Stochastic Approximation Approaches to Group Distributionally Robust Optimization and Beyond

Lijun Zhang, *Senior Member, IEEE,* Haomin Bai, Peng Zhao, *Member, IEEE,* Zhi-Hua Zhou, *Fellow, IEEE*

Abstract—This paper investigates group distributionally robust optimization (GDRO) with the goal of learning a model that performs well over m different distributions. First, we formulate GDRO as a stochastic convex-concave saddle-point problem and solve it using stochastic mirror descent (SMD) with m samples per iteration, attaining a nearly optimal sample complexity. To reduce the number of samples required in each round from m to 1, we cast GDRO as a two-player game, where one player conducts SMD and the other executes an online algorithm for non-oblivious multi-armed bandits, maintaining the same sample complexity. Next, we extend GDRO to address heterogeneous distributions that contain outliers. In such a scenario, we propose to optimize the average top-k risk instead of the maximum risk, thereby mitigating the impact of outlier distributions. Similar to the case of vanilla GDRO, we develop two stochastic approaches: one uses m samples per iteration via SMD, and the other consumes k samples per iteration through SMD and an online algorithm for non-oblivious combinatorial semi-bandits. Moreover, we propose *anytime* versions of the proposed algorithms, which can return solutions at any time without predefining the number of iterations.

Index Terms—Group distributionally robust optimization, Stochastic convex-concave saddle-point problem, Non-oblivious online learning, Bandits, Average top-k risk

1 Introduction

In the classical statistical machine learning, our goal is to minimize the risk with respect to a *fixed* distribution $\mathcal{P}_0$ [1], i.e.,

$$\min_{\mathbf{w}\in\mathcal{W}} \left\{R_0(\mathbf{w}) = \mathrm{E}_{\mathbf{z}\sim\mathcal{P}_0}\left[\ell(\mathbf{w};\mathbf{z})\right]\right\}, \tag{1}$$

where $\mathbf{z} \in \mathcal{Z}$ is a sample drawn from $\mathcal{P}_0$, $\mathcal{W}$ denotes a hypothesis class, and $\ell(\mathbf{w};\mathbf{z})$ is a loss measuring the prediction error of model $\mathbf{w}$ on $\mathbf{z}$. During the past decades, various methods have been developed to optimize (1), and can be grouped in two categories: sample average approximation (SAA) and stochastic approximation (SA) [2]. In SAA, we minimize an empirical risk defined as the average loss over a set of samples drawn from $\mathcal{P}_0$, and in SA, we directly solve the original problem by using stochastic observations of the objective $R_0(\cdot)$.

However, a model trained on a single distribution may lack robustness in the sense that (i) it could suffer high error on minority subpopulations, though the average loss is small; (ii) its performance could degenerate dramatically when tested on a different distribution. Distributionally robust optimization (DRO) provides a principled way to address those limitations by minimizing the worst-case risk in a neighborhood of $\mathcal{P}_0$ [3]. Recently, it has attracted great interest in optimization [4], statistics [5], operations research [6], and machine learning [7], [8], [9], [10]. In this paper, we consider an emerging class of DRO problems, named as Group DRO (GDRO), which optimizes the maximum risk

$$\mathcal{L}_{\max}(\mathbf{w}) = \max_{i\in[m]} \left\{R_i(\mathbf{w}) = \mathrm{E}_{\mathbf{z}\sim\mathcal{P}_i}\left[\ell(\mathbf{w};\mathbf{z})\right]\right\} \tag{2}$$

over a finite number of distributions, denoted as $\mathcal{P}_1, \ldots, \mathcal{P}_m$ [11], [12]. Mathematically, GDRO can be formulated as a minimax stochastic problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \mathcal{L}_{\max}(\mathbf{w}) = \min_{\mathbf{w}\in\mathcal{W}} \max_{i\in[m]} \left\{R_i(\mathbf{w})\right\}. \tag{3}$$

A motivating example is federated learning, where a centralized model is deployed at multiple clients, each of which faces a (possibly) different data distribution [13].

Supposing that samples can be drawn from all distributions freely, we develop efficient SA approaches for (3), in favor of their light computations over SAA methods. Following prior work [14, § 3.2], we can cast (3) as a stochastic convex-concave saddle-point problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{q}\in\Delta_m} \left\{\phi(\mathbf{w},\mathbf{q}) = \sum_{i=1}^m q_i R_i(\mathbf{w})\right\}, \tag{4}$$

where $\Delta_m = \{\mathbf{q} \in \mathbb{R}^m | \mathbf{q} \geq \mathbf{0}, \sum_{i=1}^m q_i = 1\}$ is the $(m-1)$-dimensional simplex, and then solve (4) by the mirror descent SA method, namely stochastic mirror descent (SMD). In fact, several recent studies have adopted this (or similar) strategy to optimize (4). But, unfortunately, we found that existing results are unsatisfactory since they either deliver a loose sample complexity [12], suffer subtle dependency issues in analysis [15], [16], or hold only in expectation [17].

As a starting point, we first provide a routine application of SMD to (4), and discuss the theoretical guarantee. In each iteration, we draw 1 sample from every distribution to construct unbiased estimators of $R_i(\cdot)$ and its gradient, and then update both $\mathbf{w}$ and $\mathbf{q}$ by SMD. The proposed method achieves an $O(\sqrt{(\log m)/T})$ convergence rate in expectation

• *Lijun Zhang, Haomin Bai, Peng Zhao and Zhi-Hua Zhou are with the State Key Laboratory of Novel Software Technology, and School of Artificial Intelligence, Nanjing University, Nanjing 210023, China. E-mail: {zhanglj, baihm, zhaop, zhouzh}@lamda.nju.edu.cn*
This work was partially supported by National Science and Technology Major Project (2022ZD0114801), NSFC (62361146852), and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118).

and with high probability, where T is the total number of iterations. As a result, we obtain an $O(m(\log m)/\epsilon^2)$ sample complexity for finding an ϵ-optimal solution of (4), which matches the $\Omega(m/\epsilon^2)$ lower bound [16] up to a logarithmic factor, and is tighter than the existing $O(m^2(\log m)/\epsilon^2)$ bound [12] by a factor of m. While being straightforward, this result seems *new* for GDRO. Additionally, we note that the above method requires setting the number of iterations T in advance, which may restrict its applicability. To avoid this limitation, we further propose an *anytime* algorithm by using time-varying step sizes, and obtain an $\widetilde{O}(\sqrt{(\log m)/t})$[1] convergence rate at each iteration t.

Then, we proceed to reduce the number of samples used per round from m to 1. We remark that a naive uniform sampling over m distributions does not work well, and yields a higher sample complexity [12]. As an alternative, we formulate the problem (4) as a two-player game, where we employ online learning methods with stochastic observations for both players and explicitly address the *non-oblivious* nature of the online process. Specifically, we use SMD to update $\mathbf{w}$, and Exp3-IX, a powerful algorithm for non-oblivious multi-armed bandits (MAB) [18], with stochastic rewards to update $\mathbf{q}$. In this way, our algorithm only needs 1 sample per round and attains an $O(\sqrt{m(\log m)/T})$ convergence rate, implying the same $O(m(\log m)/\epsilon^2)$ sample complexity. Similarly, we also put forward an anytime variant, achieving an $\widetilde{O}(\sqrt{m(\log m)/t})$ convergence rate.

Subsequently, we extend GDRO to address heterogeneous distributions, where the risks vary significantly across distributions [19]. The widely acknowledged sensitivity of the *max* operation to outliers implies that GDRO could be dominated by a single outlier distribution, while neglecting others [20]. Inspired by the average top-k loss for supervised learning [21], we modify our objective from the maximum risk $\mathcal{L}_{\max}(\mathbf{w})$ in GDRO to the average top-k risk:

$$\mathcal{L}_k(\mathbf{w}) = \max_{\mathcal{I}\in\mathbf{B}_{m,k}} \left\{ \frac{1}{k}\sum_{i\in\mathcal{I}} R_i(\mathbf{w}) \right\}, \tag{5}$$

where $\mathbf{B}_{m,k}$ is the set of subsets of $[m]$ with size k, i.e., $\mathbf{B}_{m,k} = \{\mathcal{I} \subseteq [m] || \mathcal{I}| = k\}$. This modification aims to mitigate the impact of outlier distributions while still including GDRO as a special case when $k = 1$.

We refer to the minimization of $\mathcal{L}_k(\mathbf{w})$ as average top-k risk optimization (AT$_k$RO), and develop two stochastic algorithms. Similar to GDRO, AT$_k$RO can be formulated as a stochastic convex-concave saddle-point problem, akin to (4), with the only difference being that the domain of $\mathbf{q}$ is the capped simplex defined by m and k, instead of the standard simplex. Therefore, we can employ SMD to update $\mathbf{w}$ and $\mathbf{q}$, which uses m samples in each round. Theoretical analysis demonstrates that this approach achieves an $O(\sqrt{(\log(m/k))/T})$ convergence rate, implying an $O(m(\log(m/k))/\epsilon^2)$ sample complexity. Furthermore, to circumvent the limitation of predefining the total number of iterations T, we introduce an anytime version that attains an $\widetilde{O}(\sqrt{(\log(m/k))/t})$ convergence rate.

Following the second approach for GDRO, we reduce the number of samples required in each round from m to k by casting AT$_k$RO as a two-player game. For updating $\mathbf{w}$, we construct unbiased stochastic gradients and still apply SMD. For updating $\mathbf{q}$, we model the online problem as an instance of non-oblivious combinatorial semi-bandits to satisfy the capped simplex constraint. Then, we extend Exp3-IX to design a strategy, which selects k distributions and draws 1 sample from each per round. The algorithm is proved to achieve an $O(\sqrt{m(\log m)/(kT)})$ convergence rate, yielding an $O(m(\log m)/\epsilon^2)$ sample complexity. Similarly, we design an anytime variant that uses 1 sample per round and achieves an $\widetilde{O}(\sqrt{m(\log m)/t})$ rate.

This paper extends our conference version [22] by developing anytime algorithms, investigating a new scenario, and conducting more experiments, as detailed below[2].

- First, we adapt the two SA algorithms for GDRO to operate in an anytime manner. In the conference paper, our algorithms for GDRO required predefining the total number of iterations T to set step sizes. We design anytime algorithms by adopting time-varying step sizes, and provide the corresponding theoretical analysis.
- Second, we explore the scenario of heterogeneous distributions with high-risk outliers. To mitigate the impact of outliers, we formulate AT$_k$RO and propose two algorithms: one uses SMD with m samples per round and achieves an $O(m(\log(m/k))/\epsilon^2)$ sample complexity; the other combines SMD with an extension of Exp3-IX using k samples per round and achieves a sample complexity of $O(m(\log m)/\epsilon^2)$. Both algorithms are further extended to anytime versions.
- Last, we construct a heterogeneous data set and perform experiments to verify the advantages of AT$_k$RO. Additionally, we compare the performance of the anytime algorithms with their non-anytime counterparts, and further validate the effectiveness of the proposed algorithms in a non-convex setting.

