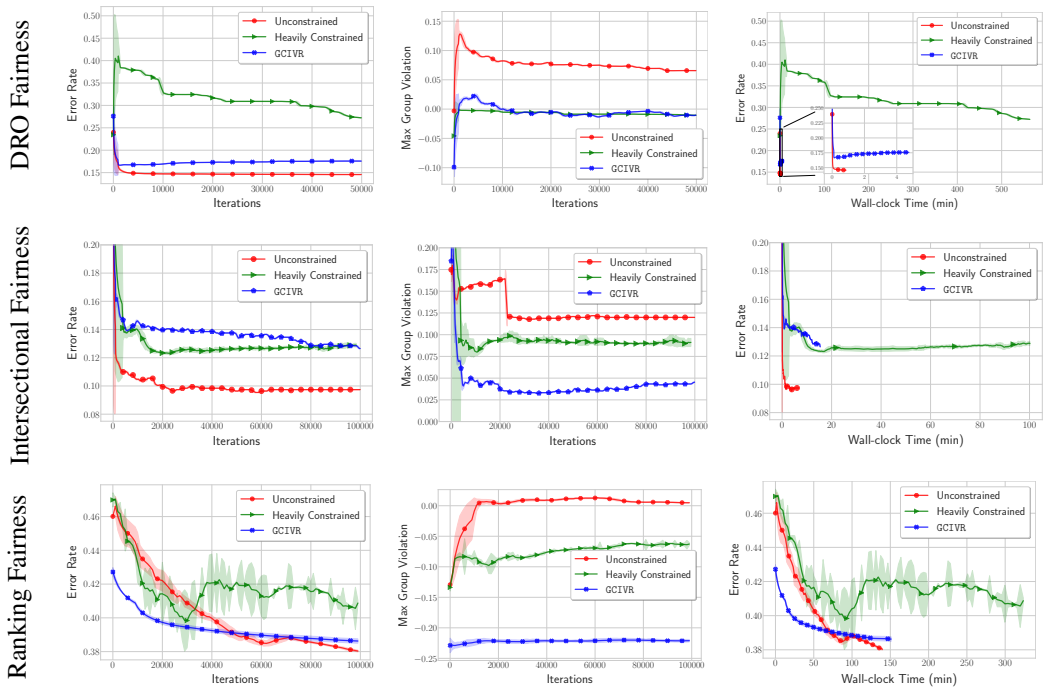
Distributionally robust optimization for fairness constraints. The first experiment setup is based on Wang et al. (2020); Narasimhan et al. (2020b), where we want to enforce equality of opportunity (Hardt et al., 2016) constraints for different groups while the group membership is noisy and changing during the training. Hence, the problem is to make the solution distributionally robust among different protected groups in the problem. Based on the setup in Narasimhan et al. (2020b), we assume we have access to the marginal probability of the true groups ($\mathbb{P}\left(g_i = j | \hat{g}_i = k\right)$, where g_i is the true group membership and $\hat{g}_i$ is the noisy group membership). Hence, to enforce fairness constraints, we consider all possible proxy groups using this marginal probability, which can increase the number of constraints greatly. The goal in this case for equal opportunity is to have the true positive rate of each group in the vicinity of the true positive rate (tpr) of the overall data, that is: $\mathsf{tpr}(g = j) \geq \mathsf{tpr}(ALL) - \epsilon$ for every proxy group we define.

In this experiment, we use the Adult dataset (Dua & Graff, 2017), and consider race groups of "white", "black", and "other" as protected groups. We train a linear classifier with logistic regression, and report the overall error rate of the classifier, as well as the maximum violation of the fairness constraints (equal opportunity) over true group memberships. We set $\epsilon = 0.05$ and the noise level to 0.3. Again, we compare with unconstrained optimization and heavily-constrained algorithm with a linear model as its multiplier model.First row of Figure 1 shows the results for this experiment, where the proposed **GCIVR** algorithm can achieve the same level of constraint violation on true group memberships as the heavily constraint while outperforms it in terms of overall error rate. Hence, the solution found by the **GCIVR** dominates heavily-constrained solution. In terms of runtime speed, the first row of Figure 1(c) clearly shows the advantage of **GCIVR** over heavily-constrained with much lower overhead to the unconstrained optimization.

