Efficient Mirror Descent Ascent Methods for Nonsmooth Minimax Problems

Anonymous Author(s)
Affiliation
Address
email

Abstract

In the paper, we propose a class of novel and efficient mirror descent ascent methods to solve the nonsmooth nonconvex-strongly-concave minimax problems by using dynamic mirror functions, and introduce a convergence analysis framework to conduct rigorous theoretical analysis for our mirror descent ascent methods. For our stochastic algorithms, we first prove that the mini-batch stochastic mirror descent ascent (SMDA) method obtains a sample complexity of $O(\kappa^3\epsilon^{-4})$ for finding an ϵ-stationary point, where κ denotes the condition number. Further, we propose an accelerated stochastic mirror descent ascent (VR-SMDA) method based on the variance reduced technique. We prove that our VR-SMDA method achieves a lower sample complexity of $O(\kappa^3\epsilon^{-3})$. For our deterministic algorithm, we prove that our deterministic mirror descent ascent (MDA) achieves a lower sample complexity of $O(\kappa\epsilon^{-2})$ under mild conditions, which improves the best known complexity by a factor of $O(\sqrt{\kappa})$. We conduct the experiments on fair classifier and robust neural network training tasks to demonstrate the efficiency of our new algorithms.

1 Introduction

Minimax optimization recently has attracted increased interest largely due to advances in machine learning application such as generative adversarial networks (GANs) [9, 27], robust neural networks training [22], fair learning [21], federated learning [6], and policy evaluation [29]. In the paper, we study the following nonsmooth nonconvex-strongly-concave minimax problem:

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} F(x,y) = \big\{f(x,y) + g(x) - h(y)\big\}, \tag{1}$$

where the function $f(x,y):\mathbb{R}^d\times\mathbb{R}^p\to\mathbb{R}$ is smooth nonconvex in $x\in\mathbb{R}^d$ and μ-strongly concave in $y\in\mathbb{R}^p$, and the functions $g(x)$ and $h(y)$ are convex and possibly nonsmooth. Here the constraint sets $\mathcal{X}\subseteq\mathbb{R}^d$ and $\mathcal{Y}\subseteq\mathbb{R}^p$ are compact and convex. In many machine learning problems, $f(x,y)$ generally represents loss function and is a stochastic form, *i.e.*, $f(x,y)=\mathbb{E}_\xi[f(x,y;\xi)]$, where the random variable ξ follows an unknown distribution. Here both $g(x)$ and $f(x)$ frequently denote the nonsmooth regularization such as $g(x)=\lambda\|x\|_1$. In fact, the above problem (1) comes from many machine learning problems, such as fair classifier, robust training, nonlinear temporal-difference learning in reinforcement learning [29] and robust federated learning.

When $g(x)=0$ and $h(y)=0$ in the problem (1), the classic gradient descent ascent (GDA) methods [26, 19] can effectively solve this problem, which alternatively conducts a gradient descent update on the variable x and a gradient ascent update on the variable y in each iteration. At the same time, some stochastic GDA methods [26, 19, 20, 13, 30, 12] have been proposed to solve the stochastic minimax problem (1), where $f(x,y)=\mathbb{E}_\xi[f(x,y;\xi)]$. More recently, some works [2, 4, 5] focus on more general minimax problem (1), where both $g(x)$ and $h(y)$ are possibly nonsmooth. Meanwhile,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Table 1: **Sample complexity** comparison of the representative proximal gradient descent ascent (PGDA) methods for obtaining an ϵ-stationary point of the **nonsmooth nonconvex** minimax problem (1). Note that these comparison methods don't rely on specific assumptions on the problem (1). Since [5] studies the deterministic PGDA relied on the Kurdyka-Lojasiewicz (KL) geometry assumption, it be excluded. Here κ denotes condition number of objective function $f(x, y)$.

Type	Algorithm	Reference	Loop(s)	Sample Complexity
Deterministic	MAPGDA	[2]	Double	$O(\kappa^{3/2}\epsilon^{-2})$
	PAGDA	[4]	Single	$O(\kappa^{2}\epsilon^{-2})$
	MDA	Ours	Single	$O(\kappa\epsilon^{-2})$
Stochastic	PASGDA	[4]	Single	$O(\kappa^{3}\epsilon^{-4})$
	SMDA	Ours	Single	$O(\kappa^{3}\epsilon^{-4})$
	VR-SMDA	Ours	Double	$O(\kappa^{3}\epsilon^{-3})$

some (stochastic) proximal gradient descent ascent (PGDA) methods [2, 4, 5] have been presented to solve the problem (1). However, they still suffer from the large sample complexities for finding an stationary point of the nonsmooth nonconvex minimax problem (1) without specific assumptions such as KL geometry (Please see Table 1).