2 Related Work

Distributionally robust optimization (DRO) stems from the pioneering work of [23], and has gained a lot of interest with the advancement of robust optimization [24], [25]. It has been applied to diverse machine learning tasks, including adversarial training [26], algorithmic fairness [27], class imbalance [28], long-tail learning [29], label shift [30], etc.

In general, DRO is formulated to reflect our uncertainty about the target distribution. To ensure good performance under distribution perturbations, it minimizes the risk w.r.t. the worst distribution in an uncertainty set, i.e.,

$$\min_{\mathbf{w}\in\mathcal{W}} \sup_{\mathcal{P}\in\mathcal{S}(\mathcal{P}_0)} \left\{ \mathrm{E}_{\mathbf{z}\sim\mathcal{P}}\left[\ell(\mathbf{w};\mathbf{z})\right] \right\}, \tag{6}$$

where $\mathcal{S}(\mathcal{P}_0)$ is a set of probability distributions around $\mathcal{P}_0$. There are mainly three ways to construct $\mathcal{S}(\mathcal{P}_0)$: (i) enforcing moment constraints [31], (ii) defining a neighborhood around $\mathcal{P}_0$ by a distance function such as the f-divergence [3], the Wasserstein distance [32], and the Sinkhorn distance [33], and (iii) hypothesis testing of goodness-of-fit [34]. Additional related work on empirical DRO and optimization algorithms for DRO is deferred to Appendix A.

1. We use the $\widetilde{O}$ notation to hide constant factors as well as polylogarithmic factors in t.

2. Due to space limitations, the journal version omits the imbalanced data setting presented in our conference paper [22].

The main focus of this paper is the GDRO problem in (3)/(4), instead of the traditional DRO in (6). Sagawa et al. [12] have applied SMD [14] to (4), but only obtain a sub-optimal sample complexity of $O(m^2(\log m)/\epsilon^2)$ due to large gradient variance. Subsequent works attempt to lower the sample complexity by reusing samples [15] and applying techniques from MAB [16], but their analysis suffers dependency issues. Carmon and Hausler [17, Proposition 2] successfully establish an $O(m(\log m)/\epsilon^2)$ sample complexity by combining SMD and gradient clipping, but their result holds only in expectation. More recently, sharper sample complexity bounds with improved dependence on m have been obtained under additional sparsity assumptions [35].

To deal with heterogeneous noise across distributions, minimax regret optimization (MRO) [10] replaces the risk $R_i(\mathbf{w})$ with excess risk $R_i(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{W}} R_i(\mathbf{w})$, while more general calibration terms can also prevent a single distribution from dominating the maximum [36]. Efficient methods have been investigated for MRO [37], as well as for empirical GDRO and empirical MRO [38]. AT$_k$RO shares a similar motivation with MRO, but replaces the maximum risk in GDRO with the average top-k risk. In addition, GDRO has a similar spirit to collaborative PAC learning [39], [40], [41], in the sense that both aim to find a single model that performs well on multiple distributions. Finally, related distribution-shift phenomena have also been studied in cross-domain applications [42], such as image generation, where representation methods [43], [44] have shown promising effectiveness in jointly modeling and leveraging diverse domain distributions.

3 SA Approaches to GDRO

In this section, we present two efficient SA approaches for GDRO, which achieve the same sample complexity but use a different number of samples in each round (m versus 1).

3.1 Preliminaries

First, we state the general setup of mirror descent [14]. We equip the domain $\mathcal{W}$ with a distance-generating function $\nu_w(\cdot)$, which is 1-strongly convex with respect to certain norm $\|\cdot\|_w$. We define the Bregman distance associated with $\nu_w(\cdot)$ as

$$B_w(\mathbf{u},\mathbf{v}) = \nu_w(\mathbf{u}) - \big[\nu_w(\mathbf{v}) + \langle\nabla\nu_w(\mathbf{v}), \mathbf{u}-\mathbf{v}\rangle\big].$$

For the simplex Δ_m, we choose the negative entropy (neg-entropy) function $\nu_q(\mathbf{q}) = \sum_{i=1}^m q_i \ln q_i$, which is 1-strongly convex with respect to the vector ℓ_1-norm $\|\cdot\|_1$, as the distance-generating function. Similarly, $B_q(\cdot,\cdot)$ is the Bregman distance associated with $\nu_q(\cdot)$.

Then, we introduce the standard assumptions about the domain and the loss function.

Assumption 1. *The domain $\mathcal{W}$ is convex and its diameter measured by $\nu_w(\cdot)$ is bounded by D, i.e.,*

$$\max_{\mathbf{w}\in\mathcal{W}} \nu_w(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{W}} \nu_w(\mathbf{w}) \le D^2. \tag{7}$$

For Δ_m, it is easy to verify that its diameter measured by the neg-entropy function is bounded by $\sqrt{\ln m}$.

Assumption 2. *For all $i \in [m]$, the risk function $R_i(\mathbf{w}) = \mathrm{E}_{\mathbf{z}\sim\mathcal{P}_i}[\ell(\mathbf{w};\mathbf{z})]$ is convex.*

To simplify presentations, we assume the loss belongs to $[0,1]$, and its gradient is also bounded.

Assumption 3. *For all $i \in [m]$, we have*

$$0 \le \ell(\mathbf{w};\mathbf{z}) \le 1,\ \forall \mathbf{w}\in\mathcal{W},\ \mathbf{z}\sim\mathcal{P}_i. \tag{8}$$

Assumption 4. *For all $i \in [m]$, we have*

$$\|\nabla\ell(\mathbf{w};\mathbf{z})\|_{w,*} \le G,\ \forall \mathbf{w}\in\mathcal{W},\ \mathbf{z}\sim\mathcal{P}_i \tag{9}$$

where $\|\cdot\|_{w,*}$ is the dual norm of $\|\cdot\|_w$.

Note that it is possible to relax the bounded assumption in (9) to light tail conditions, such as the sub-Gaussian property [45].

Last, we discuss the performance measure. To analyze the convergence property, we measure the quality of an approximate solution $(\bar{\mathbf{w}},\bar{\mathbf{q}})$ to (4) by the error

$$\epsilon_\phi(\bar{\mathbf{w}},\bar{\mathbf{q}}) = \max_{\mathbf{q}\in\Delta_m} \phi(\bar{\mathbf{w}},\mathbf{q}) - \min_{\mathbf{w}\in\mathcal{W}} \phi(\mathbf{w},\bar{\mathbf{q}}), \tag{10}$$

which directly controls the optimality of $\bar{\mathbf{w}}$ to the original problem (3) [22, (9)]:

$$\max_{i\in[m]} R_i(\bar{\mathbf{w}}) - \min_{\mathbf{w}\in\mathcal{W}} \max_{i\in[m]} R_i(\mathbf{w}) \le \epsilon_\phi(\bar{\mathbf{w}},\bar{\mathbf{q}}). \tag{11}$$

3.2 Stochastic Mirror Descent for GDRO

To apply SMD, the key is to construct stochastic gradients of the function $\phi(\mathbf{w},\mathbf{q})$ in (4). We first present its true gradients with respect to $\mathbf{w}$ and $\mathbf{q}$:

$$\nabla_{\mathbf{w}}\phi(\mathbf{w},\mathbf{q}) = \sum_{i=1}^m q_i \nabla R_i(\mathbf{w}),$$
$$\nabla_{\mathbf{q}}\phi(\mathbf{w},\mathbf{q}) = [R_1(\mathbf{w}),\ldots,R_m(\mathbf{w})]^\top.$$

In each round t, denote by $\mathbf{w}_t$ and $\mathbf{q}_t$ the current solutions. We draw one sample $\mathbf{z}_t^{(i)}$ from every distribution $\mathcal{P}_i$, and define stochastic gradients as

$$\begin{aligned}\mathbf{g}_w(\mathbf{w}_t,\mathbf{q}_t) &= \sum_{i=1}^m q_{t,i}\nabla\ell(\mathbf{w}_t;\mathbf{z}_t^{(i)}),\\ \mathbf{g}_q(\mathbf{w}_t,\mathbf{q}_t) &= [\ell(\mathbf{w}_t;\mathbf{z}_t^{(1)}),\ldots,\ell(\mathbf{w}_t;\mathbf{z}_t^{(m)})]^\top.\end{aligned} \tag{12}$$

Obviously, they are unbiased estimators of the true gradients: $\mathrm{E}_{t-1}[\mathbf{g}_w(\mathbf{w}_t,\mathbf{q}_t)] = \nabla_{\mathbf{w}}\phi(\mathbf{w}_t,\mathbf{q}_t)$ and $\mathrm{E}_{t-1}[\mathbf{g}_q(\mathbf{w}_t,\mathbf{q}_t)] = \nabla_{\mathbf{q}}\phi(\mathbf{w}_t,\mathbf{q}_t)$ where $\mathrm{E}_{t-1}[\cdot]$ represents the expectation conditioned on the randomness until round $t-1$. Moreover, $\mathbf{g}_w(\mathbf{w}_t,\mathbf{q}_t)$ can be simplified to

$$\tilde{\mathbf{g}}_w(\mathbf{w}_t,\mathbf{q}_t) = \nabla\ell(\mathbf{w}_t;\mathbf{z}_t^{(i_t)}), \tag{13}$$

where $i_t \in [m]$ is drawn randomly according to $\mathbf{q}_t$.