Fairness Constraints on intersectional groups. In this task we ought to learn a linear classifier with logistic regression to predict the crime rate for a community on Communities and Crime dataset (Dua & Graff, 2017). This dataset contains 1994 instances of communities each with 128 features to predict the per capita crime rate for each community in the US. The labels are high and low crime rates to represent if a community is above the 70th percentile of the data or not. To add fairness constraints, we first determine different communities based on the percentages of the Black, Hispanic and Asian population in each community, as discussed in Narasimhan et al. (2020b); Cotter et al. (2019). Similar to Narasimhan et al. (2020b), we generate 1000 thresholds of the form $(\tau_1, \tau_2, \tau_3) \in [0, 1]^3$ to define each group. Consider the population of each race as (p_1, p_2, p_3), then a community belongs to group g_{τ_1,τ_2,τ_3} if $p_i \geq \tau_i \, \forall i \in [3]$. Then the fairness constraints enforce that the error rate of each group should be in the neighborhood of the overall error by margin of ϵ. That is $\mathsf{err}\left(g_{\tau_1,\tau_2,\tau_3}\right) \leq \mathsf{err}\left(ALL\right) + \epsilon$. Similar to Narasimhan et al. (2020b), we set $\epsilon = 0.01$ and not consider groups with less than 1% of data.

We compare with the unconstrained optimization and Heavily-constrained algorithm with a 2-layer neural network, each with 100 nodes as their Lagrange model, as described in their paper. We compare the trade-off between the error rate and maximum violation of fairness constraints among groups. The second row of Figure 1 portrays this comparison for the test dataset. As it can be inferred, the proposed **GCIVR** algorithm achieves the same error rate as Heavily-constrained, but both higher than the unconstrained optimization. However, comparing the maximum violation of constraints, it is clear that **GCIVR** outperforms both optimization methods. Also, the third figure shows the runtime of different algorithms. It is clear that **GCIVR** adding a minimal overhead to the unconstrained optimization, while heavily-constrained increases the runtime more than 100 times.

Per-query fairness constraints in ranking. In the third problem we evaluate our algorithm on the fairness in the ranking problem. In this case, we intend to impose per-query constraints on the



(a) Error Rate (b) Max Constraint Violation (c) Error Rate (Runtime)

Figure 1: Comparison of the proposed **GCIVR** algorithm with unconstrained optimization and heavily-constrained algorithm Narasimhan et al. (2020b) in three different tasks of fairness and DRO. Each row shows the result for one task based on the error rate, max constraint violations and the runtime of the codes. In all the cases solutions learned by **GCIVR** dominates the heavily-constrained solution and it converges way faster.

learning to rank problem as defined in Narasimhan et al. (2020a;b). Similar to those setups, we divide document-query pairs into two groups based on the 40th percentile of the their QualityScore features. Hence, in this problem, we want to learn a ranking function $f : \mathcal{D} \times \mathcal{Q} \to \mathbb{R}$, which maps a pair of document-query features to a real number as the score. Consider the groups of g_0 and g_1 as mentioned before, we want the error rate for these groups to be close to each other. In other terms, if the pairwise error rate is defined as: $\mathsf{err}_{i,j}(q) = \mathbb{E}\left[\mathbf{1}\left\{f(d,q) < f\left(d',q\right)\right\} \mid y > y', d \in g_i, d' \in g_j\right]$, where y and y' are the respective binary labels. Then, the fairness constraints can be satisfied as $|\mathsf{err}_{0,1}(q) - \mathsf{err}_{1,0}(q)| \leq \epsilon \;\; \forall q \in \mathcal{Q}$.

For this experiment, we use Microsoft Learning to Rank Dataset (MSLR-WEB10K) (Qin & Liu, 2013), which contains 10K queries and 136 features. For this experiment we use a non-convex objective, where the model is a two-layer neural network each with 128 nodes and cross-entropy as the loss function. We compare against the unconstrained optimization and the heavily-constrained algorithm with an one-layer neural network with 64 nodes as its multiplier model. We use 1000 queries in the training and 100 queries in the test datasets. The third row in Figure 1 shows the error rate and maximum violation of groups constraints. As it is clear **GCIVR** outperforms heavily-constrained in both error rate and maximum violations of groups by a large margin. In terms of runtime, **GCIVR** is very close to the unconstrained optimization, and about $2\times$ faster than heavily-constrained.

