Stochastic Constrained DRO with a Complexity Independent of Sample Size

Anonymous authors
Paper under double-blind review

Abstract

Distributionally Robust Optimization (DRO), as a popular method to train robust models against distribution shift between training and test sets, has received tremendous attention in recent years. In this paper, we propose and analyze stochastic algorithms that apply to both non-convex and convex losses for solving Kullback–Leibler divergence constrained DRO problem. Compared with existing methods solving this problem, our stochastic algorithms not only enjoy competitive if not better complexity independent of sample size but also just require a constant batch size at every iteration, which is more practical for broad applications. We establish a nearly optimal complexity bound for finding an ϵ-stationary solution for non-convex losses and an optimal complexity for finding an ϵ-optimal solution for convex losses. Empirical studies demonstrate the effectiveness of the proposed algorithms for solving non-convex and convex constrained DRO problems.

1 Introduction

Large-scale optimization of DRO has recently garnered increasing attention due to its promising performance on handling noisy labels, imbalanced data and adversarial data (Namkoong & Duchi, 2017; Zhu et al., 2019; Qi et al., 2020a; Chen & Paschalidis, 2018). Various primal-dual algorithms can be used for solving various DRO problems (Rafique et al., 2021; Nemirovski et al., 2009). However, primal-dual algorithms inevitably suffer from additional overhead for handling a n dimensionality dual variable, where n is the sample size. This is an undesirable feature for large-scale deep learning, where n could be in the order of millions or even billions. Hence, a recent trend is to design dual-free algorithms for solving various DRO problems (Qi et al., 2021; Jin et al., 2021; Levy et al., 2020).

In this paper, we provide efficient dual-free algorithms solving the following constrained DRO problem, which are still lacking in the literature,

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\{\mathbf{p}\in\Delta_n : D(\mathbf{p},\mathbf{1}/n)\le\rho\}} \sum_{i=1}^{n} p_i \ell_i(\mathbf{w}) - \lambda_0 D(\mathbf{p},\mathbf{1}/n), \tag{1}$$

where $\mathbf{w}$ denotes the model parameter, $\mathcal{W}$ is closed convex set, $\Delta_n = \{\mathbf{p}\in\mathbb{R}^n : \sum_{i=1}^{n} p_i = 1, p_i \ge 0\}$ denotes a n-dimensional simplex, $\ell_i(\mathbf{w})$ denotes a loss function on the i-th data, $D(\mathbf{p},\mathbf{1}/n) = \sum_{i=1}^{n} p_i \log(p_i n)$ represents the Kullback–Leibler (KL) divergence measure between $\mathbf{p}$ and uniform probabilities $\mathbf{1}/n \in \mathbb{R}^n$, and ρ is the constraint parameter, and $\lambda_0 > 0$ is a small constant. A small KL regularization on $\mathbf{p}$ is added to ensure the objective in terms of $\mathbf{w}$ is smooth for deriving fast convergence.

There are several reasons for considering the above constrained DRO problem. First, existing dual-free algorithms are not satisfactory (Qi et al., 2021; Jin et al., 2021; Levy et al., 2020; Hu et al., 2021). They are either restricted to problems with no additional constraints on the dual variable $\mathbf{p}$ except for the simplex constraint (Qi et al., 2021; Jin et al., 2021), or restricted to convex analysis or have a requirement on the batch size that depends on accuracy level (Levy et al., 2020; Hu et al., 2021). Second, the Kullback–Leibler divergence measure is a more natural metric for measuring the distance between two distributions than other divergence measures, e.g., Euclidean distance. Third, compared with KL-regularized DRO problem without constraints, the above KL-constrained DRO formulation allows it to automatically decide a proper regularization effect that depends on the optimal solution by tuning the constraint upper bound ρ. The question to be addressed is the following:

> *Can we develop stochastic algorithms whose oracle complexity is optimal for both convex and non-convex losses, and its per-iteration complexity is independent of sample size n without imposing any requirements on the (large) batch size in the meantime?*

We address the above question by (i) deriving an equivalent primal-only formulation that is of a compositional form; (ii) designing two algorithms for non-convex losses and extending them for convex losses; (iii) establishing an optimal complexity for both convex and non-convex losses. In particular, for a non-convex and smooth loss function $\ell_i(\mathbf{w})$, we achieve an oracle complexity of $\widetilde{\mathcal{O}}(1/\epsilon^3)$[1] for finding an ϵ-stationary solution; and for a convex and smooth loss function, we achieve an oracle complexity of $\mathcal{O}(1/\epsilon^2)$ for finding an ϵ-optimal solution. We would like to emphasize that these results are on par with the best complexities that can be achieved by primal-dual algorithms (Huang et al., 2020; Namkoong & Duchi, 2016). But our algorithms have a per-iteration complexity of $\mathcal{O}(d)$, which is independent of the sample size n. The convergence comparison of different methods for solving (1) is shown in Table 1.

To achieve these results, we first convert the problem (1) into an equivalent problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \min_{\lambda\geq\lambda_0} F(\mathbf{w},\lambda) := \lambda \log\left(\frac{1}{n}\sum_{i=1}^{n}\exp\left(\frac{\ell_i(\mathbf{w})}{\lambda}\right)\right) + (\lambda-\lambda_0)\rho. \tag{2}$$

By considering $\mathbf{x} = (\mathbf{w}^\top,\lambda)^\top \in \mathbb{R}^{d+1}$ as a single variable to be optimized, the objective function is a compositional function of $\mathbf{x}$ in the form of $f(g(\mathbf{x}))$, where $g(\mathbf{x}) = \left[\lambda, \frac{1}{n}\sum_{i=1}^{n}\exp\left(\frac{\ell_i(\mathbf{w})}{\lambda}\right)\right] \in \mathbb{R}^2$ and $f(g) = g_1\log(g_2) + g_1\rho$. However, there are several challenges to be addressed for achieving optimal complexities for both convex and non-convex loss functions $\ell_i(\mathbf{w})$. First, the problem $F(\mathbf{x})$ is non-smooth in terms of $\mathbf{x}$ given the domain constraint $\mathbf{w}\in\mathcal{W}$ and $\lambda\geq\lambda_0$. Second, the outer function $f(g)$'s gradient is non-Lipschtiz continuous in terms of the second coordinate g_2 if λ is unbounded, which is essential for all existing stochastic compositional optimization algorithms. Third, to the best of our knowledge, no optimal complexity in the order of $\mathcal{O}(1/\epsilon^2)$ has been achieved for a convex compositional function except for Zhang & Lan (2021), which assumes f is convex and component-wisely non-decreasing and hence is not applicable to (2).

To address the first two challenges, we derive an upper bound for the optimal λ assuming that $\ell_i(\mathbf{w})$ is bounded for $\mathbf{w}\in\mathcal{W}$, i.e., $\lambda\in[\lambda_0,\tilde{\lambda}]$, which allows us to establish the smoothness condition of $F(\mathbf{x})$ and $f(g)$. Then we consider optimizing $\bar{F}(\mathbf{x}) = F(\mathbf{x}) + \delta_{\mathcal{X}}(\mathbf{x})$, where $\delta_{\mathcal{X}}(\mathbf{x}) = 0$ if $\mathbf{x}\in\mathcal{X} = \{\mathbf{x} = (\mathbf{w}^\top,\lambda)^\top : \mathbf{w}\in\mathcal{W}, \lambda\in[\lambda_0,\tilde{\lambda}]\}$. By leveraging the smoothness conditions of F and f, we design stochastic algorithms by utilizing a recursive variance-reduction technique to compute a stochastic estimator of the gradient of $F(\mathbf{x})$, which allows us to achieve a complexity of $\widetilde{\mathcal{O}}(1/\epsilon^3)$ for finding a solution $\bar{\mathbf{x}}$ such that $\mathbb{E}[\mathrm{dist}(0,\hat{\partial}\bar{F}(\bar{\mathbf{x}}))] \leq \epsilon$. To address the third challenge, we consider optimizing $\bar{F}_\mu(\mathbf{x}) = \bar{F}(\mathbf{x}) + \mu\|\mathbf{x}\|^2/2$ for a small μ. We prove that $\bar{F}_\mu(\mathbf{x})$ satisfies a Kurdyka-Łojasiewicz inequality, which allows us to boost the convergence of the aforementioned algorithm to enjoy an optimal complexity of $\mathcal{O}(1/\epsilon^2)$ for finding an ϵ-optimal solution to $\bar{F}(\mathbf{x})$. Besides the optimal algorithms, we also present simpler algorithms with worse complexity, which are more practical for deep learning applications without requiring two backpropagations at two different points per iteration as in the optimal algorithms.