Then, we use SMD to update $\mathbf{w}_t$ and $\mathbf{q}_t$:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \big\{\eta_w\langle\mathbf{g}_w(\mathbf{w}_t,\mathbf{q}_t),\mathbf{w}\rangle + B_w(\mathbf{w},\mathbf{w}_t)\big\}, \tag{14}$$

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\Delta_m} \big\{\eta_q\langle-\mathbf{g}_q(\mathbf{w}_t,\mathbf{q}_t),\mathbf{q}\rangle + B_q(\mathbf{q},\mathbf{q}_t)\big\}, \tag{15}$$

where $\eta_w > 0$ and $\eta_q > 0$ are two step sizes that will be determined later. The updating rule of $\mathbf{w}_t$ depends on the choice of the distance-generating function $\nu_w(\cdot)$. If $\nu_w(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$, (14) becomes stochastic gradient descent (SGD), i.e.,

Algorithm 1 Stochastic Mirror Descent for GDRO

Input: step sizes η_w and η_q

1: Initialize $\mathbf{w}_1$ and $\mathbf{q}_1$ according to (17)
2: **for** $t = 1$ to T **do**
3: For each $i \in [m]$, draw a sample $\mathbf{z}_t^{(i)}$ from $\mathcal{P}_i$
4: Construct the stochastic gradients defined in (12)
5: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ via (14) and (15), respectively
6: **end for**
7: **return** $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^T \mathbf{w}_t$ and $\bar{\mathbf{q}} = \frac{1}{T}\sum_{t=1}^T \mathbf{q}_t$

$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\big[\mathbf{w}_t - \eta_w \mathbf{g}_w(\mathbf{w}_t, \mathbf{q}_t)\big]$, where $\Pi_{\mathcal{W}}[\cdot]$ denotes the Euclidean projection onto $\mathcal{W}$. Since $B_q(\mathbf{q}, \mathbf{q}_t)$ is defined in terms of the neg-entropy, (15) is equivalent to

$$q_{t+1,i} = \frac{q_{t,i}\exp\left(\eta_q \ell(\mathbf{w}_t; \mathbf{z}_t^{(i)})\right)}{\sum_{j=1}^m q_{t,j}\exp\left(\eta_q \ell(\mathbf{w}_t; \mathbf{z}_t^{(j)})\right)},\ \forall i \in [m] \tag{16}$$

which is the Hedge algorithm [46] applied to a maximization problem. In the beginning, we initialize

$$\mathbf{w}_1 = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \nu_w(\mathbf{w}),\ \text{and}\ \mathbf{q}_1 = \frac{1}{m}\mathbf{1}_m, \tag{17}$$

where $\mathbf{1}_m$ is the m-dimensional vector consisting of 1's. In the last step, we return the averaged iterates $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^T \mathbf{w}_t$ and $\bar{\mathbf{q}} = \frac{1}{T}\sum_{t=1}^T \mathbf{q}_t$ as final solutions. The complete procedure is summarized in Algorithm 1.

Based on the theoretical guarantee of SMD for stochastic convex-concave optimization [14, § 3.1], we have the following theorem for Algorithm 1.

Theorem 1. *Under Assumptions 1-4, and setting* $\eta_w = D^2\sqrt{\frac{8}{5T(D^2G^2+\ln m)}}$ *and* $\eta_q = (\ln m)\sqrt{\frac{8}{5T(D^2G^2+\ln m)}}$ *in Algorithm 1, we have*

$$\mathrm{E}\big[\epsilon_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}})\big] \le 2\sqrt{\frac{10(D^2G^2+\ln m)}{T}},$$

and with probability at least $1-\delta$,

$$\epsilon_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) \le \left(8 + 2\ln\frac{2}{\delta}\right)\sqrt{\frac{10(D^2G^2+\ln m)}{T}}.$$

Remark 1. Theorem 1 shows that Algorithm 1 achieves an $O(\sqrt{(\log m)/T})$ convergence rate. Since it uses m samples per round, the sample complexity is $O(m(\log m)/\epsilon^2)$, nearly matching the $\Omega(m/\epsilon^2)$ lower bound [16, Theorem 5].

Remark 2 (Comparisons with Related Work [12])**.** Since the number of samples used in each round of Algorithm 1 is m, it is natural to ask whether it can be reduced to a small constant. Indeed, the stochastic algorithm of Sagawa et al. [12] requires 1 sample per round, but suffers a large sample complexity. In each round t, they first generate a random index $i_t \in [m]$ uniformly, and draw 1 sample $\mathbf{z}_t^{(i_t)}$ from $\mathcal{P}_{i_t}$. The stochastic gradients are constructed as

$$\begin{aligned}\hat{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t) &= q_{t,i_t} m \nabla \ell(\mathbf{w}_t; \mathbf{z}_t^{(i_t)}),\\ \hat{\mathbf{g}}_q(\mathbf{w}_t, \mathbf{q}_t) &= [0, \ldots, 0, m\ell(\mathbf{w}_t; \mathbf{z}_t^{(i_t)}), 0, \ldots, 0]^\top.\end{aligned} \tag{18}$$

Then, these gradients are used to update $\mathbf{w}_t$ and $\mathbf{q}_t$ as in (14) and (15). However, it attains a slow convergence rate of $O(m\sqrt{(\log m)/T})$, leading to an $O(m^2(\log m)/\epsilon^2)$ sample complexity, which is higher than that of Algorithm 1 by a factor of m. This slowdown arises because the optimization error depends on the dual norm of the gradients in (18), which is larger by a factor of m than that in (12).

Remark 3 (Comparisons with Related Work [15])**.** To reduce the number of samples required in each round, Haghtalab et al. [15] propose to reuse samples for multiple iterations. To approximate $\nabla_{\mathbf{w}}\phi(\mathbf{w}_t, \mathbf{q}_t)$, they construct $\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)$ in (13) using 1 sample. To approximate $\nabla_{\mathbf{q}}\phi(\mathbf{w}_t, \mathbf{q}_t)$, they draw m samples $\mathbf{z}_\tau^{(1)}, \ldots, \mathbf{z}_\tau^{(m)}$, one from each distribution, at round $\tau = mk+1$, $k \in \mathbb{Z}$, and reuse them for m iterations to construct the following gradient:

$$\mathbf{g}'_q(\mathbf{w}_t, \mathbf{q}_t) = [\ell(\mathbf{w}_t; \mathbf{z}_\tau^{(1)}), \ldots, \ell(\mathbf{w}_t; \mathbf{z}_\tau^{(m)})]^\top, \tag{19}$$

for $t = \tau, \ldots, \tau+m-1$. Then, they treat $\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)$ and $\mathbf{g}'_q(\mathbf{w}_t, \mathbf{q}_t)$ as stochastic gradients, and update $\mathbf{w}_t$ and $\mathbf{q}_t$ by SMD. In this way, their algorithm uses 2 samples on average in each iteration. However, the gradient in (19) is no longer an unbiased estimator of the true gradient $\nabla_{\mathbf{q}}\phi(\mathbf{w}_t, \mathbf{q}_t)$ at rounds $t = \tau+2, \ldots, \tau+m-1$, making their analysis ungrounded. To see this, from the updating rule of SMD, we know that $\mathbf{w}_{\tau+2}$ depends on $\mathbf{q}_{\tau+1}$, which in turn depends on the m samples drawn at round τ, and thus $\mathrm{E}\left[\ell(\mathbf{w}_{\tau+2}; \mathbf{z}_\tau^{(i)})\right] \ne R_i(\mathbf{w}_{\tau+2}),\ i = 1, \ldots, m$.

3.2.1 Anytime Extension of Algorithm 1

In Theorem 1, the dependence of step sizes η_w and η_q on the total number of iterations T complicates practical implementation, as T needs to be set in advance. Moreover, Algorithm 1 only returns a solution after T iterations. To avoid these limitations, we propose an anytime extension of Algorithm 1 by employing time-varying step sizes. We note that there is a long-standing history of designing anytime algorithms in optimization and related areas [47], [48].

Specifically, we replace the fixed step sizes η_w and η_q in (14) and (15) with time-varying step sizes [14]

$$\eta_t^w = D^2\sqrt{\frac{2}{Ct}},\ \text{and}\ \eta_t^q = (\ln m)\sqrt{\frac{2}{Ct}}, \tag{20}$$

where $C = D^2G^2 + \ln m$. To enable anytime capability, we maintain the weighted averages of the iterates:

$$\bar{\mathbf{w}}_t = \sum_{j=1}^t \frac{\eta_j^w \mathbf{w}_j}{\sum_{k=1}^t \eta_k^w},\ \text{and}\ \bar{\mathbf{q}}_t = \sum_{j=1}^t \frac{\eta_j^q \mathbf{q}_j}{\sum_{k=1}^t \eta_k^q}, \tag{21}$$

which can be returned as solutions whenever required, and provide the following theoretical guarantee at each round.

Theorem 2. *Under Assumptions 1-4, and modifying step sizes as* (20) *in Algorithm 1, we have for all* $t \in \mathbb{Z}_+$,

$$\mathrm{E}\big[\epsilon_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)\big] \le \frac{\sqrt{D^2G^2+\ln m}}{\sqrt{2}\left(\sqrt{t+1}-1\right)}\left(5 + 3\ln t\right),$$

and for each $t \in \mathbb{Z}_+$, *with probability at least* $1-\delta$,

$$\epsilon_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t) = O\left(\frac{\sqrt{\log m}}{\sqrt{t}}\left(\log t \cdot \log\frac{1}{\delta}\right)\right).$$

Remark 4. The convergence rate of the anytime extension is slower by a factor of $O(\log t)$ than that of Algorithm 1. However, the modified algorithm possesses the anytime characteristic, i.e., it can return a solution at any round.

Algorithm 2 Non-oblivious Online Learning for GDRO

Input: step sizes η_w and η_q, and IX coefficient γ

1: Initialize $\mathbf{w}_1$ and $\mathbf{q}_1$ according to (17)
2: **for** $t = 1$ to T **do**
3: Generate $i_t \in [m]$ according to $\mathbf{q}_t$, and draw a sample $\mathbf{z}_t^{(i_t)}$ from distribution $\mathcal{P}_{i_t}$
4: Construct the stochastic gradient in (13) and the IX loss estimator in (23)
5: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ via (22) and (24), respectively
6: **end for**
7: **return** $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$ and $\bar{\mathbf{q}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{q}_t$

3.3 Non-oblivious Online Learning for GDRO

In this section, we explore methods to reduce the number of samples used per round from m to 1. As shown in (13), we can use 1 sample to construct a stochastic gradient $\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)$ for $\mathbf{w}_t$ with small norm under Assumption 4. Thus, the error related to $\mathbf{w}_t$ is relatively easy to control. However, we do not have such guarantees for the stochastic gradient of $\mathbf{q}_t$. Recall that the infinity norm of $\hat{\mathbf{g}}_q(\mathbf{w}_t, \mathbf{q}_t)$ in (18) is upper bounded by m. The reason is that we insist on the unbiasedness of the stochastic gradient, which induces large variance. To address this issue, we adopt online learning techniques to balance the bias and the variance.

It is now well known that convex-concave saddle-point problems can be solved by playing two online learning algorithms against each other [49], [50], [51]. This transformation allows us to exploit no-regret algorithms developed in online learning to bound the optimization error. To solve problem (4), we ask the 1st player to minimize a sequence of convex functions $\{\phi(\mathbf{w}, \mathbf{q}_t) = \sum_{i=1}^{m} q_{t,i} R_i(\mathbf{w})\}_{t\in[T]}$ over $\mathbf{w} \in \mathcal{W}$ and the 2nd player maximizes a sequence of linear functions $\{\phi(\mathbf{w}_t, \mathbf{q}) = \sum_{i=1}^{m} q_i R_i(\mathbf{w}_t)\}_{t\in[T]}$ over $\mathbf{q} \in \Delta_m$. We highlight that there is an important difference between our stochastic convex-concave problem and its deterministic counterpart. Here, the two players cannot directly observe the loss function, and can only approximate $R_i(\mathbf{w}) = \mathrm{E}_{\mathbf{z}\sim\mathcal{P}_i}\big[\ell(\mathbf{w}; \mathbf{z})\big]$ by drawing samples from $\mathcal{P}_i$. The stochastic setting makes the problem more challenging, and in particular, we need to take care of the *non-oblivious* nature of the learning process. In this context, "non-oblivious" refers to the fact that the online functions depend on the past decisions of the players.