In this paper, thus, we propose a class of novel mirror descent ascent methods by using dynamic (adaptive) mirror functions. Specifically, our methods perform an adaptive mirror descent update on the variable x and an adaptive mirror ascent update on the variable y alternatively in each iteration. Our new algorithmic framework is general and can generate many popular methods and their variants by adopting different mirror functions. For example, by adopting the mirror functions $\psi(x) = \frac{1}{2}\|x\|^2$ and $\phi(y) = \frac{1}{2}\|y\|^2$, our methods will include the classic (proximal) gradient descent ascent algorithms. Our main contributions are summarized as follows:

1) We propose a class of novel mirror descent ascent methods to solve the minimax problem (1) by using daynamic mirror functions. Moreover, we prove that the **stochastic** mirror descent ascent (SMDA) method has a sample complexity of $O(\kappa^3\epsilon^{-4})$ for finding an ϵ-stationary point.
2) We further introduce an accelerated **stochastic** mirror descent ascent (VR-SMDA) method by using the variance reduced technique of SARAH/SPIDER [24, 8]. Moreover, we prove that the VR-SMDA reaches a lower sample complexity of $O(\kappa^3\epsilon^{-3})$.
3) We provide a convergence analysis framework for these mirror descent ascent methods. In particular, we prove that our **deterministic** adaptive mirror descent ascent (MDA) reaches a lower sample complexity of $O(\kappa\epsilon^{-2})$ than the existing methods.

2 Related Works

In this section, we review the existing minimax optimization methods and stochastic mirror descent methods, respectively.

2.1 Minimax Optimization

Minimax optimization recently has been widely studied in machine learning community. The global convergence of strongly-convex-strongly-concave minimax optimization has been studied in [28, 18, 23]. Due to the popularity of nonconvex models in machine learning, many recent studies focused on the nonconvex minimax problems such as robustly deep neural networks training and GANs. For example, some effective gradient descent ascent methods [26, 25, 19, 30, 12, 11] have been proposed for the nonconvex-(strongly) concave minimax optimization. At the same time, Yang et al. [31] and Diakonikolas et al. [7] have studied a class of specific nonconvex-nonconcave minimax problems. More recently, some research works [2, 4, 5] began to study the nonsmooth nonconvex minimax problem (1).

2.2 Mirror Descent

Mirror descent method [3] is a powerful optimization tool because it can fit the geometry in optimization problem by using appropriate mirror functions. Beck et al. [3] studied the mirror descent for convex optimization. Subsequently, Lei et al. [16] integrated the variance reduced technique to the mirror descent algorithm for stochastic convex optimization. More recently, the variance-reduced adaptive stochastic mirror descent algorithm [17] was proposed to solve the nonsmooth nonconvex

finite-sum mini optimization. Moreover, Babanezhad et al. [1] presented a mirror-type algorithm for convex minimax optimization. At the same time, two typical mirror descent methods [26] were proposed for weakly convex minimax optimization. In fact, few work focuses on explicitly using the mirror-decent-type method to solve the nonsmooth nonconvex minimax problems. To the best of our knowledge, **this paper is the first one to study the mirror descent ascent methods for solving the nonsmooth nonconvex minimax problems**.

Notations: For two vectors x and y, $\langle x, y\rangle$ denotes their inner product. $\|\cdot\|$ denotes the ℓ_2 norm for vectors and spectral norm for matrices, respectively. $\nabla_x f(x,y)$ and $\nabla_y f(x,y)$ denote the partial derivatives *w.r.t.* variables x and y respectively, and let $\nabla f(x,y) = (\nabla_x f(x,y), \nabla_y f(x,y))$. $\partial f(x)$ denotes the subgradient set of function $f(x)$. Given a convex closed set $\mathcal{X}$, we define a projection operation $\text{Proj}_{\mathcal{X}}(\alpha) = \min_{x\in\mathcal{X}} \|x-\alpha\|^2$. Given the mini-batch samples $\mathcal{B} = \{\xi_i\}_{i=1}^b$, we let $\nabla f_{\mathcal{B}}(x) = \frac{1}{b}\sum_{i=1}^b f(x;\xi)$.