6 Conclusion

In this paper, we showed that many DRO problems or heavily constrained optimization problems can be cast into a general framework based on finite-sum composite optimization. To solve this composite finite-sum optimization, we introduced centralized and distributed algorithms. We theoretically illustrated that our algorithm converges with an almost optimal number of constraint checks (for Wasserstein distance) or gradient calls (for χ^2 or KL divergence metrics). Finally, we validated our theory with a number of experiments.

REFERENCES

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

Robert Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. *arXiv preprint arXiv:1707.01047*, 2017.

Andrew Cotter, Maya Gupta, and Jan Pfeifer. A light touch for heavily constrained sgd. In *Conference on Learning Theory*, pp. 729–771. PMLR, 2016.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pp. 300–332. PMLR, 2019.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *arXiv preprint arXiv:1802.08626*, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *arXiv preprint arXiv:1610.02581*, 2016.

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

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*, 2018.

Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pp. 1929–1938. PMLR, 2018.

Zhouyuan Huo, Bin Gu, Ji Liu, and Heng Huang. Accelerated method for stochastic composition optimization with nonsmooth regularization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

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

Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations Research & Management Science in the Age of Analytics*, pp. 130–166. INFORMS, 2019.

Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *arXiv preprint arXiv:2010.05893*, 2020.

Xiangru Lian, Mengdi Wang, and Ji Liu. Finite-sum composition optimization via variance reduced gradient descent. In *Artificial Intelligence and Statistics*, pp. 1159–1167. PMLR, 2017.

Mehrdad Mahdavi, Tianbao Yang, Rong Jin, Shenghuo Zhu, and Jinfeng Yi. Stochastic gradient descent with only one projection. In *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NIPS*, volume 29, pp. 2208–2216, 2016.

Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Serena Wang. Pairwise fairness for ranking and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5248–5255, 2020a.

Harikrishna Narasimhan, Andrew Cotter, Yichen Zhou, Serena Wang, and Wenshuo Guo. Approximate heavily-constrained learning with lagrange multiplier models. *Advances in Neural Information Processing Systems*, 33, 2020b.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pp. 2613–2621. PMLR, 2017.

Qi Qi, Zhishuai Guo, Yi Xu, Rong Jin, and Tianbao Yang. A practical online method for distributionally deep robust optimization. *arXiv preprint arXiv:2006.10138*, 2020a.

Qi Qi, Yan Yan, Zixuan Wu, Xiaoyu Wang, and Tianbao Yang. A simple and effective framework for pairwise deep metric learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pp. 375–391. Springer, 2020b.

Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. *arXiv preprint arXiv:1509.09259*, 2015.

Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Mengdi Wang and Dimitri P Bertsekas. Incremental constraint projection methods for variational inequalities. *Mathematical Programming*, 150(2):321–363, 2015.

Mengdi Wang and Dimitri P Bertsekas. Stochastic first-order methods with random constraint projection. *SIAM Journal on Optimization*, 26(1):681–717, 2016.

Mengdi Wang, Ji Liu, and Ethan X Fang. Accelerating stochastic composition optimization. *arXiv preprint arXiv:1607.07329*, 2016.

Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2): 419–449, 2017.

Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Michael I Jordan. Robust optimization for fairness with noisy protected groups. *arXiv preprint arXiv:2002.09343*, 2020.

Guangzeng Xie, Luo Luo, and Zhihua Zhang. A general analysis framework of lower complexity bounds for finite-sum optimization. *arXiv preprint arXiv:1908.08394*, 2019.

Yan Yan, Yi Xu, Qihang Lin, Lijun Zhang, and Tianbao Yang. Stochastic primal-dual algorithms with faster convergence than $o(1/\sqrt{T})$ for problems without bilinear structure. *arXiv preprint arXiv:1904.10112*, 2019.

Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Sharp analysis of epoch stochastic gradient descent ascent methods for min-max optimization. 2020.

Yaodong Yu, Tianyi Lin, Eric Mazumdar, and Michael I Jordan. Fast distributionally robust learning with variance reduced min-max optimization. *arXiv preprint arXiv:2104.13326*, 2021.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

Jingzhao Zhang, Aditya Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. Coping with label shift via distributionally robust optimisation. *arXiv preprint arXiv:2010.12230*, 2020a.

Junyu Zhang and Lin Xiao. A composite randomized incremental gradient method. In *International Conference on Machine Learning*, pp. 7454–7462. PMLR, 2019a.

Junyu Zhang and Lin Xiao. A stochastic composite gradient method with incremental variance reduction. *arXiv preprint arXiv:1906.10186*, 2019b.

Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pp. 4012–4023. PMLR, 2020b.