Next, we discuss the online algorithms used by the two players. As shown in Section 3.2, the 1st player can easily obtain a stochastic gradient with small norm by using 1 sample. So, we model the problem faced by the 1st player as "non-oblivious online convex optimization (OCO) with stochastic gradients", and still use SMD to update its solution. In each round t, with 1 sample drawn from $\mathcal{P}_i$, the 2nd player can estimate the value of $R_i(\mathbf{w}_t)$ which is the coefficient of q_i. Since the 2nd player is maximizing a linear function over the simplex, the problem can be modeled as "non-oblivious multi-armed bandits (MAB) with stochastic rewards". And fortunately, we have powerful online algorithms for non-oblivious MAB [52], [53], whose regret grows sublinearly with m. In this paper, we choose the Exp3-IX algorithm [18], and generalize its theoretical guarantee to stochastic rewards. In contrast, if we apply SMD with $\hat{\mathbf{g}}_q(\mathbf{w}_t, \mathbf{q}_t)$ in (18), the regret scales at least linearly with m.

The complete procedure is presented in Algorithm 2, and we explain key steps below. In each round t, we generate an index $i_t \in [m]$ from the probability distribution $\mathbf{q}_t$, and then draw a sample $\mathbf{z}_t^{(i_t)}$ from the distribution $\mathcal{P}_{i_t}$. With the stochastic gradient in (13), we use SMD to update $\mathbf{w}_t$:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \big\{\eta_w \langle \tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t), \mathbf{w}\rangle + B_w(\mathbf{w}, \mathbf{w}_t)\big\}. \quad (22)$$

Then, we reuse the sample $\mathbf{z}_t^{(i_t)}$ to update $\mathbf{q}_t$ according to Exp3-IX, which first constructs the Implicit-eXploration (IX) loss estimator [54]:

$$\tilde{s}_{t,i} = \frac{1 - \ell(\mathbf{w}_t, \mathbf{z}_t^{(i_t)})}{q_{t,i} + \gamma} \cdot \mathbb{I}[i_t = i],\ \forall i \in [m], \quad (23)$$

where $\gamma > 0$ is the IX coefficient and $\mathbb{I}[A]$ equals to 1 when the event A is true and 0 otherwise, and then performs a mirror descent update:

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\Delta_m} \big\{\eta_q \langle \tilde{\mathbf{s}}_t, \mathbf{q}\rangle + B_q(\mathbf{q}, \mathbf{q}_t)\big\}. \quad (24)$$

Compared with (15), the only difference is that the stochastic gradient is now replaced with the IX loss estimator $\tilde{\mathbf{s}}_t$. However, it is not an instance of SMD, as $\tilde{\mathbf{s}}_t$ is no longer an unbiased stochastic gradient. The main advantage of $\tilde{\mathbf{s}}_t$ is that it reduces the variance of the gradient estimator by sacrificing a little bit of unbiasedness, which is crucial for a high probability guarantee, and thus can deal with non-oblivious adversaries. Since we still use the entropy regularizer, (24) also enjoys an explicit form similar to (16).

We present the theoretical guarantee of Algorithm 2. The analysis proceeds by bounding the regret of each player individually. For the 1st player, we address the non-obliviousness by the "ghost iterate" technique [14]. For the 2nd player, we extend Exp3-IX to stochastic rewards and obtain a corresponding regret bound. By combining these two results, we obtain the following theorem.

Theorem 3. *Under Assumptions 1-4, and setting* $\eta_w = \frac{2D}{G\sqrt{5T}}$, $\eta_q = \sqrt{\frac{\ln m}{mT}}$ *and* $\gamma = \frac{\eta_q}{2}$ *in Algorithm 2, we have*

$$\mathrm{E}\big[\epsilon_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}})\big] \le O\left(\sqrt{\frac{m\log m}{T}}\right),$$

and with probability at least $1 - \delta$,

$$\epsilon_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) = O\left(\sqrt{\frac{m\log m}{T}} + \sqrt{\frac{m}{T\log m}}\log\frac{1}{\delta}\right).$$

Remark 5. Theorem 3 shows that with 1 sample per round, Algorithm 2 achieves an $O(\sqrt{m(\log m)/T})$ convergence rate, maintaining the $O(m(\log m)/\epsilon^2)$ sample complexity.

Remark 6 (Comparisons with Related Work [16])**.** Soma et al. [16] apply online algorithms to optimize $\mathbf{w}$ and $\mathbf{q}$, but ignore the non-oblivious property. As a result, their guarantees, based on analysis for the oblivious setting [55], cannot justify the optimality of their algorithm for (4). Specifically, their results imply that for any *fixed* $\mathbf{w}$ and $\mathbf{q}$ independent of $\bar{\mathbf{w}}$ and $\bar{\mathbf{q}}$, $\mathrm{E}\left[\phi(\bar{\mathbf{w}}, \mathbf{q}) - \phi(\mathbf{w}, \bar{\mathbf{q}})\right] = O\left(\sqrt{\frac{m}{T}}\right)$. However, this bound cannot control $\epsilon_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) = \max_{\mathbf{q}\in\Delta_m}\phi(\bar{\mathbf{w}}, \mathbf{q}) - \min_{\mathbf{w}\in\mathcal{W}}\phi(\mathbf{w}, \bar{\mathbf{q}}) = \phi(\bar{\mathbf{w}}, \widehat{\mathbf{q}}) - \phi(\widehat{\mathbf{w}}, \bar{\mathbf{q}})$ in (10), since $\widehat{\mathbf{w}} =$

$\operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} \phi(\mathbf{w}, \bar{\mathbf{q}})$ and $\widehat{\mathbf{q}} = \operatorname{argmax}_{\mathbf{q}\in\Delta_m} \phi(\bar{\mathbf{w}}, \mathbf{q})$ *depend* on $\bar{\mathbf{q}}$ and $\bar{\mathbf{w}}$, respectively.

Remark 7. After we pointed out the dependence issue of reusing samples, Haghtalab et al. [56] modified their method by incorporating bandit algorithms to optimize $\mathbf{q}$. From our understanding, the idea of applying bandits to GDRO was *firstly* proposed by Soma et al. [16], and then refined by us.

3.3.1 Anytime Extension of Algorithm 2

Following the extension of Algorithm 1, we adapt Algorithm 2 into an anytime variant by employing time-varying step sizes in both SMD and Exp3-IX. Specifically, in the t-th round, we replace η_w in (22), η_q in (24), and γ in (23) with

$$\eta_t^w = \frac{D}{G\sqrt{t}}, \ \eta_t^q = \sqrt{\frac{\ln m}{mt}}, \text{ and } \gamma_t = \frac{\eta_t^q}{2}, \tag{25}$$

respectively, and output $\bar{\mathbf{w}}_t$ and $\bar{\mathbf{q}}_t$ in (21) as solutions.

Compared to the original Algorithm 2, our modifications are relatively minor. Nevertheless, the theoretical analysis is different, since the error $\epsilon_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)$ is governed by the *weighted average regret* of the two players, rather than the standard regret. By separately analyzing the upper bounds of the weighted average regrets for both players (see Appendix E.4), we derive the following theorem.

Theorem 4. *Under Assumptions 1-4, and modifying parameters as* (25) *in Algorithm 2, we have for all* $t \in \mathbb{Z}_+$,

$$\mathrm{E}\left[\epsilon_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)\right] = O\left(\frac{\sqrt{m\log m}\log t}{\sqrt{t}}\right),$$

and for each $t \in \mathbb{Z}_+$, *with probability at least* $1-\delta$,

$$\epsilon_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t) = O\left(\frac{\sqrt{m\log m}\log t + \left(\sqrt{m/\log m} + \sqrt{\log t}\right)\log\frac{1}{\delta}}{\sqrt{t}}\right).$$

Remark 8. Similar to Theorem 2, the convergence rate in Theorem 4 is $O(\log t)$ times slower than that in Theorem 3.

4 AT$_k$RO FOR HETEROGENEOUS DISTRIBUTIONS

GDRO is effective in dealing with homogeneous distributions, where the risks of all distributions are roughly of the same order. However, its effectiveness diminishes when confronted with heterogeneous distributions. This stems from the sensitivity of the *max* operator to outlier distributions with significantly high risks, causing it to focus solely on outliers and overlook others [20]. To address this issue, research in robust supervised learning has introduced the approach of minimizing the average of the k largest individual losses [8], [21]. Inspired by these studies, we propose to optimize the average top-k risk $\mathcal{L}_k(\mathbf{w})$ in (5), which can mitigate the influence of outliers.

4.1 Preliminaries

By replacing $\mathcal{L}_{\max}(\mathbf{w})$ with $\mathcal{L}_k(\mathbf{w})$ in (3), we obtain the average top-k risk optimization (AT$_k$RO) problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \mathcal{L}_k(\mathbf{w}) = \min_{\mathbf{w}\in\mathcal{W}} \max_{\mathcal{I}\in\mathbf{B}_{m,k}} \left\{\frac{1}{k}\sum_{i\in\mathcal{I}} R_i(\mathbf{w})\right\}, \tag{26}$$

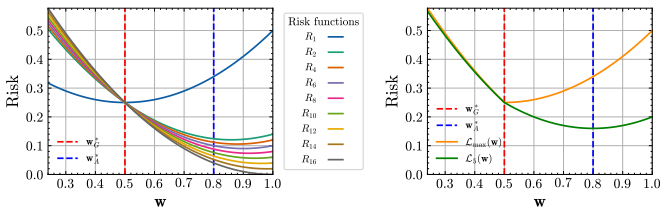

Fig. 1. Graphical illustrations of Example 1. (a) The individual risk $R_i(\mathbf{w})$ for 9 out of 16 distributions. (b) $\mathcal{L}_{\max}(\mathbf{w})$ and $\mathcal{L}_5(\mathbf{w})$

which reduces to GDRO when $k = 1$. Before introducing specific optimization algorithms, we present an example to illustrate the difference between GDRO and AT$_k$RO.