3 Preliminaries

In the section, we first review some basic information about mirror descent method.

Given a ρ-strongly convex and continuously-differentiable function $\psi(x)$, *i.e.*, $\langle x_1 - x_2, \nabla\psi(x_1) - \nabla\psi(x_2)\rangle \geq \rho\|x_1 - x_2\|^2$, we define a Bregman distance for any $x, z \in \mathcal{X}$:

$$D_\psi(z,x) = \psi(z) - \psi(x) - \langle \nabla\psi(x), z - x\rangle. \tag{2}$$

To solve the problem $\min_{x\in\mathcal{X}} f(x)$, the mirror descent method updates the variable x at $t+1$-th step:

$$x_{t+1} = \min_{x\in\mathcal{X}} \big\{\langle \nabla f(x_t), x\rangle + \frac{1}{\alpha} D_\psi(x, x_t)\big\}, \tag{3}$$

where $\alpha > 0$ is stepsize.

Next, we present some mild assumptions for Problem (1).

Assumption 1. *(Smoothness) For our deterministic and mini-batch stochastic algorithms (MDA and SMDA), we assume that the function $f(x,y)$ has an L_f-Lipschitz gradient, i.e., for all $x_1, x_2 \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, we have*

$$\|\nabla f(x_1,y_1) - \nabla f(x_2,y_2)\| \leq L_f\|(x_1,y_1) - (x_2,y_2)\|. \tag{4}$$

For our variance-reduced stochastic algorithm (VR-SMDA), we assume that each component function $f(x,y;\xi)$ has an L_f-Lipschitz gradient, i.e., for all $x_1, x_2 \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, we have

$$\|\nabla f(x_1,y_1;\xi) - \nabla f(x_2,y_2;\xi)\| \leq L_f\|(x_1,y_1) - (x_2,y_2)\|, \ \forall \xi. \tag{5}$$

Based on the above assumption (5), we can obtain the above assumption (4). Clearly, the assumption (4) is milder than (5). In Assumption 1, the assumption (4) is commonly used in the minimax optimization [19, 2, 4, 5]. While the assumption (5) is frequently used in the variance-reduced stochastic optimization [20, 13].

Assumption 2. *Each component function $f(x,y;\xi)$ has an unbiased stochastic gradient with bounded variance σ^2, i.e.,*

$$\mathbb{E}[\nabla f(x,y;\xi)] = \nabla f(x,y), \quad \mathbb{E}\|\nabla f(x,y;\xi) - \nabla f(x,y)\|^2 \leq \sigma^2. \tag{6}$$

Assumption 3. *The function $f(x,y)$ is μ-strongly concave w.r.t y, i.e., for all $x \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$, we have $\|\nabla_y f(x,y_1) - \nabla_y f(x,y_2)\| \geq \mu\|y_1 - y_2\|$. Then the following inequality holds*

$$f(x,y_1) \leq f(x,y_2) + \langle \nabla_y f(x,y_2), y_1 - y_2\rangle - \frac{\mu}{2}\|y_1 - y_2\|^2. \tag{7}$$

Assumption 4. *The functions $g(x)$ and $h(y)$ are convex but possibly nonsmooth.*

Assumption 3 shows that the function $f(x,y)$ is μ-strongly concave w.r.t y. Assumption 4 shows that the function $h(y)$ is convex. Thus, the function $\{f(x,y) - h(y)\}$ is strongly concave in $y \in \mathcal{Y}$, there exists a unique solution to the problem $\max_{y\in\mathcal{Y}}\{f(x,y) - h(y)\}$ for any x. Let $y^*(x) = \arg\max_{y\in\mathcal{Y}}\{f(x,y) - h(y)\}$, and we define a function $\Phi(x) = f(x,y^*(x)) - h(y^*(x)) = \max_{y\in\mathcal{Y}}\{f(x,y) - h(y)\}$.

Assumption 5. *For any $\alpha \in \mathbb{R}$, the sub-level set $\{x : \Phi(x) + g(x) \leq \alpha\}$ is compact. The function $\Phi(x) + g(x)$ is bounded below in $\mathcal{X}$,* i.e., $F^* = \inf_{x\in\mathcal{X}}\{\Phi(x) + g(x)\} > -\infty$.