2 Related Work

DRO springs from the robust optimization literature (Bertsimas et al., 2018; Ben-Tal et al., 2013) and has been extensively studied in machine learning and statistics (Namkoong & Duchi, 2017; Duchi et al., 2016; Staib & Jegelka, 2019; Deng et al., 2020; Qi et al., 2020b; Duchi & Namkoong, 2021), and operations research (Rahimian & Mehrotra, 2019; Delage & Ye, 2010). Depending on how to constrain or regularize the uncertain variables, there are constrained DRO formulations that specify a constraint set for the uncertain variables, and regularized DRO formulations that use a regularization term in the objective for regularizing

---

[1] $\widetilde{\mathcal{O}}$ omits a logarithmic dependence over ϵ.

Table 1: Summary of algorithms solving KL-constrained DRO problem. Complexity represents the oracle complexity for achieving $\mathbb{E}[\text{dist}(0, \hat{\partial}\bar{F}(\mathbf{x}))] \leq \epsilon$ or other first-order stationarity convergence for the non-convex setting and $\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}_*)] \leq \epsilon$ for the convex setting. Per Iter Cost denotes the per-iteration computational complexity. The algorithm styles include primal-dual (PD), primal only (P), and compositional (COM). “-" means not available in the original paper.

Setting	Algorithms	Reference	Complexity	Batch Size	Per Iter Cost	Style
Non-convex	PG-SMD2[2]	(Rafique et al., 2021)	$\mathcal{O}(1/\epsilon^4)$	$\mathcal{O}(1)$	$\mathcal{O}(n+d)$	PD
	AccMDA	(Huang et al., 2020)	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1)$	$\mathcal{O}(n+d)$	PD
	Dual SGM	(Levy et al., 2020)	-	$\mathcal{O}(1)$	$\mathcal{O}(d)$	P
	SCDRO	This work	$\mathcal{O}(1/\epsilon^4)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$	COM
	ASCDRO		$\widetilde{\mathcal{O}}(1/\epsilon^3)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$	COM
Convex	FastDRO[3]	(Levy et al., 2020)	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1/\epsilon)$	$\mathcal{O}(\frac{d}{\epsilon})$	P
	SPD	(Namkoong & Duchi, 2016)	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1)$	$\mathcal{O}(n+d)$	PD
	Dual SGM	(Levy et al., 2020)	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$	P
	RSCDRO	This work	$\mathcal{O}(1/\epsilon^3)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$	COM
	RASCDRO		$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$	COM

the uncertain variables (Levy et al., 2020). Duchi et al. (2016) showed that minimizing constrained DRO with f-divergence including a χ^2-divergence constraint and a KL-divergence constraint, is equivalent to adding variance regularization for the Empirical Risk Minimization (ERM) objective, which is able to reduce the uncertainty and improve the generalization performance of the model.

Primal-Dual Algorithms. Many primal-dual algorithms designed for the min-max problems can be directly applied to optimize the constrained DRO problem. The algorithms proposed in (Nemirovski et al., 2009; Juditsky et al., 2011; Yan et al., 2019; Namkoong & Duchi, 2016; Yan et al., 2020; Song et al., 2021; Alacaoglu et al., 2022) are applicable to solving (1) when ℓ is a convex function. Recently, Rafique et al. (2021) and Yan et al. (2020) proposed non-convex stochastic algorithms for solving non-convex strongly convex min-max problems, which are applicable to solving (1) when ℓ is a weakly convex function or smooth. Many primal-dual stochastic algorithms have been proposed for solving non-convex strongly concave problems with a state of the art oracle complexity of $\mathcal{O}(1/\epsilon^3)$ for finding a stationary solution (Huang et al., 2020; Luo et al., 2020; Tran-Dinh et al., 2020). However, the primal-dual algorithms require maintaining and updating an $\mathcal{O}(n)$ dimensional vector for updating the dual variable.

Constrained DRO. Recently, Levy et al. (2020) proposed sample independent algorithms based on gradient estimators for solving a group of DRO problems in the convex setting. To be more specific, they achieved a convergence rate of $\widetilde{\mathcal{O}}(1/\epsilon^2)$ for the χ^2-constrained/regularized and CVaR-constrained convex DRO problems and the batch size of logarithmically dependent on the inverse accuracy level $\mathcal{O}(\log(1/\epsilon))$ with the help of multi-level Monte-Carlo (MLMC) gradient estimator. For the KL-constrained DRO objective and other more general setting, they achieve a convergence rate of $\mathcal{O}(1/\epsilon^3)$ under a Lipschitz continuity assumption on the inverse CDF of the loss function and a mini-batch gradient estimator with a batch size in the order $\mathcal{O}(1/\epsilon)$ (please refer to Table 3 in Levy et al. (2020)). In addition, Levy et al. (2020) also proposed a simple stochastic gradient method for solving the dual expression of the DRO formulation, which is called Dual SGM. In terms of convergence, they only discussed the convergence guarantee for the χ^2-regularized and CVaR penalized convex DRO problems (cf. Claim 3 in their paper). However, there is still gap for proving the convergence rate of Dual SGM for non-convex KL-constrained DRO problems due to similar challenges mentioned in the previous section, in particular establishing the smoothness condition in terms of the primal variable and the Lagrangian multipliers (denoted as $\mathbf{x}, \nu, \eta$ respectively in their paper). This paper makes unique contributions for addressing these challenges by (i) removing η in Dual SGM and deriving the box constraint for our Lagrangian multiplier λ for proving the smoothness condition; (ii) establishing an optimal complexity in the order of $\mathcal{O}(1/\epsilon^3)$ in the presence of non-smooth box constraints, which, to the best of our knowledge, is the first time for solving a non-convex constrained compositional optimization problem.

---

[2] PG-SMD2 refers to PG-SMD algorithm under Assumption D2 in Rafique et al. (2021).

[3] FastDRO is name of the GitHub repository of Levy et al. (2020), and we use the name “FastDRO” to refer to the algorithm based on mini-batch gradient estimator in Levy et al. (2020).

Wang et al. (2021) studies the Sinkhorn distance constraint, a variant of Wasserstein distance based on entropic regularization. An efficient batch gradient descent with a bisection search algorithm has been proposed to obtain a near-optimal solution with an arbitrarily small sub-optimality gap. However, no non-asymptotic convergence results are established in their paper. Duchi & Namkoong (2021) developed a convex DRO framework with f-divergence constraints to improve model robustness. The author developed the finite-sample minimax upper and lower bounds and the non-asymptotic convergence rate of $\mathcal{O}(1/\sqrt{n})$, and provided the empirical studies on real distributional shifts tasks with existing interior point solver (Udell et al., 2014) and gradient descent with backtracking Armijo line-searches (Boyd et al., 2004). However, no stochastic algorithms that directly optimize the considered constrained DRO with non-asymptotic convergence rates are provided in their paper.

Regularized DRO. DRO with KL divergence regularization objective has shown superior performance for addressing data imbalanced problems (Qi et al., 2021; 2020a; Li et al., 2020; 2021). Jin et al. (2021) proposed a mini-batch normalized gradient descent with momentum that can find a first-order ϵ stationary point with an oracle complexity of $\mathcal{O}(1/\epsilon^4)$ for KL-regularized DRO and χ^2 regularized DRO with a non-convex loss. They solve the challenge that the loss function could be unbounded. Qi et al. (2021) proposed online stochastic compositional algorithms to solve KL-regularized DRO. They leveraged a recursive variance reduction technique (STORM (Cutkosky & Orabona, 2019)) to compute a gradient estimator for the model parameter $\mathbf{w}$ only. They derived a complexity of $\widetilde{\mathcal{O}}(1/\epsilon^3)$ for a general non-convex problem and improved it to $\mathcal{O}(1/(\mu\epsilon))$ for a problem that satisfies an μ-PL condition. Qi et al. (2020a) reports a worse complexity for a simpler algorithm for solving KL-regularized DRO. Li et al. (2020; 2021) studied the effectiveness of KL regularized objective on different applications, such as enforcing fairness between subgroups, and handling the class imbalance.