Example 1. *We define the hypothesis space as* $\mathcal{W} = [0, 1]$ *and the Bernoulli distribution as* $\mathrm{Ber}(\mu, 1)$, *which outputs 1 with probability* μ *and 0 with probability* $1-\mu$. *Then, we consider 16 distributions:* $\mathrm{Ber}(\mu_i, 1)$ *where* μ_i *is sequentially set to* $0.5, 0.86, 0.87, \ldots, 0.99, 1$. *The loss function is defined as* $\ell(\mathbf{w}; \mathbf{z}) = (\mathbf{w}-\mathbf{z})^2$ *for a random sample* $\mathbf{z} \in \{0, 1\}$ *drawn from these distributions. We denote the solutions of GDRO and AT*$_5$*RO by* $\mathbf{w}_G^*$ *and* $\mathbf{w}_A^*$, *respectively. It is easy to show that* $\mathbf{w}_G^* = 0.5$ *and* $\mathbf{w}_A^* = 0.8$, *as detailed in Appendix G.*

Fig. 1 shows a portion of the risk functions, the objectives of GDRO and AT$_5$RO, and their respective solutions. As illustrated in Fig. 1a, distribution $\mathcal{P}_1$ is significantly different from the others, suggesting it could be an outlier. Fig. 1b further shows that GDRO mainly focuses on $\mathcal{P}_1$, yielding $\mathbf{w}_G^* = \operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} R_1(\mathbf{w}) = 0.5$. While achieving low risk on $\mathcal{P}_1$, $\mathbf{w}_G^*$ performs poorly on the remaining distributions. In contrast, AT$_5$RO yields a more balanced solution $\mathbf{w}_A^* = 0.8$ by accounting for the top-5 risks. On $\mathcal{P}_2, \ldots, \mathcal{P}_{16}$, $\mathbf{w}_A^*$ achieves an average risk 0.168 lower than $\mathbf{w}_G^*$, with only a 0.09 increase on $\mathcal{P}_1$. Thus, AT$_5$RO mitigates the outlier effect of $\mathcal{P}_1$ and demonstrates superior robustness over GDRO.

Similar to the case of GDRO, (26) can be cast as a stochastic convex-concave saddle-point problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\mathbf{q}\in\mathcal{S}_{m,k}} \left\{\phi(\mathbf{w}, \mathbf{q}) = \sum_{i=1}^m q_i R_i(\mathbf{w})\right\}, \tag{27}$$

where $\mathcal{S}_{m,k} = \left\{\mathbf{q} \in \mathbb{R}^m \mid 0 \le q_i \le \frac{1}{k}, \sum_{i=1}^m q_i = 1\right\}$ is the capped simplex which can be viewed as the slice of the hyper-cube $[0, 1/k]^m$ cut by a hyper-plane $\mathbf{q}^\top\mathbf{1} = 1$. The difference between (4) and (27) lies in the domain of $\mathbf{q}$, which is Δ_m and $\mathcal{S}_{m,k}$ respectively.

Note that a similar convex–concave optimization problem has been studied in previous works [8], [51]. However, they focus on the deterministic setting, whereas our work considers the stochastic case. Consequently, their algorithms are not applicable here, necessitating the design of efficient stochastic approaches for (27). By replacing Δ_m in (10) with $\mathcal{S}_{m,k}$, we obtain the performance measure of an approximate solution $(\bar{\mathbf{w}}, \bar{\mathbf{q}})$ to (27), i.e., $\epsilon'_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) = \max_{\mathbf{q}\in\mathcal{S}_{m,k}} \phi(\bar{\mathbf{w}}, \mathbf{q}) - \min_{\mathbf{w}\in\mathcal{W}} \phi(\mathbf{w}, \bar{\mathbf{q}})$, which controls the optimality of $\bar{\mathbf{w}}$ to (26).

Algorithm 3 Stochastic Mirror Descent for AT$_k$RO

Input: step sizes η_w and η_q

1: Initialize $\mathbf{w}_1$ and $\mathbf{q}_1$ according to (17)
2: **for** $t = 1$ to T **do**
3: For each $i \in [m]$, draw a sample $\mathbf{z}_t^{(i)}$ from $\mathcal{P}_i$
4: Construct the stochastic gradients defined in (12)
5: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ via (14) and (30), respectively
6: **end for**
7: **return** $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$ and $\bar{\mathbf{q}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{q}_t$

4.2 Stochastic Mirror Descent for AT$_k$RO

Similar to Section 3.2, we also use SMD to optimize (27), with the only difference being the update rule for $\mathbf{q}$.

Since the objectives of (27) and (4) are identical, the stochastic gradients $\mathbf{g}_w(\mathbf{w}_t, \mathbf{q}_t)$ and $\mathbf{g}_q(\mathbf{w}_t, \mathbf{q}_t)$ in (12) also serve as unbiased estimators of true gradients $\nabla_{\mathbf{w}}\phi(\mathbf{w}_t, \mathbf{q}_t)$ and $\nabla_{\mathbf{q}}\phi(\mathbf{w}_t, \mathbf{q}_t)$, respectively. In the t-th round, we reuse (14) to update $\mathbf{w}_t$, and modify the update of $\mathbf{q}_t$ as

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} \{\eta_q\langle -\mathbf{g}_q(\mathbf{w}_t, \mathbf{q}_t), \mathbf{q}\rangle + B_q(\mathbf{q}, \mathbf{q}_t)\}. \quad (28)$$

Since the domain is not the simplex Δ_m, the explicit form in (16) does not apply to (28). By defining $B_q(\mathbf{q}, \mathbf{q}_t)$ in terms of the negative entropy, we employ the following lemma to reformulate (28) as a Bregman projection problem [57].

Lemma 1. *Consider a mirror descent defined as*

$$\mathbf{q} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} \{\eta\langle \mathbf{g}, \mathbf{q}\rangle + B_q(\mathbf{q}, \mathbf{q}_0)\}, \quad (29)$$

where $\mathbf{g}, \mathbf{q}_0 \in \mathbb{R}^m$ and $B_q(\cdot,\cdot)$ is the Bregman distance induced by the neg-entropy. Then, (29) is equivalent to $\mathbf{q} = \operatorname{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} B_q(\mathbf{q}, \hat{\mathbf{q}})$ with $\hat{q}_i = q_{0,i}e^{-\eta g_i}$ for all $i \in [m]$.

By Lemma 1, problem (28) reduces to

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} B_q(\mathbf{q}, \hat{\mathbf{q}}_t), \quad (30)$$

where $\hat{q}_{t,i} = q_{t,i}\exp(\eta_q \ell(\mathbf{w}_t; \mathbf{z}_t^{(i)}))$ for all $i \in [m]$. To solve (30), we employ the method of Salem et al. [57], summarized in Appendix H.1, with a time complexity of $O(m + k\ln k)$.

We present the entire procedure in Algorithm 3, and state the following theorem.

Theorem 5. *Under Assumptions 1-4, and setting $\eta_w = D^2\sqrt{\frac{8}{5T(D^2G^2+\ln\frac{m}{k})}}$ and $\eta_q = (\ln\frac{m}{k})\sqrt{\frac{8}{5T(D^2G^2+\ln\frac{m}{k})}}$ in Algorithm 3, we have*

$$\mathrm{E}\big[\epsilon'_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}})\big] \le 2\sqrt{\frac{10(D^2G^2 + \ln\frac{m}{k})}{T}},$$

and with probability at least $1-\delta$,

$$\epsilon'_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) \le \left(8 + 2\ln\frac{2}{\delta}\right)\sqrt{\frac{10(D^2G^2 + \ln\frac{m}{k})}{T}}.$$

Remark 9. The above theorem indicates that Algorithm 3 attains an $O(\sqrt{(\log(m/k))/T})$ convergence rate. Since it requires m samples in each iteration, the sample complexity is $O(m(\log(m/k))/\epsilon^2)$.

4.2.1 Anytime Extension of Algorithm 3

We adapt Algorithm 3 to an anytime version by modifying η_w in (14) and η_q in (28)/(30) to be time-varying step sizes

$$\eta_t^w = D^2\sqrt{\frac{2}{C't}}, \text{ and } \eta_t^q = \left(\ln\frac{m}{k}\right)\sqrt{\frac{2}{C't}}, \quad (31)$$

where $C' = D^2G^2 + \ln\frac{m}{k}$. When required, we return ($\bar{\mathbf{w}}_t$, $\bar{\mathbf{q}}_t$) as the output with the following theoretical guarantee.

Theorem 6. *Under Assumptions 1-4, and modifying step sizes as* (31) *in Algorithm 3, we have for all $t \in \mathbb{Z}_+$,*

$$\mathrm{E}\big[\epsilon'_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)\big] \le \frac{\sqrt{D^2G^2 + \ln\frac{m}{k}}}{\sqrt{2}\left(\sqrt{t+1}-1\right)}\left(5 + 3\ln t\right),$$

and for each $t \in \mathbb{Z}_+$, with probability at least $1-\delta$,

$$\epsilon'_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t) = O\left(\frac{\sqrt{1+\log\frac{m}{k}}}{\sqrt{t}}\left(\log t \cdot \log\frac{1}{\delta}\right)\right).$$

Remark 10. Similar to previous cases, the convergence rate of the anytime extension is slower by a factor of $O(\log t)$.

4.3 Non-oblivious Online Learning for AT$_k$RO

Building on the two-player game in Section 3.3, we can leverage online learning techniques to reduce the number of samples required in each round for AT$_k$RO from m to k.

The 1st player faces the same problem, specifically minimizing the sequence of convex functions $\{\phi(\mathbf{w}, \mathbf{q}_t) = \sum_{i=1}^m q_{t,i}R_i(\mathbf{w})\}_{t\in[T]}$ under the constraint $\mathbf{w} \in \mathcal{W}$. Therefore, it can still be framed as "non-oblivious OCO with stochastic gradients" and solved using SMD. In contrast, the 2nd player tackles a different challenge: maximizing the sequence of linear functions $\{\phi(\mathbf{w}_t, \mathbf{q}) = \sum_{i=1}^m q_i R_i(\mathbf{w}_t)\}_{t\in[T]}$, constrained by $\mathbf{q} \in \mathcal{S}_{m,k}$ rather than $\mathbf{q} \in \Delta_m$. Since the domain is the capped simplex, it is natural to ask the 2nd player to select the k highest-risk distributions from m distributions, reflecting the combinatorial nature of the problem. After drawing one sample from each selected distribution, the 2nd player observes k stochastic rewards, which fits into a semi-bandit structure. This leads to modeling the 2nd player's problem as "non-oblivious combinatorial semi-bandits with stochastic rewards". For the 2nd player, we can certainly apply existing algorithms designed for non-oblivious combinatorial semi-bandits [58], [59], [60]. Here, to maintain consistency with Algorithm 2, we extend the Exp3-IX algorithm to address this scenario.