Assumption 5 is a basic assumption in minimax optimization used in [5].

Algorithm 1 (Stochastic) Mirror Descent Ascent Algorithm

Input: T, stepsizes $\{\gamma_t > 0, \lambda_t > 0, \eta_t \in (0,1]\}_{t=1}^T$, mini-batch size b ;
initialize: $x_1 \in \mathcal{X}$ and $y_1 \in \mathcal{Y}$;
for $t = 1, 2, \ldots, T$ **do**
  MDA: Compute partial derivatives $v_t = \nabla_x f(x_t, y_t)$ and $w_t = \nabla_y f(x_t, y_t)$;
  SMDA: Generate randomly mini-batch samples $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$ with $|\mathcal{B}_t| = b$, and compute stochastic partial derivatives $v_t = \nabla_x f_{\mathcal{B}_t}(x_t, y_t)$ and $w_t = \nabla_y f_{\mathcal{B}_t}(x_t, y_t)$;
  Given the mirror functions ψ_t and ϕ_t;
  Update $x_{t+1} = \arg\min_{x\in\mathcal{X}} \big\{\langle v_t, x\rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t)\big\}$;
  Update $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$ where $\tilde{y}_{t+1} = \arg\max_{y\in\mathcal{Y}} \big\{\langle w_t, y\rangle - h(y) - \frac{1}{\lambda_t} D_{\phi_t}(y, y_t)\big\}$;
end for
Output: x_ζ and y_ζ chosen uniformly random from $\{x_t, y_t\}_{t=1}^T$.

4 Mirror Descent Ascent Methods

In the section, we propose a class of novel mirror descent ascent methods to solve Problem (1). Specifically, we first propose a deterministic mirror descent ascent (MDA) method, and stochastic mirror descent ascent (SMDA) method. Based on them, we further introduce an accelerated stochastic mirror descent ascent (VR-SMDA) using variance reduced technique of SARAH/SPIDER [24, 8].

4.1 MDA and SMDA algorithms

When $f(x, y)$ is a deterministic function, we propose a deterministic mirror descent ascent (MDA) method to solve the deterministic problem (1). When $f(x, y) = \mathbb{E}[f(x; \xi)]$ is a stochastic function, we propose a stochastic mirror descent ascent (SMDA) to solve the stochastic problem (1). Specifically, Algorithm 1 shows the algorithmic framework of the MDA and SMDA algorithms for solving the problem (1).

In Algorithm 1, we use (stochastic) mirror decent to update the parameter x, and simultaneously use (stochastic) mirror ascent to update the parameter y. When choose the mirror functions $\psi_t(x) = \frac{1}{2}\|x\|^2$ and $\phi_t(y) = \frac{1}{2}\|y\|^2$ for all $t \geq 1$, we have $D_{\psi_t}(x, x_t) = \frac{1}{2}\|x - x_t\|^2$ and $D_{\phi_t}(y, y_t) = \frac{1}{2}\|y - y_t\|^2$. Under this case, Algorithm 1 will reduce the standard (stochastic) proximal gradient descent ascent algorithm. When choose the mirror functions $\psi_t(x) = \frac{1}{2}x^T H_t x$ and $\phi_t(y) = \frac{1}{2}y^T G_t y$ for all $t \geq 1$, we have $D_{\psi_t}(x, x_t) = \frac{1}{2}(x - x_t)^T H_t(x - x_t)$ and $D_{\phi_t}(y, y_t) = \frac{1}{2}(y - y_t)^T G_t(y - y_t)$, where $H_t \succeq \rho I_d$ and $G_t \succeq \rho I_p$. For example, given $\alpha \in (0, 1)$ and $\rho > 0$, we can generate the matrices H_t and G_t like as in ADAM algorithm [14], defined as

$$\tilde{v}_0 = 0,\ \tilde{v}_t = \alpha\tilde{v}_{t-1} + (1-\alpha)\nabla_x f(x_t, y_t; \xi_t)^2, \quad H_t = \text{diag}(\sqrt{\tilde{v}_t} + \rho),\ t \geq 1 \tag{8}$$

$$\tilde{w}_0 = 0,\ \tilde{w}_t = \alpha\tilde{w}_{t-1} + (1-\alpha)\nabla_y f(x_t, y_t; \xi_t)^2, \quad G_t = \text{diag}(\sqrt{\tilde{w}_t} + \rho),\ t \geq 1 \tag{9}$$

Under this case, our SMDA algorithm will reduce a novel adaptive gradient descent ascent algorithm.