Compositional Functions and DRO. The connection between compositional functions and DRO formulations have been observed and leveraged in the literature. Dentcheva et al. (2017) studied the statistical estimation of compositional functionals with applications to estimating conditional-value-at-risk measures, which is closely related to the CVaR constrained DRO. However, they do not consider stochastic optimization algorithms. To the best of our knowledge, Qi et al. (2021) was the first to use stochastic compositional optimization algorithms to solve KL-regularized DRO problems. Our work is different in that we solve KL-constrained DRO problems, which is more challenging than KL-regularized DRO problems. The benefits of using compositional optimization for solving DRO include (i) we do not need to maintain and update a high dimensional dual variable as in the primal-dual methods (Rafique et al., 2021); (i) we do not need to worry about the batch size as in MLMC-based stochastic methods (Levy et al., 2020; Hu et al., 2021).

3 Preliminaries

In this section, we introduce notations, definitions and assumptions. We show that (1) is equivalent to (2) in Section F in Appendix. **Notations:** Let $\|\cdot\|$ denotes the Euclidean norm of a vector or the spectral norm of a matrix. And $\mathbf{x} = (\mathbf{w}^\top, \lambda)^\top \in \mathbb{R}^{d+1}$, $g_i(\mathbf{x}) = \exp(\frac{\ell_i(\mathbf{w})}{\lambda})$ and $g(\mathbf{x}) = \mathbb{E}_{i\sim\mathcal{D}}[\exp(\frac{\ell_i(\mathbf{w})}{\lambda})]$ where $\mathcal{D}$ denotes the training set and i denotes the index of the sample randomly generated from $\mathcal{D}$. Let $f_\lambda(\cdot) = \lambda\log(\cdot) + \lambda\rho$, and $\nabla f_\lambda(g) = \frac{\lambda}{g}$ denotes the gradient of f in terms of g. Let $\Pi_{\mathcal{X}}(\cdot)$ denote an Euclidean projection onto the domain $\mathcal{X}$. Let $[T] = \{1, \dots, T\}$ and $\tau \sim [T]$ denotes a random selected index. We make the following standard assumptions regarding to the problem (2).

Assumption 1. *There exists R, G, C, and L such that*

(a) The domain of model parameter $\mathcal{W}$ is bounded by R, i.e., for all $\mathbf{w} \in \mathcal{W}$, we have $\|\mathbf{w}\| \le R$.

(b) $\ell_i(\mathbf{w})$ is L-smooth, i.e., $\|\nabla\ell_i(\mathbf{w}_1) - \nabla\ell_i(\mathbf{w}_2)\| \le L\|\mathbf{w}_1 - \mathbf{w}_2\|$, $\forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}, i \sim \mathcal{D}$.

(c) $\ell_i(\mathbf{w})$ is G-Lipschitz continuous function and bounded by C, i.e., $\|\nabla\ell_i(\mathbf{w})\| \le G$ and $|\ell_i(\mathbf{w})| \le C$ for all $\mathbf{w} \in \mathcal{W}$ and $i \sim \mathcal{D}$.

(d) There exists a positive constant $\Delta < \infty$ and an initial solution $(\mathbf{w}_1, \lambda_1)$ such that $F(\mathbf{w}_1, \lambda_1) - \min\limits_{\mathbf{w}\in\mathcal{W}} \min\limits_{\lambda\ge\lambda_0} F(\mathbf{w}, \lambda) \le \Delta$.

Assumption 2. *Let* σ_g, $\sigma_{\nabla g}$ *be positive constants and* $\sigma^2 = \max\{\sigma_g, \sigma_{\nabla g}\}$. *For* $i \sim \mathcal{D}$, *assume that* $\mathbb{E}[\|g_i(\mathbf{x}) - g(\mathbf{x})\|^2] \leq \sigma_g^2, \quad \mathbb{E}[\|\nabla g_i(\mathbf{x}) - \nabla g(\mathbf{x})\|^2] \leq \sigma_{\nabla_g}^2$.

Remark: Assumption 1 (a), i.e., the boundness condition of $\mathcal{W}$ is also assumed in Levy et al. (2020), which is mainly used for convex analysis. Assumption 1(b), (c), i.e., the Lipstchiz continuity and smoothness of loss function, and the variance bounds for g_i and its gradient in Assumption 2 can be derived from Assumption 1 (b), such that $\mathbb{E}[\|g_i(\mathbf{x}) - g(\mathbf{x})\|^2] \leq \mathbb{E}[\|g_i(\mathbf{x})\|^2] \leq \exp(\frac{2C}{\lambda_0})$, and $\mathbb{E}[\|\nabla g_i(\mathbf{x}) - \nabla g(\mathbf{x})\|^2] \leq \mathbb{E}[\|\nabla g_i(\mathbf{x})\|^2] \leq \exp(\frac{2C}{\lambda_0})(G^2 + \frac{C^2}{\lambda_0})^4$

However, $F(\mathbf{w}, \lambda)$ is not necessarily smooth in terms of $\mathbf{x} = (\mathbf{w}^\top, \lambda)^\top$ if λ is unbounded. To address this concern, we prove that optimal λ is indeed bounded.

Lemma 1. *The optimal solution of the dual variable* λ^* *to the problem (2) is upper bounded by* $\tilde{\lambda} = \lambda_0 + C/\rho$, *where* C *is the upper bound of the loss function and* ρ *is the constraint parameter.*

Thus, we could constrain the domain of λ in the DRO formulation (2) with the upper bound $\tilde{\lambda}$, and obtain the following equivalent formulation:

$$\min_{\mathbf{w}\in\mathcal{W}} \min_{\lambda_0 \leq \lambda \leq \tilde{\lambda}} \lambda \log\left(\frac{1}{n}\sum_{i=1}^{n} \exp\left(\frac{\ell_i(\mathbf{w})}{\lambda}\right)\right) + \lambda\rho. \tag{3}$$

The upper bound $\tilde{\lambda}$ guarantees the smoothness of $F(\mathbf{w}, \lambda)$ and the smoothness of $f_\lambda(\cdot)$, which are critical for the proposed algorithms to enjoy fast convergence rates.

Lemma 2. $F(\mathbf{w}, \lambda)$ *is* L_F*-smooth for any* $\mathbf{w} \in \mathcal{W}$ *and* $\lambda \in [\lambda_0, \tilde{\lambda}]$, *where* $L_F = \tilde{\lambda}L_g^2 + 2L_g + \tilde{\lambda}L_{\nabla_g} + 1 + \tilde{\lambda}$. L_g *and* L_{∇_g} *are constants independent of sample size* n *and explicitly derived in Lemma 7 .*

Below, we let $\mathcal{X} = \{\mathbf{x}|\mathbf{w} \in \mathcal{W}, \lambda_0 \leq \lambda \leq \tilde{\lambda}\}$, $\delta_\mathcal{X}(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{X}$, and $\delta_\mathcal{X}(\mathbf{x}) = \infty$ if $\mathbf{x} \notin \mathcal{X}$. The problem (3) is equivalent to :

$$\min_{\mathbf{x}\in\mathbb{R}^{d+1}} \bar{F}(\mathbf{x}) := F(\mathbf{x}) + \delta_\mathcal{X}(\mathbf{x}), \tag{4}$$

Since $\bar{F}$ is non-smooth, we define the regular subgradient as follows.

Definition 1 (Regular Subgradient). *Consider a function* $\Phi : \mathbb{R}^n \to \overline{\mathbb{R}}$ *and* $\Phi(\bar{\mathbf{x}})$ *is finite at a point* $\bar{\mathbf{x}}$. *For a vector* $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}$ *is a regular subgradient of* Φ *at* $\bar{\mathbf{x}}$, *written* $\mathbf{v} \in \hat{\partial}\Phi(\bar{\mathbf{x}})$, *if*

$$\liminf_{\mathbf{x}\to\bar{\mathbf{x}}} \frac{\Phi(\mathbf{x}) - \Phi(\bar{\mathbf{x}}) - \mathbf{v}^\top(\mathbf{x} - \bar{\mathbf{x}})}{\|\mathbf{x} - \bar{\mathbf{x}}\|} \geq 0.$$

Since $F(\mathbf{x})$ is differentiable, we use $\hat{\partial}\bar{F}(\mathbf{x}) = \nabla F(\mathbf{x}) + \hat{\partial}\delta_\mathcal{X}(\mathbf{x})$ (see Exercise 8.8 in Rockafellar & Wets (1998)) in the analysis. Recall the definition of subgradient of a convex function $\bar{F}$ which is denoted by $\partial\bar{F}$. When $\bar{F}(\mathbf{x})$ is convex, we have $\hat{\partial}\bar{F}(\mathbf{x}) = \partial\bar{F}(\mathbf{x})$ (see Proposition 8.2 in Rockafellar & Wets (1998)). The $\mathrm{dist}(0, \hat{\partial}\bar{F}(\mathbf{x}))$ measures the distance between the origin and the regular subgradient set of $\bar{F}$ at $\mathbf{x}$. The oracle complexity is defined below:

Definition 2 (Oracle Complexity). *Let* $\epsilon > 0$ *be a small constant, the oracle complexity is defined as the number of processing samples* $\mathbf{z}$ *in order to achieve* $\mathbb{E}[\mathrm{dist}(0, \hat{\partial}\bar{F}(\mathbf{x}))] \leq \epsilon$ *for a non-convex loss function or* $\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}_*)] \leq \epsilon$ *for a convex loss function.*

---

[4] We would like to point out that the variance bound and the smoothness constant L_F are exponentially dependent on the problem parameters, so are these constants in some other stochastic methods solving constrained DRO, like Dual SGM in Levy et al. (2020).