In the following, we detail the modifications relative to Algorithm 2. At each round t, we select k indices from $[m]$ according to $\mathbf{q}_t$, generating a set $\mathcal{I}_t$ that satisfies

$$|\mathcal{I}_t| = k, \text{ and } \Pr[i \in \mathcal{I}_t] = kq_{t,i}, \ \forall i \in [m]. \quad (32)$$

To this end, we use the DepRound algorithm [61] with $(k, \mathbf{q}_t)$ as inputs to generate $\mathcal{I}_t$, which meets the requirement (32) with $O(m)$ time and space complexities. A detailed description is provided in Appendix H.2. We note that DepRound has been applied in various combinatorial semi-bandit algorithms [51], [60], [62]. Once $\mathcal{I}_t$ is obtained, we draw one sample $\mathbf{z}_t^{(i)}$ from distribution $\mathcal{P}_i$ for all $i \in \mathcal{I}_t$.

Algorithm 4 Non-oblivious Online Learning for AT$_k$RO

Input: step sizes η_w and η_q, and IX coefficient γ

1: Initialize $\mathbf{w}_1$ and $\mathbf{q}_1$ according to (17)
2: **for** $t = 1$ to T **do**
3: Generate $\mathcal{I}_t = \text{DepRound}(k, \mathbf{q}_t)$
4: For each $i \in \mathcal{I}_t$, draw a sample $\mathbf{z}_t^{(i)}$ from $\mathcal{P}_i$
5: Construct the stochastic gradient in (33) and the modified IX loss estimator in (34)
6: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ via (22) and (36), respectively
7: **end for**
8: **return** $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$ and $\bar{\mathbf{q}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{q}_t$

Next, the 1st player constructs the stochastic gradient as:

$$\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t) = \frac{1}{k}\sum_{i\in\mathcal{I}_t}\nabla\ell(\mathbf{w}_t; \mathbf{z}_t^{(i)}), \tag{33}$$

which is an unbiased estimator of $\nabla_{\mathbf{w}}\phi(\mathbf{w}_t, \mathbf{q}_t)$ by (32). Then, we update $\mathbf{w}_t$ by applying the mirror descent (22) with $\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)$ in (33). For the 2nd player, we modify the IX loss estimator for the combinatorial semi-bandit setting:

$$\tilde{s}_{t,i} = \frac{1 - \ell(\mathbf{w}_t, \mathbf{z}_t^{(i)})}{kq_{t,i} + \gamma} \cdot \mathbb{I}[i \in \mathcal{I}_t],\ \forall i \in [m] \tag{34}$$

and then update $\mathbf{q}_t$ by mirror descent

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} \left\{\eta_q\langle\tilde{\mathbf{s}}_t, \mathbf{q}\rangle + B_q(\mathbf{q}, \mathbf{q}_t)\right\}. \tag{35}$$

Compared with (23), (34) incorporates two key changes. First, we replace $\mathbb{I}[i_t = i]$ with $\mathbb{I}[i \in \mathcal{I}_t]$ to utilize all the k observed losses $\{\ell(\mathbf{w}_t, \mathbf{z}_t^{(i)}) | i \in \mathcal{I}_t\}$. Second, since $\Pr[i \in \mathcal{I}_t] = kq_{t,i}$, the denominator of $\tilde{s}_{t,i}$ is adjusted accordingly. By Lemma 1, we similarly transform (35) as:

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} B_q(\mathbf{q}, \hat{\mathbf{q}}_t), \tag{36}$$

where $\hat{q}_{t,i} = q_{t,i}e^{-\eta_q\tilde{s}_{t,i}}$ and solve (36) using the method of Salem et al. [57]. The complete procedure is presented in Algorithm 4.

In the theoretical analysis, the regret bound for the 1st player in Theorem 3 remains applicable, since the only change $\|\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)\|_{w,*}^2 = \|\frac{1}{k}\sum_{i\in\mathcal{I}_t}\nabla\ell(\mathbf{w}_t; \mathbf{z}_t^{(i)})\|_{w,*}^2 \le G^2$ does not alter the conclusion. For the 2nd player, we derive a new regret bound. By combining these results, we obtain the optimization error bound for Algorithm 4 as follows.

Theorem 7. *Under Assumptions 1-4, and setting* $\eta_w = \frac{2D}{G\sqrt{5T}}$, $\eta_q = \sqrt{\frac{k\ln m}{mT}}$ *and* $\gamma = \frac{\eta_q}{2}$ *in Algorithm 4, we have*

$$\mathrm{E}\left[\epsilon'_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}})\right] = O\left(\sqrt{\frac{m\log m}{kT}}\right),$$

and with probability at least $1 - \delta$,

$$\epsilon'_\phi(\bar{\mathbf{w}}, \bar{\mathbf{q}}) = O\left(\sqrt{\frac{m\log m}{kT}} + \sqrt{\frac{m}{kT\log m}}\log\frac{1}{\delta}\right).$$

Remark 11. Theorem 7 demonstrates that Algorithm 4 obtains an $O(\sqrt{m(\log m)/(kT)})$ convergence rate. Since it consumes k samples per iteration, the sample complexity is $O(m(\log m)/\epsilon^2)$, slightly higher than that of Algorithm 3.

Algorithm 5 Non-oblivious Online Learning for AT$_k$RO with Anytime Capability

1: Initialize $\mathbf{w}_1$ and $\mathbf{q}_1$ according to (17)
2: **for** $t = 1, 2, \cdots$ **do**
3: Generate $i_t \in [m]$ according to $\mathbf{q}_t$, and draw a sample $\mathbf{z}_t^{(i_t)}$ from distribution $\mathcal{P}_{i_t}$
4: Construct the stochastic gradient in (13) and the IX loss estimator in (38)
5: Update $\mathbf{w}_t$ and $\mathbf{q}_t$ via (37) and (39), respectively
6: **end for**

4.3.1 Anytime Extension of Algorithm 4

Following Section 3.3.1, it is natural to adopt time-varying parameters to make Algorithm 4 anytime. While the analysis for the 1st player is straightforward, that for the 2nd player presents a technical challenge. Neu [18] established two concentration results for the IX loss estimator (23): Corollary 1 in his paper for fixed parameters and Lemma 1 for time-varying parameters. In Section 4.3, we extend Neu's Corollary 1 to combinatorial semi-bandits, yielding Theorem 13 in Appendix E.7. However, we are unable to extend his Lemma 1 to combinatorial semi-bandits. Additionally, we have not found any algorithms in the literature that use time-varying parameters in non-oblivious combinatorial semi-bandits.

To circumvent the aforementioned challenge, we propose an anytime algorithm for AT$_k$RO from a different perspective. The key observation is that *we are not dealing with a true bandit problem but are instead exploiting bandit techniques to solve* (27). In our algorithm, the 2nd player is not required to select exactly k distinct arms. It is perfectly fine to select just 1 arm, as long as we can bound the regret in terms of the linear functions $\{\phi(\mathbf{w}_t, \mathbf{q})\}_{t\in[T]}$, subject to the constraint $\mathbf{q} \in \mathcal{S}_{m,k}$. To this end, we propose to modify the anytime extension of Algorithm 2 proposed in Section 3.3.1.

We describe the key steps as follows. Recall the time-varying parameters η_t^w, η_t^q, and γ_t in (25). In each round, we update $\mathbf{w}_t$ via SMD in (22) with a time-varying step size:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \left\{\eta_t^w\langle\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t), \mathbf{w}\rangle + B_w(\mathbf{w}, \mathbf{w}_t)\right\}, \tag{37}$$

where $\tilde{\mathbf{g}}_w(\mathbf{w}_t, \mathbf{q}_t)$ is defined in (13). Similarly, we use a time-varying parameter to construct the IX loss estimator

$$\tilde{s}_{t,i} = \frac{1 - \ell(\mathbf{w}_t, \mathbf{z}_t^{(i_t)})}{q_{t,i} + \gamma_t} \cdot \mathbb{I}[i_t = i],\ \forall i \in [m]. \tag{38}$$

The only change required is to adjust the domain in the mirror descent (24) to $\mathcal{S}_{m,k}$:

$$\mathbf{q}_{t+1} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{S}_{m,k}} \left\{\eta_t^q\langle\tilde{\mathbf{s}}_t, \mathbf{q}\rangle + B_q(\mathbf{q}, \mathbf{q}_t)\right\}, \tag{39}$$

which can also be reduced to a neg-entropy Bregman projection problem. If demanded, we return $(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)$ in (21) as the solution. The full procedure is summarized in Algorithm 5.

Following Theorem 4, we establish the theoretical guarantee for Algorithm 5 shown below.

Theorem 8. *Under Assumptions 1-4, and setting parameters as* (25) *in Algorithm 5, we have for all* $t \in \mathbb{Z}_+$,

$$\mathrm{E}\left[\epsilon'_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t)\right] = O\left(\frac{\sqrt{m\log m}\log t}{\sqrt{t}}\right),$$

TABLE 1
Notation for Algorithms.

Algorithms	Notation	Highlights
Alg. 1 of [12]	SMD(1)	SMD with 1 sample per round
Alg. 1	SMD(m)	SMD with m samples per round
Anytime Alg. 1	SMD(m)$_{\mathrm{a}}$	SMD(m) with time-varying step sizes
Alg. 2	Online(1)	Online learning method with 1 sample per round
Anytime Alg. 2	Online(1)$_{\mathrm{a}}$	Online(1) with time-varying step sizes
Alg. 3	AT$_k$RO(m)	SMD(m) for AT$_k$RO
Anytime Alg. 3	AT$_k$RO(m)$_{\mathrm{a}}$	SMD(m) with time-varying step sizes for AT$_k$RO
Alg. 4	AT$_k$RO(k)	Online learning method with k samples per round for AT$_k$RO
Alg. 5	AT$_k$RO(1)$_{\mathrm{a}}$	Anytime online method with 1 sample per round for AT$_k$RO

and for each $t \in \mathbb{Z}_+$, with probability at least $1-\delta$,

$$\epsilon'_\phi(\bar{\mathbf{w}}_t, \bar{\mathbf{q}}_t) = O\left(\frac{\sqrt{m\log m}\log t + \left(\sqrt{m/\log m} + \sqrt{\log t}\right)\log\frac{1}{\delta}}{\sqrt{t}}\right).$$

Remark 12. Since Algorithm 5 uses only 1 sample per iteration, it is not surprising that its convergence rate is slower than Algorithm 4 by a factor of $\widetilde{O}(\sqrt{k})$.

5 Experiments

We evaluate the effectiveness of the proposed algorithms through experiments.