In the problem (1), the functions $g(x)$ and $h(y)$ are generally nonsmooth, e.g., $g(x) = \nu_1\|x\|_1$ and $h(y) = \nu_2\|y\|_1$ with $\nu_1, \nu_2 > 0$. When choose the mirror functions $\psi_t(x) = \frac{1}{2}x^T H_t x$ and $\phi_t(y) = \frac{1}{2}y^T G_t y$ for all $t \geq 1$, and the matrices H_t and G_t are diagonal, e.g., generated from the above (8) and (9), we can use the soft thresholding operator $S(a, \lambda) = \text{sign}(a)\max(|a| - \lambda, 0) = \arg\min_z\{\frac{1}{2}(z - a)^2 + \lambda|z|\}$ to obtain the closed-form solutions of the following subproblems:

$$\min_{x\in\mathcal{X}} \big\{\langle v_t, x\rangle + \nu_1\|x\|_1 + \frac{1}{2\gamma_t}(x - x_t)^T H_t(x - x_t)\big\}, \tag{10}$$

$$\max_{y\in\mathcal{Y}} \big\{\langle w_t, y\rangle - \nu_2\|y\|_1 - \frac{1}{2\lambda_t}(y - y_t)^T G_t(y - y_t)\big\}, \tag{11}$$

where $H_t = \text{diag}(h_{1,t}, \cdots, h_{d,t})$ with $h_{i,t} > 0$ for $i \in [d]$, and $G_t = \text{diag}(g_{1,t}, \cdots, g_{p,t})$ with $g_{j,t} > 0$ for $j \in [p]$. Without loss of generality, assume the constraint sets $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}^p$.

Algorithm 2 Accelerated Stochastic Mirror Descent Ascent (VR-SMDA) Algorithm

Input: T, q, stepsizes $\{\gamma_t > 0, \lambda_t > 0, \eta_t \in (0,1]\}_{t=1}^T$, mini-batch sizes b and b_1; ;
initialize: $x_1 \in \mathcal{X}$ and $y_1 \in \mathcal{Y}$;
for $t = 1, 2, \ldots, T$ **do**
  if $\mod (t, q) = 0$ **then**
    Randomly generate mini-batch samples $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$ with $|\mathcal{B}_t| = b$;
    Compute stochastic partial derivatives $v_t = \nabla_x f_{\mathcal{B}_t}(x_t, y_t)$ and $w_t = \nabla_y f_{\mathcal{B}_t}(x_t, y_t)$;
  else
    Randomly generate mini-batch samples $\mathcal{I}_t = \{\xi_t^i\}_{i=1}^{b_1}$ with $|\mathcal{I}_t| = b_1$;
    Compute stochastic partial derivatives

$$v_t = \nabla_x f_{\mathcal{I}_t}(x_t, y_t) - \nabla_x f_{\mathcal{I}_t}(x_{t-1}, y_{t-1}) + v_{t-1}, \tag{14}$$
$$w_t = \nabla_y f_{\mathcal{I}_t}(x_t, y_t) - \nabla_y f_{\mathcal{I}_t}(x_{t-1}, y_{t-1}) + w_{t-1}; \tag{15}$$

  end if
  Given the mirror functions ψ_t and ϕ_t;
  Update $x_{t+1} = \arg\min_{x \in \mathcal{X}} \big\{\langle v_t, x\rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t)\big\}$;
  Update $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$ where $\tilde{y}_{t+1} = \arg\max_{y \in \mathcal{Y}} \big\{\langle w_t, y\rangle - h(y) - \frac{1}{\lambda_t} D_{\phi_t}(y, y_t)\big\}$;
end for
Output: x_ζ and y_ζ chosen uniformly random from $\{x_t, y_t\}_{t=1}^T$.