4 Stochastic Constrained DRO with Non-convex Losses

In this section, we present two stochastic algorithms for solving (4). The first algorithm is simpler yet practical for deep learning applications. The second algorithm is an accelerated one with a better complexity, which is more complex than the first algorithm.

Algorithm 1 SCDRO$(\mathbf{x}_1, \mathbf{v}_1, u_1, s_1, \eta_1, T_1)$

```
Input: $\mathbf{w}_1 \in \mathcal{W}, \lambda_1 \geq \lambda_0, \mathbf{x}_1 = (\mathbf{w}_1^\top, \lambda_1)^\top$
Initialization: Draw a sample $\xi_1 \sim \mathcal{D}$, and calculate $s_1 = \exp(\ell_i(\mathbf{w}_1)/\lambda_1)$,
    $\mathbf{v}_1 = \nabla f_{\lambda_1}(s_1)\nabla_{\mathbf{w}} g_i(\mathbf{x}_1)) \in \mathbb{R}^d$
    $u_1 = \nabla f_{\lambda_1}(s_1)\nabla_{\lambda} g_i(\mathbf{x}_1) + \log(s_1) + \rho \in \mathbb{R}$
for $t = 1, \cdots, T$ do
    Update $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta \mathbf{z}_t)$
    Draw a sample $\xi_i \sim \mathcal{D}$
    Let $s_{t+1} = (1-\beta)s_t + \beta g_i(\mathbf{x}_{t+1})$
    Update $\mathbf{v}_{t+1}, u_{t+1}$ according to (6)
end for
return: $(\mathbf{x}_\tau, \mathbf{v}_\tau, u_\tau, s_\tau)$, where $\tau \sim [T]$
```

Algorithm 2 ASCDRO$(\mathbf{x}_1, \mathbf{v}_1, u_1, s_1, \eta_1, T_1)$

```
Input: $\mathbf{w}_1 \in \mathcal{W}, \lambda_1 \geq \lambda_0$, $\mathbf{x}_1 = (\mathbf{w}_1^\top, \lambda_1)^\top$
Initialization: Draw a sample $\xi_1 \sim \mathcal{D}$, and calculate $s_1 = \exp(\ell_i(\mathbf{w}_1)/\lambda_1)$,
    $\mathbf{v}_1 = \nabla_{\mathbf{w}} g_i(\mathbf{x}_1) \in \mathbb{R}^d$
    $u_1 = \nabla_{\lambda} g_i(\mathbf{x}_1) \in \mathbb{R}$
for $t = 1, \cdots, T$ do
    Update $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta \mathbf{z}_t)$, where $\mathbf{z}_t$ is given in (8)
    Draw a sample $\xi_i \sim \mathcal{D}$
    Update $s_{t+1}, \mathbf{v}_{t+1}, u_{t+1}$ according to (7)
end for
return: $(\mathbf{x}_\tau, \mathbf{v}_\tau, u_\tau, s_\tau)$, where $\tau \sim [T]$
```

4.1 Basic Algorithm: SCDRO

A major concern of the algorithm design is to compute a stochastic gradient estimator of the gradient of $F(\mathbf{x})$. At iteration t, the gradient of $F(\mathbf{x}_t)$ is given by

$$\begin{aligned} \nabla_{\mathbf{w}} F(\mathbf{x}_t) &= \nabla f_{\lambda_t}(g(\mathbf{x}_t))\nabla_{\mathbf{w}} g(\mathbf{x}_t) \\ \nabla_{\lambda} F(\mathbf{x}_t) &= \nabla f_{\lambda_t}(g(\mathbf{x}_t))\nabla_{\lambda} g(\mathbf{x}_t) + \log(g(\mathbf{x}_t)) + \rho. \end{aligned} \tag{5}$$

Both $\nabla_\lambda g(\mathbf{x}_t)$ and $\nabla_{\mathbf{w}} g(\mathbf{x}_t)$ can be estimated by unbiased estimator denoted by $\nabla g_i(\mathbf{x}_t)$. The concern lies at how to estimate $g(\mathbf{x}_t)$ inside $\nabla f_{\lambda_t}(\cdot)$. The first algorithm SCDRO is applying existing techniques for two-level compositional function. In particular, we estimate $g(\mathbf{x}_t)$ by a sequence of s_t, which is updated by moving average $s_t = (1-\beta)s_{t-1} + \beta g_i(\mathbf{x}_t)$. Then we substitute $g(\mathbf{x}_t)$ in $\nabla_{\mathbf{w}} F(\mathbf{x}_t)$ and $\nabla_\lambda F(\mathbf{x}_t)$ with s_t, and invoke the following moving average to obtain the gradient estimators in terms of $\mathbf{w}_t$ and λ_t, respectively,

$$\begin{aligned} \mathbf{v}_t &= (1-\beta)\mathbf{v}_{t-1} + \beta \nabla f_{\lambda_t}(s_t)\nabla_{\mathbf{w}} g_i(\mathbf{x}_t) \\ u_t &= (1-\beta)u_{t-1} + \beta(\nabla f_{\lambda_t}(s_t)\nabla_\lambda g_i(\mathbf{x}_t) + \log(s_t) + \rho). \end{aligned} \tag{6}$$

Finally we complete the update step of $\mathbf{x}_t$ by $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta \mathbf{z}_t)$, where $\mathbf{z}_t = (\mathbf{v}_t^\top, u_t)^\top$.

We would like to point out the moving average estimator for tracking the inner function $g(\mathbf{w})$ is widely used for solving compositional optimization problems (Wang et al., 2017; Qi et al., 2021; Zhang & Xiao, 2019; Zhou et al., 2019). Using the moving average for computing a stochastic gradient estimator of a compositional function was first used in the NASA method proposed in Ghadimi et al. (2020). The proposed method SCDRO is presented in Algorithm 1. It is similar to NASA but with a simpler design on the update of $\mathbf{x}_{t+1}$. We directly use projection after an SGD-tyle update. In contrast, NASA uses two steps to update $\mathbf{x}_{t+1}$. As a consequence, NASA has two parameters for updating $\mathbf{x}_{t+1}$ while SCDRO only has one parameter η for updating $\mathbf{x}_{t+1}$. It is this simple change that allows us to extend SCDRO for convex problems in the next section. Below, we present the convergence rate of our basic algorithm SCDRO for a non-convex loss function.

Theorem 1. *Suppose the Assumption 1 and 2 hold, and set* $\beta = \frac{1}{\sqrt{T}}, \eta = \frac{\beta}{20 L_F^2}$. *Then after running Algorithm 1 T iterations, we have* $\mathbb{E}[\mathrm{dist}(0, \hat{\partial}\bar{F}(\mathbf{x}_\tau))^2] \leq (624\sigma^2 + 280\Delta)\frac{L_F^2}{\sqrt{T}} + \frac{20 L_F^2 \Delta}{T}$.