5.1 Data Sets and Experimental Settings

We evaluate methods on two synthetic data sets, the Adult data set [63] and MNIST [64]. The two synthetic data sets both contain $m = 20$ groups: one is homogeneous for comparing GDRO algorithms, and the other is heterogeneous for evaluating AT$_k$RO algorithms. The Adult data set is partitioned into $m = 6$ groups according to race and gender. For these three data sets, we train linear models with logistic loss. Although our theoretical guarantees rely on convexity, we further evaluate the proposed algorithms in a non-convex setting on MNIST. Specifically, we construct $m = 10$ label-dominated groups and train a two-layer CNN followed by a linear classifier with the cross-entropy loss. To estimate the risk $R_i(\cdot)$, we approximate the expectation over $\mathcal{P}_i$ by empirical averages over samples from the corresponding group. All results are averaged over five independent runs with different random seeds, and shaded regions indicate standard deviations. More details are deferred to Appendix C and the algorithm notation is summarized in Table 1.

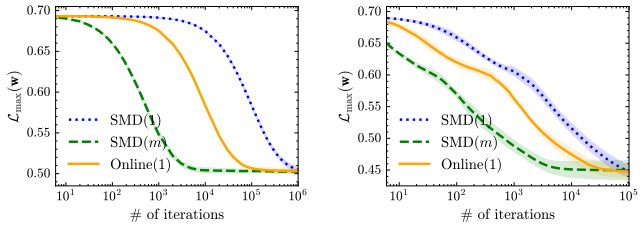

Fig. 2. Homogeneous settings: maximum risk versus the number of iterations. Left: Synthetic data set. Right: Adult data set.

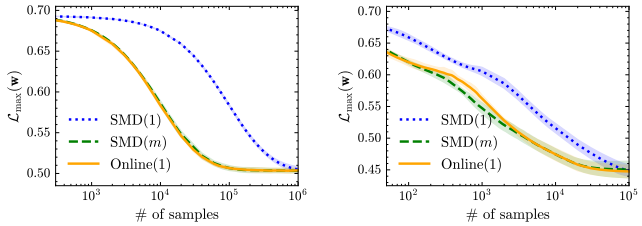

Fig. 3. Homogeneous settings: maximum risk versus the number of samples. Left: Synthetic data set. Right: Adult data set.

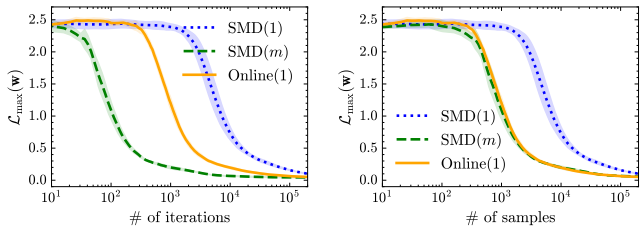

Fig. 4. Homogeneous settings with MNIST. Left: $\mathcal{L}_{\max}(\mathbf{w})$ versus the number of iterations. Right: $\mathcal{L}_{\max}(\mathbf{w})$ versus the number of samples.

5.2 GDRO on Homogeneous Distributions

For experiments on the synthetic data sets, samples are generated on the fly following the protocol in Appendix C. For those on the Adult data set and MNIST, samples are drawn with replacement from each group, i.e., $\mathcal{P}_i$ is the empirical distribution over the data in the i-th group.

We compare SMD(1) with our algorithms SMD(m) and Online(1). Fig. 2 plots the maximum risk $\mathcal{L}_{\max}(\mathbf{w})$ with respect to the number of iterations. We observe that SMD(m) is faster than Online(1), which in turn outperforms SMD(1). This observation is consistent with our theories, since their convergence rates are $O(\sqrt{(\log m)/T})$, $O(\sqrt{m(\log m)/T})$, and $O(m\sqrt{(\log m)/T})$, respectively. Fig. 3 shows $\mathcal{L}_{\max}(\mathbf{w})$ versus the number of samples. As can be seen, the curves of SMD(m) and Online(1) are very close, indicating that they share the same sample complexity, i.e., $O(m(\log m)/\epsilon^2)$. On the other hand, SMD(1) needs more samples to reach a target precision, which aligns with its higher sample complexity, i.e., $O(m^2(\log m)/\epsilon^2)$. We further conduct experiments with a CNN on MNIST to examine the performance of the algorithms in a non-convex setting. As shown in Fig. 4, the results exhibit the same empirical pattern as in the convex experiments. This suggests the effectiveness of the algorithms beyond the convex setting, as well as the empirical sample-efficiency advantage of SMD(m) and Online(1) over SMD(1).

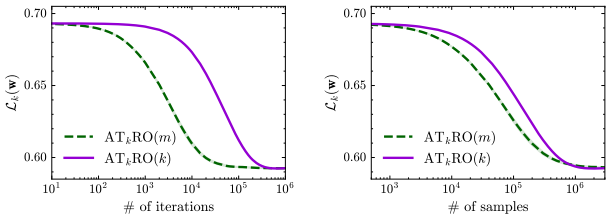

Fig. 5. Heterogeneous settings with the synthetic data set. $\mathcal{L}_k(\mathbf{w})$ versus (a) the number of iterations and (b) the number of samples.

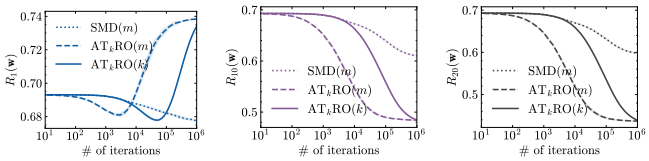

Fig. 6. Heterogeneous settings with the synthetic data set: individual risks versus the number of iterations. Left: $\mathcal{P}_1$. Middle: $\mathcal{P}_{10}$. Right: $\mathcal{P}_{20}$.

5.3 AT$_k$RO on Heterogeneous Distributions

For experiments on heterogeneous distributions, we use the second synthetic data set, where $\mathcal{P}_1$ is constructed as an outlier. We first compare AT$_k$RO(m) and AT$_k$RO(k) with $k = 3$, and plot the changes of the average top-k risk $\mathcal{L}_k(\mathbf{w})$ in Fig. 5. Theorems 5 and 7 show that their convergence rates are $O(\sqrt{(\log(m/k))/T})$ and $O(\sqrt{m(\log m)/(kT)})$, and the sample complexities are $O(m(\log(m/k))/\epsilon^2)$ and $O(m(\log m)/\epsilon^2)$, respectively. Consistent with these results, Fig. 5(a) indicates that AT$_k$RO(m) converges faster than AT$_k$RO(k), and Fig. 5(b) shows that AT$_k$RO(m) requires slightly fewer samples to achieve the same objective value.

Additionally, to demonstrate the advantages of AT$_k$RO, we examine the performance of directly applying SMD(m), which is designed for GDRO, to the synthetic data set. Fig. 6 presents the changes in risk across a subset of distributions for SMD(m), AT$_k$RO(m) and AT$_k$RO(k). We observe that SMD(m) concentrates entirely on the outlier distribution $\mathcal{P}_1$ and achieves the lowest final risk on $\mathcal{P}_1$, approximately 0.061 lower than AT$_k$RO(m) and 0.056 lower than AT$_k$RO(k). However, for the remaining 19 distributions $\{\mathcal{P}_2, \ldots, \mathcal{P}_{20}\}$, the risk of SMD($m$) is approximately 0.126 higher on average than those of the other two algorithms. Therefore, we conclude that AT$_k$RO can mitigate the impact of the outlier distribution and deliver a more balanced model than GDRO in heterogeneous distributions.

5.4 Anytime Capability

To demonstrate the benefits of the anytime capability, we compare SMD(m) and Online(1) with their anytime extensions SMD(m)$_\mathrm{a}$ and Online(1)$_\mathrm{a}$ on the Adult data set, and AT$_k$RO(m) and AT$_k$RO(k) with AT$_k$RO(m)$_\mathrm{a}$ and AT$_k$RO(1)$_\mathrm{a}$ on the second synthetic data set, where $k = 3$.

We assign a preset value of $T = 2000$ for SMD(m) and Online(1), and $T = 50000$ for AT$_k$RO(m) and AT$_k$RO(k). When the actual number of iterations exceeds the preset number T, we continue running the four algorithms with the initial parameters. As illustrated in Fig. 7, non-anytime

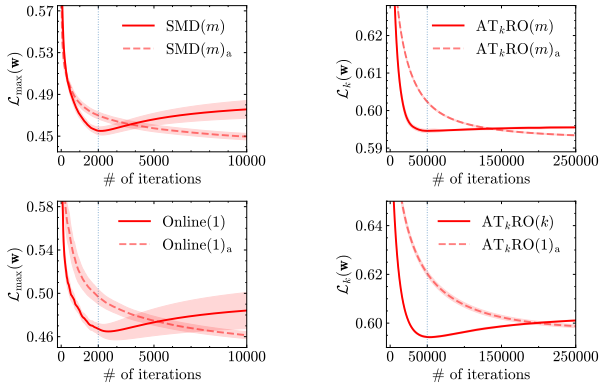

Fig. 7. The performance of different methods versus the number of iterations. Blue dashed lines indicate the predetermined T for non-anytime algorithms. Left: Adult data set. Right: Synthetic data set.

algorithms initially reduce the objective (the maximum risk or the average top-k risk) more rapidly than anytime algorithms before reaching the predetermined T, where they achieve minimal values. However, as the number of iterations increases, their curves plateau or even increase due to sub-optimal parameters. In contrast, the anytime algorithms, with time-varying step sizes, consistently reduce their objectives over time, eventually falling below the value attained by the corresponding non-anytime algorithms.

6 Conclusion

For the GDRO problem, we develop two SA approaches: one employs SMD with m samples per round and the other combines SMD with an algorithm for non-oblivious MAB using 1 sample in each iteration. Both achieve a near-optimal sample complexity of $O(m(\log m)/\epsilon^2)$. Then, we consider heterogeneous distributions involving high-risk outliers. In this scenario, we formulate the AT$_k$RO problem and propose two algorithms: one applies SMD with m samples per round, attaining a sample complexity of $O(m(\log(m/k))/\epsilon^2)$; the other combines SMD with an algorithm for non-oblivious combinatorial semi-bandits, using k samples per round and achieving an $O(m(\log m)/\epsilon^2)$ sample complexity. We further develop anytime SA algorithms for both GDRO and AT$_k$RO.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer, 2000.
[2] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, 2nd ed. Springer, 2003.
[3] A. Ben-Tal, D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.
[4] A. Shapiro, "Distributionally robust stochastic programming," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2258–2275, 2017.
[5] J. C. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *The Annals of Statistics*, vol. 49, no. 3, pp. 1378 – 1406, 2021.
[6] J. C. Duchi, P. W. Glynn, and H. Namkoong, "Statistics of robust optimization: A generalized empirical likelihood approach," *Mathematics of Operations Research*, vol. 46, no. 3, pp. 946–969, 2021.