Specifically, we obtain, for $i \in [d]$

$$S\big(x_{i,t} - \frac{\gamma_t}{h_{i,t}} v_{i,t}, \frac{\gamma_t \nu_1}{h_{i,t}}\big) = \arg\min_{x_i \in \mathbb{R}} \big\{\langle v_{i,t}, x_i\rangle + \nu_1 |x_i| + \frac{h_{i,t}}{2\gamma_t}(x_i - x_{i,t})^2\big\}. \tag{12}$$

Similarly, we can obtain, for $j \in [p]$

$$S\big(y_{j,t} + \frac{\lambda_t}{g_{j,t}} w_{j,t}, \frac{\lambda_t \nu_2}{g_{j,t}}\big) = \arg\max_{y_j \in \mathbb{R}} \big\{\langle w_{j,t}, y_j\rangle - \nu_2 |y_j| - \frac{g_{j,t}}{2\lambda_t}(y_j - y_{j,t})^2\big\}. \tag{13}$$

4.2 VR-SMDA Algorithm

In this subsection, we propose an accelerated stochastic mirror descent ascent (VR-SMDA) algorithm to solve the stochastic problem (1). Algorithm 2 describes the detailed algorithmic framework of the VR-SMDA method.

In Algorithm 1, we only draw a mini-batch samples $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$ at each iteration. Clearly, the mini-batch samples will take large variances in our SMDA algorithm. Thus, we use the variance reduced technique of SPIDER in our VR-SMDA algorithm to accelerate it. Specifically, when $\mod (t, q) = 0$, we draw a relative large batch samples $\mathcal{B}_t = \{\xi_t^i\}_{i=1}^b$ to estimate our stochastic partial derivatives v_t and w_t; when $\mod (t, q) \neq 0$, we only draw a mini-batch samples $\mathcal{I}_t = \{\xi_t^i\}_{i=1}^{b_1}$ $(b > b_1)$ to estimate v_t and w_t in (14) and (15). Since samples $\mathcal{I}_t$ are independent to variables $\{x_t, x_{t-1}, y_t, y_{t-1}, v_{t-1}\}$, by using Assumption 2, we have

$$\mathbb{E}_{\mathcal{I}_t}[v_t] = \nabla_x f(x_t, y_t) - \nabla_x f(x_{t-1}, y_{t-1}) + v_{t-1} \neq \nabla_x f(x_t, y_t), \tag{16}$$
$$\mathbb{E}_{\mathcal{I}_t}[w_t] = \nabla_y f(x_t, y_t) - \nabla_y f(x_{t-1}, y_{t-1}) + w_{t-1} \neq \nabla_y f(x_t, y_t). \tag{17}$$

Thus, the partial derivative estimators v_t and w_t are biased.

In our algorithms, we simultaneously use the mirror ascent iteration and momentum iteration to update the variable y. In the momentum iteration $y_{t+1} = (1 - \eta_t)y_t + \eta_t \tilde{y}_{t+1}$, we choose $\eta_t \in (0, 1]$ to ensure $y_{t+1} \in \mathcal{Y}$ for all $t \geq 1$ since $y_1 \in \mathcal{Y}$. In fact, we use this momentum iteration to keep the gap between the terms $\|y_{t+1} - y^*(x_{t+1})\|^2$ and $\|y_t - y^*(x_t)\|^2$ that guarantee the convergence of our algorithms (Please see the convergence analysis in supplementary materials).

5 Convergence Analysis

In this section, we study the convergence properties of our algorithms (*i.e.*, MDA, SMDA and VR-SMDA) under some mild conditions. All related proofs are provided in supplementary materials.

We first introduce a useful convergence metric $\mathbb{E}\|\mathcal{G}_t\|$ to measure convergence properties of our algorithms. Given the generated parameters x_t at iteration t by our algorithms, we define the generalized gradient at iteration t as

$$\mathcal{G}_t = \frac{1}{\gamma_t}(x_t - x_{t+1}^+), \tag{18}$$

$$x_{t+1}^+ = \arg\min_{x\in\mathcal{X}} \big\{\langle \nabla\Phi(x_t), x\rangle + g(x) + \frac{1}{\gamma_t} D_{\psi_t}(x, x_t)\big\}, \tag{19}$$

where $\Phi(x) = f(x, y^*(x)) - h(y^*(x)) = \max_{y\in\mathcal{Y}}\{f(x, y) - h(y)\}$. When $\mathcal{X} = \mathbb{R}^d$ and $g(x)$ is a constant, and $\psi_t(x) = \frac{1}{2}\|x\|^2$, we have $\mathcal{G}_t = \nabla\Phi(x) = \nabla_x f(x, y^*(x))$. Under this case, our convergence metric $\mathbb{E}\|\mathcal{G}_t\| = \mathbb{E}\|\nabla_x f(x, y^*(x))\|$ is a common convergence metric used in [19].