Remark: Theorem 1 shows that SCDRO achieves a complexity of $\mathcal{O}(1/\epsilon^4)$ for finding an ϵ-stationary point, i.e., $\mathbb{E}[\mathrm{dist}(0, \hat{\partial}\bar{F}(\mathbf{x}_R))] \leq \epsilon$ for a non-convex loss function. Note that NASA (Ghadimi et al., 2020) enjoys the same oracle complexity but for a different convergence measure, i.e., $\mathbb{E}[\|\mathbf{y}(\mathbf{x},\mathbf{z})-\mathbf{x}\|^2+\|\mathbf{z}-\nabla F(\mathbf{x})\|^2]\leq\epsilon$ for a returned primal-dual pair $(\mathbf{x},\mathbf{z})$, where $\mathbf{y}(\mathbf{x},\mathbf{z})=\prod_{\mathcal{X}}[\mathbf{x}-\mathbf{z}]$. We can see that our convergence measure is more intuitive. In addition, we are able to leverage our convergence measure to establish the convergence for convex functions by using Kurdyka-Łojasiewicz (KL) inequality and the restarting trick as shown in next section. In contrast, such convergence for NASA is missing in their paper. Compared with stochastic primal-dual methods (Rafique et al., 2021; Yan et al., 2020) for the min-max formulation (1), their algorithms are double looped and have the same oracle complexity for a different convergence measure, i.e., $\mathbb{E}[\text{dist}(0,\hat{\partial}\bar{F}(\mathbf{x}_*))^2]\leq\gamma^2\|\mathbf{x}-\mathbf{x}_*\|^2]\leq\epsilon$ for some returned solution $\mathbf{x}$, where $\mathbf{x}_*$ is a reference point that is not computable. Our convergence measure is stronger as we directly measure $\mathbb{E}[\text{dist}(0,\hat{\partial}\bar{F}(\mathbf{x}_\tau))^2]$ on a returned solution $\mathbf{x}_\tau$. This is due to that we leverage the smoothness of $F(\cdot)$.

4.2 Accelerated Algorithm: ASCDRO

Our second algorithm presented in Algorithm 2 is inspired by Qi et al. (2021) for solving the KL-regularized DRO by leveraging a recursive variance reduced technique (i.e., STORM) to estimate $g(\mathbf{w}_t)$ and $\nabla g(\mathbf{w}_t)$ for computing $\nabla_{\mathbf{w}}F(\mathbf{x}_t)$ and $\nabla_\lambda F(\mathbf{x}_t)$ in (5). In particular, we use $\mathbf{v}_t$ for tracking $\nabla_{\mathbf{w}}g(\mathbf{x}_t)$, use u_t for tracking $\nabla_\lambda g(\mathbf{x}_t)$, and use s_t for tracking $g(\mathbf{x}_t)$, which are updated by:

$$\begin{aligned}\mathbf{v}_t&=\nabla_{\mathbf{w}}g_i(\mathbf{x}_t)+(1-\beta)(\mathbf{v}_{t-1}-\nabla_{\mathbf{w}}g_i(\mathbf{x}_{t-1}))\\ u_t&=\nabla_\lambda g_i(\mathbf{x}_t)+(1-\beta)(u_{t-1}-\nabla_\lambda g_i(\mathbf{x}_{t-1}))\\ s_t&=g_i(\mathbf{x}_t)+(1-\beta)(s_{t-1}-g_i(\mathbf{x}_{t-1})).\end{aligned}\tag{7}$$

A similar update to s_t has been used in Chen et al. (2021) for tracking the inner function values for two-level compositional optimization. However, they do not use similar updates for tracking the gradients as $\mathbf{v}_t, u_t$. Hence, their algorithm has a worse complexity.

Then we invoke these estimators into $\nabla_{\mathbf{w}}F(\mathbf{x}_t)$ and $\nabla_\lambda F(\mathbf{x}_t)$ to obtain the gradient estimator

$$\mathbf{z}_t=(\nabla f_{\lambda_t}(s_t)\mathbf{v}_t^\top,\nabla f_{\lambda_t}(s_t)u_t+\log(s_t)+\rho)^\top.\tag{8}$$

Below, we show ASCDRO can achieve a better convergence rate in the non-convex loss function.

Theorem 2. *Under Assumption 1 and 2, for any* $\alpha>1$, *let* $k=\frac{\alpha\sigma^{2/3}}{L_F}$, $w=\max(2\sigma^2,(16L_F^2k)^3)$ *and* $c=\frac{\sigma^2}{14L_Fk^3}+130L_F^4$. *Then after running Algorithm 2 for* T *iterations with* $\eta_t=\frac{k}{(w+t\sigma^2)^{1/3}}$ *and* $\beta_t=c\eta_t^2$, *we have* $\mathbb{E}[\text{dist}(0,\hat{\partial}\bar{F}(\mathbf{x}_\tau))^2]\leq\mathcal{O}\left(\frac{\log T}{T^{2/3}}\right)$.

Remark: Theorem 2 implies that with a polynomial decreasing step size, ASCDRO is able to find an ϵ-stationary solution such that $\mathbb{E}[\text{dist}(0,\hat{\partial}\bar{F}(\mathbf{x}_R))]\leq\epsilon$ with a near-optimal complexity $\widetilde{\mathcal{O}}(1/\epsilon^3)$. Note that the complexity $\widetilde{\mathcal{O}}(1/\epsilon^3)$ is optimal up to a logarithmic factor for solving non-convex smooth optimization problems (Arjevani et al., 2019). State-of-the-art primal-dual methods with variance-reduction for min-max problems (Huang et al., 2020) have the same complexity but for a different convergence measure, i.e, $\mathbb{E}[\frac{1}{\gamma}\|\mathbf{x}-\prod_{\mathcal{X}}[\mathbf{x}-\gamma\nabla F(\mathbf{x})]\|]\leq\epsilon$ for a returned solution $\mathbf{x}$.

5 Stochastic Algorithms for Convex Problems

In this section, we presented restarted algorithms for solving (3) with a convex loss function $\ell_i(\mathbf{w})$. The key is to restart SCDRO and ASCDRO by using a stagewise step size scheme. We define a new objective $F_\mu(\mathbf{x})=F(\mathbf{x})+\mu\|\mathbf{x}\|^2/2$ and correspondingly $\bar{F}_\mu(\mathbf{x})=F_\mu(\mathbf{x})+\delta_{\mathcal{X}}(\mathbf{x})$, where μ is a constant to be determined later. With this new objective, we have the following lemma.

Lemma 3. *Suppose that* $\ell_i(\mathbf{w})$ *is convex for all* i, *then for all* $\mathbf{x}\in\mathcal{X}$, $\bar{F}_\mu(\mathbf{x})$ *satisfies the following Kurdyka-Łojasiewicz (KL) inequality* $\text{dist}(0,\partial\bar{F}_\mu(\mathbf{x}))^2\geq 2\mu(\bar{F}_\mu(\mathbf{x})-\inf_{\mathbf{x}\in\mathcal{X}}\bar{F}_\mu(\mathbf{x}))$.

Lemma 3 allows us to obtain the convergence guarantee for convex losses. The idea of the restarted algorithm is to apply SCDRO and ASCDRO to the new objective $\bar{F}_\mu(\mathbf{x})$ by adding $\mu\mathbf{x}_t$ to $(\nabla f_{\lambda_t}(s_t)\nabla_{\mathbf{w}}g_i(\mathbf{x}_t)^\top,\nabla f_{\lambda_t}(s_t)\nabla_\lambda g_i(\mathbf{x}_t)+\log(s_t)+\rho)^\top$ in Eq. (6) of Algorithm 1 and substituting $\mathbf{z}_t$ in (8) of Algorithm 2 by $\mathbf{z}_t=(\nabla f_{\lambda_t}(s_t)\mathbf{v}_t^\top,\nabla f_{\lambda_t}(s_t)u_t+\log(s_t)+\rho)^\top+\mu\mathbf{x}_t$, and restarting SCDRO or ASCDRO with

Algorithm 3 RSCDRO or RASCDRO

```
Input: $\mathbf{w}_1 \in \mathcal{W}, \lambda_1 \in \mathbb{R}^+$, $\mathbf{x}_1 = (\mathbf{w}_1^\top, \lambda_1)^\top$
Initialization: The same as in SCDRO or ASCDRO
Let $\Lambda_k = (\mathbf{x}_k, \mathbf{v}_k, u_k, s_k)$
for $k = 1, \cdots, K$ do
    $\Lambda_{k+1} = \text{SCDRO}(\Lambda_k, \eta_k, T_k)$ or $\Lambda_{k+1} = \text{ASCDRO}(\Lambda_k, \eta_k, T_k)$
    Change $\eta_k, T_k$ according to Lemma 4 or Lemma 5
end for
return: $\mathbf{x}_K$
```

a stagewise step size to enjoy the benefit of KL inequality of $\bar{F}_\mu(\mathbf{x})$. It is notable that a stagewise step size is widely and commonly used in practice. The multi-stage restarted version of SCDRO and ASCDRO are shown Algorithm 3, to which we refer as restarted-SCDRO (RSCDRO) and restarted-ASCDRO (RASCDRO).