[7] W. Hu, G. Niu, I. Sato, and M. Sugiyama, "Does distributionally robust supervised learning give robust classifiers?" in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2029–2037.

[8] S. Curi, K. Y. Levy, S. Jegelka, and A. Krause, "Adaptive sampling for stochastic risk-averse learning," in *Advances in Neural Information Processing Systems 33*, 2020, pp. 1036–1047.

[9] J. Jin, B. Zhang, H. Wang, and L. Wang, "Non-convex distributionally robust optimization: Non-asymptotic analysis," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 2771–2782.

[10] A. Agarwal and T. Zhang, "Minimax regret optimization for robust machine learning under distribution shift," in *Proceedings of 35th Conference on Learning Theory*, 2022, pp. 2704–2729.

[11] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang, "Distributionally robust language modeling," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 4227–4237.

[12] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," in *International Conference on Learning Representations*, 2020.

[13] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 4615–4625.

[14] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[15] N. Haghtalab, M. I. Jordan, and E. Zhao, "On-demand sampling: Learning optimally from multiple distributions," in *Advances in Neural Information Processing Systems 35*, 2022, pp. 406–419.

[16] T. Soma, K. Gatmiry, and S. Jegelka, "Optimal algorithms for group distributionally robust optimization and beyond," *ArXiv e-prints*, vol. arXiv:2212.13669, 2022.

[17] Y. Carmon and D. Hausler, "Distributionally robust optimization via ball oracle acceleration," in *Advances in Neural Information Processing Systems 35*, 2022, pp. 35 866–35 879.

[18] G. Neu, "Explore no more: Improved high-probability regret bounds for non-stochastic bandits," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3168–3176.

[19] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 1544–1551.

[20] S. Shalev-Shwartz and Y. Wexler, "Minimizing the maximal loss: How and why," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 793–801.

[21] Y. Fan, S. Lyu, Y. Ying, and B. Hu, "Learning with average top-k loss," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 497–505.

[22] L. Zhang, P. Zhao, Z. Zhuang, T. Yang, and Z.-H. Zhou, "Stochastic approximation approaches to group distributionally robust optimization," in *Advances in Neural Information Processing Systems 37*, 2023, pp. 52 490–52 522.

[23] H. Scarf, "A min-max solution of an inventory problem," *Studies in the Mathematical Theory of Inventory and Production*, pp. 201–209, 1958.

[24] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009.

[25] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor, "Oracle-based robust optimization via online learning," *Operations Research*, vol. 63, no. 3, pp. 628–638, 2015.

[26] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," in *International Conference on Learning Representations*, 2018.

[27] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness without demographics in repeated loss minimization," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1929–1938.

[28] Z. Xu, C. Dan, J. Khim, and P. Ravikumar, "Class-weighted classification: Trade-offs and robust approaches," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 10 544–10 554.

[29] D. Samuel and G. Chechik, "Distributional robustness loss for long-tail learning," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9475–9484.

[30] J. Zhang, A. K. Menon, A. Veit, S. Bhojanapalli, S. Kumar, and S. Sra, "Coping with label shift via distributionally robust optimisation," in *International Conference on Learning Representations*, 2021.

[31] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.

[32] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," *Operations Research & Management Science in the Age of Analytics*, pp. 130–166, 2019.

[33] J. Wang, R. Gao, and Y. Xie, "Sinkhorn distributionally robust optimization," *ArXiv e-prints*, vol. arXiv:2109.11926, 2021.

[34] D. Bertsimas, V. Gupta, and N. Kallus, "Robust sample average approximation," *Mathematical Programming*, vol. 171, pp. 217–282, 2018.

[35] Q. M. Nguyen, N. A. Mehta, and C. A. Guzmán, "Beyond minimax rates in group distributionally robust optimization via a novel notion of sparsity," in *Proceedings of the 42nd International Conference on Machine Learning*, 2025, pp. 46 073–46 115.

[36] A. Słowik and L. Bottou, "On distributionally robust optimization and data rebalancing," in *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022, pp. 1283–1297.

[37] L. Zhang, H. Bai, W.-W. Tu, P. Yang, and Y. Hu, "Efficient stochastic approximation of minimax excess risk optimization," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 58 599–58 630.

[38] D. Yu, Y. Cai, W. Jiang, and L. Zhang, "Efficient algorithms for empirical group distributionally robust optimization and beyond," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 57 384–57 414.

[39] A. Blum, N. Haghtalab, A. D. Procaccia, and M. Qiao, "Collaborative PAC learning," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2389–2398.

[40] H. L. Nguyen and L. Zakynthinou, "Improved algorithms for collaborative PAC learning," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 7642–7650.

[41] G. N. Rothblum and G. Yona, "Multi-group agnostic PAC learnability," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 9107–9115.

[42] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1180–1189.

[43] M. Zhu, J. Li, N. Wang, and X. Gao, "A deep collaborative framework for face photo–sketch synthesis," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 10, pp. 3096–3108, 2019.

[44] ——, "Learning deep patch representation for probabilistic graphical model-based face sketch synthesis," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1820–1836, 2021.

[45] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

[46] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[47] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI Magazine*, vol. 17, no. 3, pp. 73–83, 1996.

[48] A. Cutkosky, "Anytime online-to-batch, optimism and acceleration," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 1446–1454.

[49] Y. Freund and R. E. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1, pp. 79–103, 1999.

[50] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire, "Fast convergence of regularized learning in games," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2989–2997.

[51] C. Roux, E. Wirth, S. Pokutta, and T. Kerdreux, "Efficient online-bandit strategies for minimax learning problems," *ArXiv e-prints*, vol. arXiv:2105.13939, 2021.

[52] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[53] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.

[54] T. Kocák, G. Neu, M. Valko, and R. Munos, "Efficient learning by implicit exploration in bandit problems with side observations,"

in *Advances in Neural Information Processing Systems 27*, 2014, pp. 613–621.
[55] F. Orabona, "A modern introduction to online learning," *ArXiv e-prints*, vol. arXiv:1912.13213v6, 2019.
[56] N. Haghtalab, M. I. Jordan, and E. Zhao, "On-demand sampling: Learning optimally from multiple distributions," *ArXiv e-prints*, vol. arXiv:2210.12529v2, 2023.
[57] T. Si Salem, G. Neglia, and S. Ioannidis, "No-regret caching via online mirror descent," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 8, no. 4, 2023.
[58] J.-Y. Audibert, S. Bubeck, and G. Lugosi, "Regret in online combinatorial optimization," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 31–45, 2014.
[59] G. Neu and G. Bartók, "Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits," *Journal of Machine Learning Research*, vol. 17, no. 154, pp. 1–21, 2016.
[60] N. M. Vural, H. Gokcesu, K. Gokcesu, and S. S. Kozat, "Minimax optimal algorithms for adversarial bandit problem with multiple plays," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4383–4398, 2019.
[61] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.
[62] T. Uchiya, A. Nakamura, and M. Kudo, "Algorithms for adversarial bandit problems with multiple plays," in *Algorithmic Learning Theory*, 2010, pp. 375–389.
[63] B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5XW20.
[64] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.
[65] H. Namkoong and J. C. Duchi, "Variance-based regularization with convex objectives," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2971–2980.
[66] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, pp. 115–166, 2018.
[67] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 2216–2224.
[68] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford, "Large-scale methods for distributionally robust optimization," in *Advances in Neural Information Processing Systems 33*, 2020, pp. 8847–8860.
[69] Q. Qi, Z. Guo, Y. Xu, R. Jin, and T. Yang, "An online method for a class of distributionally robust optimization with non-convex objectives," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 10 067–10 080.
[70] H. Rafique, M. Liu, Q. Lin, and T. Yang, "Weakly-convex-concave min-max optimization: Provable algorithms and applications in machine learning," *Optimization Methods and Software*, vol. 37, no. 3, pp. 1087–1121, 2022.
[71] R. J. Chen, J. J. Wang, D. F. K. Williamson, T. Y. Chen, J. Lipkova, M. Y. Lu, S. Sahai, and F. Mahmood, "Algorithmic fairness in artificial intelligence for medicine and healthcare," *Nature Biomedical Engineering*, vol. 7, pp. 719–742, 2023.
[72] Y. Bao, S. Chang, and R. Barzilay, "Predict then interpolate: A simple algorithm to learn stable classifiers," in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 640–650.
[73] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.
[74] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005, pp. 385–394.
[75] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Lijun Zhang (Senior Member, IEEE) received the B.E. and Ph.D. degrees in Software Engineering and Computer Science from Zhejiang University, China, in 2007 and 2012, respectively. He is currently a Professor of the School of Artificial Intelligence, Nanjing University, China. Prior to joining Nanjing University, he was a postdoctoral researcher at the Department of Computer Science and Engineering, Michigan State University, USA. His research interests include machine learning and optimization.

Haomin Bai received the B.Sc. degree from the School of Mathematics, Southeast University, China, in 2023. He is currently working toward the Ph.D. degree with the School of Artificial Intelligence, Nanjing University, China. His research interests include machine learning and optimization.

Peng Zhao (Member, IEEE) received his B.Sc. degree from Tongji University in 2016 and Ph.D. degree from Nanjing University in 2021.

He is currently an assistant professor at the School of Artificial Intelligence in Nanjing University. His research interests lie in the foundations of machine learning, with a focus on online learning, reinforcement learning, and optimization. He has published over 60 papers in top-tier journals like JMLR and IEEE/ACM Trans, as well as premier conferences including ICML, NeurIPS, and COLT. He regularly serves as an Area Chair for ICML and NeurIPS.

Zhi-Hua Zhou (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees (Hons.) in computer science from Nanjing University, Nanjing, China, in 1996, 1998, and 2000, respectively, all with the highest honor. He joined Nanjing University as an Assistant Professor in 2001, where he is currently a Professor and the Vice President. He is the Founding Director of the LAMDA Group. His research interests are mainly in artificial intelligence, machine learning, and data mining. He has authored the books Ensemble Methods: Foundations and Algorithms, Machine Learning, and has published more than 200 papers in top-tier international journals or conference proceedings. He holds more than 40 patents. He received various awards/honors, including the National Natural Science Award of China, the IEEE Computer Society Edward J. McCluskey Technical Achievement Award, the CCF-ACM Artificial Intelligence Award, etc. He founded Asian Conference on Machine Learning (ACML). He is the President of IJCAI Trustee, a Series Editor of Springer Lecture Notes in Artificial Intelligence, an Advisory Board Member of AI Magazine, the Editor-in-Chief of Frontiers of Computer Science. He is a member of the Chinese Academy of Sciences and Academia Europaea, and Fellow of of ACM, AAAI, AAAS, etc.