5.1 Restarted SCDRO for Convex Problems

In this subsection, we present the convergence rate of RSCDRO for convex losses. We first present a lemma that states $F_\mu(\mathbf{x}_k)$ is stagewisely decreasing.

Lemma 4. *Suppose Assumptions 1 and 2 hold, $\ell_i(\mathbf{w})$ is convex for all i, and $F_\mu(\mathbf{x}_1) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x}) \leq \Delta_\mu < \infty$. Let $\epsilon_1 = \Delta_\mu$, $\epsilon_k = \epsilon_{k-1}/2$, $\beta_k = \min\{\frac{\mu\epsilon_k}{c\sigma^2}, \frac{1}{c}\}, \eta_k = \min\{\frac{\mu\epsilon_k}{12cL_F^2\sigma^2}, \frac{1}{12cL_F^2}\}$ and $T_k = \max\{\frac{384cL_F^2\sigma^2}{\mu^2\epsilon_k}, \frac{384cL_F^2}{\mu}\}$, where $c = 384L_F^2$. Run RSCDRO, then we have $\mathbb{E}[F_\mu(\mathbf{x}_k) - \inf\limits_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})] \leq \epsilon_k$ for each stage k.*

The above lemma implies that the objective gap $\mathbb{E}[F_\mu(\mathbf{x}_k) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})]$ is decreased by a factor of 2 after each stage. Based on the above lemma, RSCDRO has the following convergence rate

Theorem 3. *Under the same assumptions and parameter settings as Lemma 4, after $K = \mathcal{O}(\log_2(\epsilon_1/\epsilon))$ stages, the output of RSCDRO satisfies $\mathbb{E}[F_\mu(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})] \leq \epsilon$, and the oracle complexity is $\mathcal{O}(1/\mu^2\epsilon)$.*

As $F_\mu(\mathbf{x}_K) - F_\mu(\mathbf{x}_*) \leq F_\mu(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})$, where $\mathbf{x}_* = \arg\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x})$. Therefore, if after K stages it holds that $\mathbb{E}[F_\mu(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})] \leq \epsilon/2$ with an oracle complexity of $\mathcal{O}(1/\mu^2\epsilon)$, we have $\mathbb{E}[F_\mu(\mathbf{x}_K) - F_\mu(\mathbf{x}_*)] \leq \epsilon/2$, i.e., $\mathbb{E}[F(\mathbf{x}_K) + \mu\|\mathbf{x}_K\|^2/2 - F(\mathbf{x}_*) - \mu\|\mathbf{x}_*\|^2/2] \leq \epsilon/2$. By Assumption 1(a) $\mathcal{W}$ is bounded by R, and then by setting $\mu = \epsilon/(2(R^2 + \tilde{\lambda}^2))$, with $\|\mathbf{x}\|^2 \leq (R^2 + \tilde{\lambda}^2)$ we have

$$\mathbb{E}[F(\mathbf{x}_K) - F(\mathbf{x}_*)] \leq \frac{\epsilon}{2} + (2(R^2 + \tilde{\lambda}^2))\frac{\mu}{2} \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \leq \epsilon$$

with an oracle complexity of $\mathcal{O}(1/\epsilon^3)$, i.e, the following coroellary holds.

Corollary 1. *Let $\mu = \epsilon/(2(R^2 + \tilde{\lambda}^2))$. Then under the same assumptions and parameter settings as Lemma 4, after $K = \mathcal{O}(\log_2(\epsilon_1/\epsilon))$ stages, the output of RSCDRO satisfies $\mathbb{E}[F(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x})] \leq \epsilon$ and the oracle complexity is $\mathcal{O}(1/\epsilon^3)$.*

Remark: Corollary 1 shows that RSCDRO achieves an oracle complexity of $\mathcal{O}(1/\epsilon^3)$ for finding an ϵ-optimal solution. i.e., $\mathbb{E}[F(\mathbf{x}) - F(\mathbf{x}_*)] \leq \epsilon$ for the convex loss function with a geometrically decreasing step size in a stagewise manner.

5.2 Restarted ASCDRO for Convex Problems

In this subsection, we establish a better convergence rate of RASCDRO for convex losses.

Lemma 5. *Suppose Assumptions 1 and 2 hold, $\ell_i(\mathbf{w})$ is convex for all i, and $F_\mu(\mathbf{x}_1) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x}) \leq \Delta_\mu < \infty$. Let $\epsilon_1 = \Delta_\mu$, $\epsilon_k = \epsilon_{k-1}/2$, $\beta_k = \min\{\frac{\mu\epsilon_k}{c\sigma^2}, \frac{1}{c}\}, \eta_k = \min\{\frac{\sqrt{\mu\epsilon_k}}{24cL_F\sigma^2}, \frac{1}{24cL_F^2}\}$ and $T_k = \max\{\frac{192cL_F\sigma}{\mu^{3/2}\sqrt{\epsilon_k}}, \frac{192cL_F^2\sigma^2}{\mu\epsilon_k}, \frac{192cL_F^2}{\mu}\}$, where $c = 768L_F^2$. Run RASCDRO, then we have $\mathbb{E}[F_\mu(\mathbf{x}_k) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})] \leq \epsilon_k$ for each stage k.*

The above lemma implies that the objective gap $\mathbb{E}[F_\mu(\mathbf{x}_k) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})]$ is decreased by a factor of 2 after each stage. Hence we have the following convergence rate for the RASCDRO.

Theorem 4. *Under the same assumptions and parameter settings as Lemma 5, after* $K = \mathcal{O}(\log_2(\epsilon_1/\epsilon))$ *stages, the output of RASCDRO satisfies* $\mathbb{E}[F_\mu(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F_\mu(\mathbf{x})] \leq \epsilon$, *and the oracle complexity is* $\mathcal{O}\left(\max\left(1/\mu\epsilon, 1/\mu^{3/2}\sqrt{\epsilon}\right)\right)$.

By the same method of derivation of Corollary 1, the following corollary of Theorem 4 holds.

Corollary 2. *Let* $\mu = \epsilon/(2(R^2 + \tilde{\lambda}^2))$. *Then under the same assumptions and parameter settings as Lemma 5, after* $K = \mathcal{O}(\log_2(\epsilon_1/\epsilon))$ *stages, the output of RASCDRO satisfies* $\mathbb{E}[F(\mathbf{x}_K) - \inf_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x})] \leq \epsilon$ *and the oracle complexity is* $\mathcal{O}(1/\epsilon^2)$.

Remark: Corollary 2 shows that RASCDRO achieves the claimed oracle complexity $\mathcal{O}(1/\epsilon^2)$ for finding an ϵ-optimal solution, which is optimal for solving convex smooth optimization problems (Nemirovsky & Yudin, 1983). Finally, we note that a similar complexity was established in (Zhang & Lan, 2020) for constrained convex compositional optimization problems. However, their analysis requires each level function to be convex, which does not apply to our case as the outer function $f_\lambda(\cdot)$ is non-convex.

6 Experiments

In this section, we verify the effectiveness of the proposed algorithms in solving imbalanced classification problems. We show that the proposed methods outperform baselines under both the convex and non-convex settings in terms of convergence speed, and generalization performance. In addition, we study the influence of ρ to the robustness of different optimization methods in supplement.

Baselines. For the comparison of convergence speed, we compare with different algorithms for optimizing the same objective (1), including, stochastic primal-dual algorithms, namely PG-SMD2 (Rafique et al., 2021) for a non-convex loss, and SPD (Namkoong & Duchi, 2016) for a convex loss, Dual SGM (Levy et al., 2020) and mini-batch based SGD named FastDRO (Levy et al., 2020) for both convex and non-convex losses . For the comparison of generalization performance, we compare with different methods for optimizing different objectives, including the traditional ERM with CE loss by SGD with momentum (SGDM), KL-regularized DRO solved by RECOVER (Qi et al., 2021), and CVaR-constrained, χ^2-regularized/-constrained DRO optimized by FastDRO.

Datasets. We conduct experiments on four imbalanced datasets, namely CIFAR10-ST, CIFAR100-ST (Qi et al., 2020b), ImageNet-LT (Liu et al., 2019), and iNaturalist2018 (iNaturalist 2018 competition dataset). The original CIFAR10, CIFAR100 are balanced data, where CIFAR10 (resp. CIFAR100) has 10 (resp. 100) classes and each class has 5K (resp. 500) training images. For constructing CIFAR10-ST and CIFAR100-ST, we artificially construct imbalanced training data, where we only keep the last 100 images of each class for the first half classes, and keep other classes and the test data unchanged. ImageNet-LT is a long-tailed subset of the original ImageNet-2012 by sampling a subset following the Pareto distribution with the power value 6. It has 115.8K images from 1000 categories, which include 4980 for head class and 5 images for tail class. iNaturalist 2018 is a real-world dataset whose class-frequency follows a heavy-tail distribution. It contains 437K images from 8142 classes.

Models. For a non-convex setting (deep model), we learn ResNet20 for CIFAR10-ST, CIFAR100-ST, and ResNet50 for ImageNet-LT and iNaturalist2018, respectively. On CIFAR10-ST, CIFAR100-ST, we optimize the network from scratch by different algorithms. For the large-scale ImageNet-LT and iNaturalist2018 datasets, we optimize the last block of the feature layers and the classifier weight with other layers frozen of a pretrained ResNet50 model. This is a common training strategy in the literature (Kang et al., 2019; Qi et al., 2020a). For a convex setting (linear model), we freeze the feature layers of the pretrained models, and only fine-tune the last classifier weight. The pretrained models for ImageNet-LT, CIFAR10-ST, CIFAR100-ST are trained from scratch by optimizing the standard cross-entropy (CE) loss using SGD with momentum 0.9 for 90 epochs. The pretrained ResNet50 model for iNaturalist2018 is from the released model by Kang et al. (2019).

Parameters and Settings. For all experiments, the batch size is 128 for CIFAR10-ST and CIFAR100-ST, and 512 for ImageNet-LT and iNaturalist2018. The loss function is the CE loss. The λ_0 is set to $1e$-3. The (primal) learning rates for all methods are tuned in $\{0.01, 0.05, 0.1, 0.5, 1\}$. The learning rate for updating the

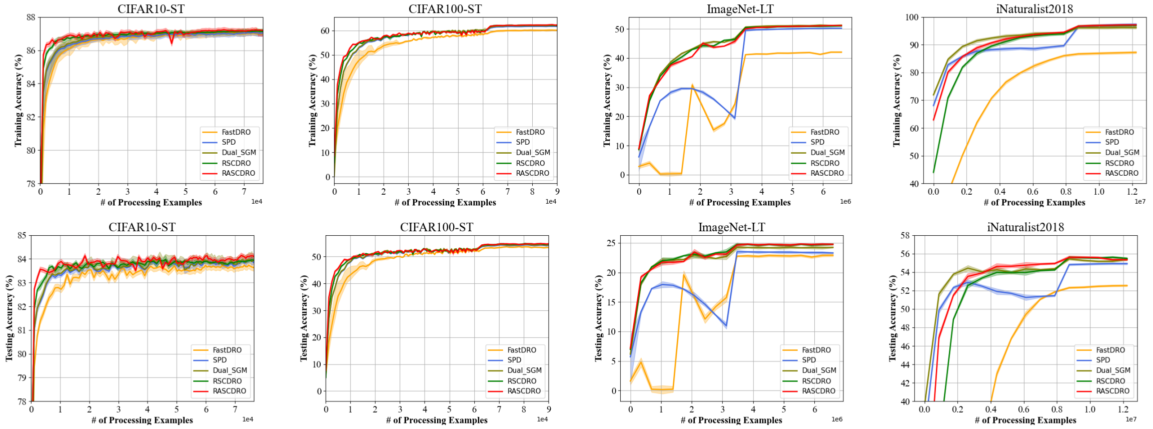

Figure 1: Training accuracy (%) , Testing accuracy (%) vs # of processed training samples for the convex setting. ρ is fixed to 0.5 on CIFAR10-ST and CIFAR100-ST, and 0.1 on ImageNet-LT and iNaturalist2018. The results are averaged over 5 independent runs.

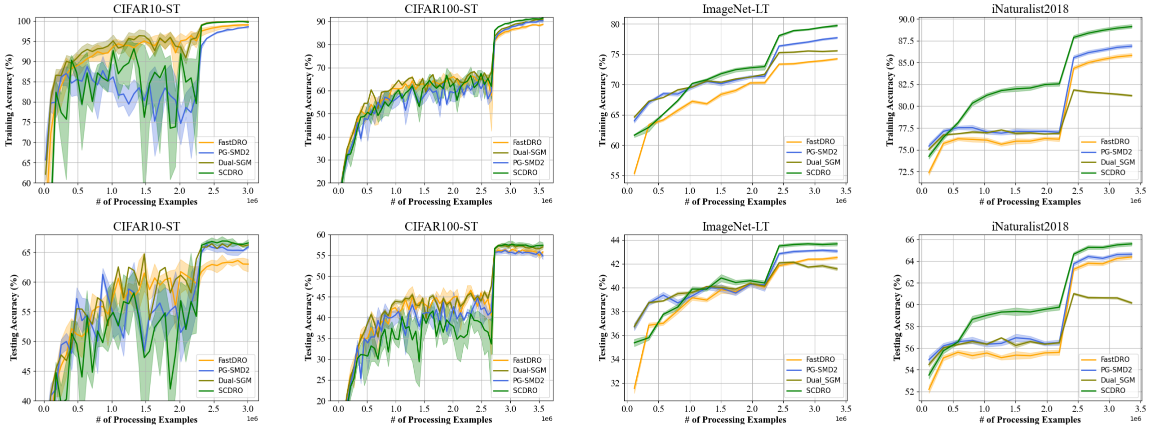

Figure 2: Training accuracy (%), Testing accuracy (%) vs # of processed training samples for the non-convex setting. ρ is fixed to 0.5 on all datasets. The results are averaged over 5 independent runs.

dual variable in PG_SMD2 and SPD is tuned in $\{1e\text{-}5, 5e\text{-}5, 1e\text{-}4, 5e\text{-}4)\}$. The momentum parameter β in our proposed algorithms and RECOVER are tuned $\{0.1 : 0.1 : 0.9\}$. For RECOVER, the hyper-parameter λ is tuned in $\{1, 50, 100\}$. The constrained parameter ρ is tuned in $\{0.1, 0.5, 1\}$ for the comparison of generalization performance unless specified otherwise. The initial λ and Larange multiplier in Dual SGM are both tuned in $\{0.1, 1, 10\}$. All our results are conducted on Tesla V100.

Convergence comparison between different baselines. In the convex setting, we compare RSCDRO and RASCDRO with SPD, FastDRO and Dual SGM baselines. We report the training accuracy and testing accuracy in terms of the number (#) of processing samples. We denote 1 pass of training data by 1 epoch. We run a total of 3 epochs for CIFAR10-ST and CIFAR100-ST and decay the learning rate by a factor of 10 at the end of 2nd epoch. Similarly, we run 60 epochs and decay the learning rate at the 30th epochs for the ImageNet-LT, and run 30 epochs and decay the learning rate at the 20th epoch for iNaturalist2018. In the nonconvex setting, we compare SCDRO with two baselines, PG-SMD2 and FastDRO. We run 120 epochs for CIFAR10-ST and CIFAR100-ST, and decay the learning rate by a factor of 10 at the 90th epoch. And we run 30 epochs for ImageNet-LT and iNaturalist2018, and decay the learning rate at the 20th epoch.

Results. We first report the results for convex setting in Figures 1. It is obvious to see that RSCDRO and RASCDRO are consistently better than baselines on CIFAR10-ST, CIFAR100-ST, and ImageNet-LT. PD-SMD2 and Dual SGM have comparable results with our proposed algorithms on the iNaturalist2018 in terms of training accuracy, but is worse in terms of testing accuracy. FastDRO has the worst performance on all the datasets. RSCDRO and RASCDRO achieve comparable results on all datasets, however, the stochastic estimator in RASCDRO requires two gradient computations per iteration, which incurs more computational cost than RSCDRO. Hence, in the non-convex setting, we focus on SCDRO. Figure 2 reports the results for non-convex setting. We can see that SCDRO achieves the best performance on all the datasets. The margin increases on the large scale ImageNet-LT and iNaturalist2018 datasets. For the three baselines, Dual SGM has better testing performance than FastDRO and PD-SGM2 on CIFAR10-ST and CIFAR100-ST. On the large scale data ImageNet-LT and iNaturalist2018, however, Dual SGM has the worst performance in terms of the testing accuracy. Furthermore, SCDRO is more stable than FastDRO and Dual SGM in different settings as the training of Dual SGM and FastDRO is comparable to SCDRO in convex settings and much worse than SCDRO in non-convex settings.

Comparison with ERM and KL-regularized DRO. Next, we compare our method for solving KL-constrained DRO (KL-CDRO) with 1) ERM+SGDM, and KL-regularized DRO (KL-RDRO) optimized by RECOVER in the non-convex setting 2) CVaR-constrained DRO, χ^2-regularized DRO χ^2-constrained DRO optimized by FastDRO in the convex setting. We conduct the experiments on the large-scale ImageNet-LT and iNaturalist2018 datasets. The results shown in Table 2 and 3 vividly demonstrate that our method for constrained DRO outperforms the ERM-based method and other popular f-divergence constrained/regularized DRO in different settings.

Table 2: Testing Accuracy in Convex Setting

	ImageNet-LT	iNaturalist2018
KL-Constraint + SCDRO	**24.08** (± 0.01)	**55.63** (± 0.03)
CVaR-Constraint + FastDRO	17.23 (± 0.03)	54.52 (± 0.11)
χ^2-Regularization + FastDRO	23.98 (± 0.01)	55.03 (± 0.03)
χ^2-Constraint + FastDRO	23.61 (± 0.01)	53.71 (± 0.05)

Table 3: Testing Accuracy in Non-Convex Setting

	ImageNet-LT	iNaturalist2018
KL-Constraint + SCDRO	**43.74**	**65.59**
ERM+SGDM	43.36	64.42
KL-Regularization + RECOVER	42.68	64.57

Sensitivity to ρ. We study the sensitivity of different methods to ρ. The results on CIFAR10-ST and CIFAR100-ST are shown in Table 4 in the supplement, which demonstrates that the testing performance is sensitive to ρ. However, our method SCDRO is better than baselines PG-SMD2 and FastDRO for different values of ρ.

Table 4: Test accuracy (%) of different methods for different constraint parameter ρ in the non-convex setting. The results are averaged over 5 independent runs.

	ρ	0.01	0.05	0.1	0.5	1
CIFAR10-ST	PG-SMD2	67.09 (± 0.59)	66.96 (± 0.71)	67.12 (± 0.61)	67.36 (± 0.36)	67.10 (± 0.61)
	FastDRO	65.41 (± 0.33)	66.15 (± 0.09)	66.24 (± 0.63)	65.98 (± 0.45)	65.68 (± 0.52)
	SCDRO	**67.73** (± 0.39)	**67.58** (± 0.48)	**67.71** (± 0.43)	**67.57** (± 0.28)	**67.96** (± 0.50)
CIFAR100-ST	PG-SMD2	57.31 (± 0.09)	56.44 (± 0.17)	55.85 (± 0.19)	52.68 (± 0.40)	48.72 (± 0.25)
	FastDRO	57.60 (± 0.32)	57.20 (± 0.42)	56.78 (± 0.40)	55.58 (± 0.62)	52.39 (± 0.31)
	SCDRO	**57.84** (± 0.15)	**57.60** (± 0.15)	**58.32** (± 0.43)	**57.90** (± 0.26)	**57.71** (± 0.24)

7 Conclusions

In this paper, we proposed dual-free stochastic algorithms for solving KL-constrained distributionally robust optimization problems for both convex and non-convex losses. The proposed algorithms have nearly optimal complexity in both settings. Empirical studies vividly demonstrate the effectiveness of the proposed algorithm for solving non-convex and convex constrained DRO problems.

References

Ahmet Alacaoglu, Volkan Cevher, and Stephen J Wright. On the complexity of a practical primal-dual coordinate method. *arXiv preprint arXiv:2201.07684*, 2022.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

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.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ruidi Chen and Ioannis C Paschalidis. A robust learning approach for regression models based on distributionally robust optimization. *Journal of Machine Learning Research*, 19(13), 2018.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. *IEEE Transactions on Signal Processing*, 69:4937–4948, 2021. doi: 10.1109/tsp.2021.3092377. URL `https://doi.org/10.1109%2Ftsp.2021.3092377`.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in Neural Information Processing Systems*, 32:15236–15245, 2019.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33, 2020.

Darinka Dentcheva, Spiridon Penev, and Andrzej Ruszczynski. Statistical estimation of composite risk functionals and risk optimization problems. *Annals of the Institute of Statistical Mathematics*, 69(4):737–760, 2017. URL `https://EconPapers.repec.org/RePEc:spr:aistmt:v:69:y:2017:i:4:d:10.1007_s10463-016-0559-8`.

C. John Duchi, W. Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 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.

Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

Yifan Hu, Xin Chen, and Niao He. On the bias-variance-cost tradeoff of stochastic optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22119–22131. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/b986700c627db479a4d9460b75de7222-Paper.pdf`.

Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv e-prints*, pp. arXiv–2008, 2020.

iNaturalist 2018 competition dataset. iNaturalist 2018 competition dataset. `https://github.com/visipedia/inat_comp/tree/master/2018`, 2018.

Jikai Jin, Bohang Zhang, Haiyang Wang, and Liwei Wang. Non-convex distributionally robust optimization: Non-asymptotic analysis. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. In *International Conference on Learning Representations*, 2020.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. On tilted losses in machine learning: Theory and applications. *arXiv preprint arXiv:2109.06141*, 2021.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. In *Advances in neural information processing systems*, pp. 2971–2980, 2017.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

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.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983. ISBN 9780471103455. URL `https://books.google.com/books?id=6ULvAAAAMAAJ`.

Qi Qi, Yi Xu, Rong Jin, Wotao Yin, and Tianbao Yang. Attentional biased stochastic gradient for imbalanced classification. *arXiv preprint arXiv:2012.06951*, 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.

Qi Qi, Zhishuai Guo, Yi Xu, Rong Jin, and Tianbao Yang. An online method for a class of distributionally robust optimization with non-convex objectives. *Advances in Neural Information Processing Systems*, 34, 2021.

Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, pp. 1–35, 2021.

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

RT Rockafellar and RJB Wets. Variational analysis springer. *MR1491362*, 1998.

Chaobing Song, Stephen J Wright, and Jelena Diakonikolas. Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums. In *International Conference on Machine Learning*, pp. 9824–9834. PMLR, 2021.

Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. *Advances in Neural Information Processing Systems*, 32:9134–9144, 2019.

Quoc Tran-Dinh, Deyi Liu, and Lam M Nguyen. Hybrid variance-reduced sgd algorithms for minimax problems with nonconvex-linear function. In *NeurIPS*, 2020.

Madeleine Udell, Karanveer Mohan, David Zeng, Jenny Hong, Steven Diamond, and Stephen Boyd. Convex optimization in julia. In *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*, pp. 18–28. IEEE, 2014.

Jie Wang, Rui Gao, and Yao Xie. Sinkhorn distributionally robust optimization. *arXiv preprint arXiv:2109.11926*, 2021.

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.

Yi Xu, Rong Jin, and Tianbao Yang. Non-asymptotic analysis of stochastic methods for non-smooth non-convex regularized problems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 2630–2640, 2019.

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

Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Optimal epoch stochastic gradient descent ascent methods for min-max optimization. In *Conference on Neural Information Processing Systems*, 2020.

Junyu Zhang and Lin Xiao. A stochastic composite gradient method with incremental variance reduction. In *Advances in Neural Information Processing Systems*, pp. 9075–9085, 2019.

Zhe Zhang and Guanghui Lan. Optimal algorithms for convex nested stochastic composite optimization, 2020. URL `https://arxiv.org/abs/2011.10076`.

Zhe Zhang and Guanghui Lan. Optimal algorithms for convex nested stochastic composite optimization. *ArXiv e-prints*, arXiv:2011.10076, 2021.

Yi Zhou, Zhe Wang, Kaiyi Ji, Yingbin Liang, and Vahid Tarokh. Momentum schemes with stochastic variance reduction for nonconvex composite optimization. *arXiv preprint arXiv:1902.02715*, 2019.

Dixian Zhu, Zhe Li, Xiaoyu Wang, Boqing Gong, and Tianbao Yang. A robust zero-sum game framework for pool-based active learning. In *The 22nd international conference on artificial intelligence and statistics*, pp. 517–526. PMLR, 2019.