Since the objective function $f(x, y)$ is μ-strongly concave in y, the standard proximal gradient ascent can easily obtain the global solution of the subproblem $\max_{y\in\mathcal{Y}}\{f(x, y) - h(y)\}$. Without loss of generalization, in our theoretical analysis, we give the mirror function $\phi_t(y) = \frac{1}{2}\|y\|^2$, and the mirror functions $\psi_t(x)$ for all $t \geq 1$ are ρ-strong convex. Here, the constant ρ can be seen as a lower bound of the strong convexity of all the mirror functions $\{\psi_t(x)\}$, as in [17].

5.1 Convergence Analysis of the SMDA and MDA Algorithms

In the subsection, we provide the convergence properties of our SMDA and MDA algorithms.

Theorem 1. *Suppose the sequence $\{x_t, y_t\}_{t=1}^T$ be generated from Algorithm 1 using stochastic partial derivatives (i.e., SMDA algorithm). Let $0 < \eta = \eta_t \leq 1$ and $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$ and $0 < \lambda = \lambda_t \leq \frac{1}{6L_f}$, we have*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}} + \frac{10\sigma}{\sqrt{3b\rho}} + \frac{20\sigma\sqrt{\eta\lambda}}{3\sqrt{\gamma\rho\mu b}}, \tag{20}$$

where $\kappa = L_f/\mu$, $L = L_f(1+\kappa)$ and $\tilde{F}(x) = \Phi(x) + g(x)$.

Remark 1. *Given $0 < \eta \leq 1$, $\lambda = O(\frac{1}{L_f})$, $\gamma = \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$ and $\rho = O(\sqrt{L_f})$, we have $\gamma\rho = O(\frac{1}{\kappa^2})$. Thus, our SMDA algorithm has a convergence rate of $O(\sqrt{\frac{\kappa^2}{T}} + \sqrt{\frac{\kappa}{b}})$. Let $b = T/\kappa$ and $\sqrt{\frac{\kappa^2}{T}} \leq \epsilon$, we have $T \geq \kappa^2\epsilon^{-2}$. Since our SMDA algorithm requires b samples to estimate the stochastic partial directives v_t and w_t at each iteration, and needs T iterations, it has a sample complexity of $bT = O(\kappa^3\epsilon^{-4})$ for finding an ϵ-stationary point.*

Theorem 2. *Suppose the sequence $\{x_t, y_t\}_{t=1}^T$ be generated from Algorithm 1 using the deterministic partial derivatives (i.e., MDA algorithm). Let $0 < \eta = \eta_t \leq 1$ and $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{9\eta\rho\mu\lambda}{800\kappa^2}, \frac{2\eta\mu\rho\lambda}{25L_f^2})$ and $0 < \lambda = \lambda_t \leq \frac{1}{6L_f}$, we have*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}}. \tag{21}$$

Remark 2. *Under the same conditions in the above Theorem 1, let $\rho = O(L_f^{(\frac{1}{2}+\nu)})$ $\nu \in [0, \frac{1}{2}]$, we have $\gamma\rho = O(\frac{1}{\kappa^{(2-2\nu)}})$. Under this case, our MDA algorithm has a sample complexity of $T = O(\kappa^{(2-2\nu)}\epsilon^{-2})$, $\nu \in [0, \frac{1}{2}]$ for finding an ϵ-stationary point. When $\nu = \frac{1}{2}$, our MDA algorithm has a lower sample complexity of $T = O(\kappa\epsilon^{-2})$.*

5.2 Convergence Analysis of the VR-SMDA Algorithm

In the subsection, we provide the convergence properties of the VR-SMDA algorithm.

Theorem 3. *Suppose the sequence $\{x_t, y_t\}_{t=1}^T$ be generated from Algorithm 2. Let $b_1 = q$, $0 < \eta = \eta_t \leq 1$, $0 < \gamma = \gamma_t \leq \min(\frac{3\rho}{4L}, \frac{\eta\mu\lambda\rho}{38L_f^2}, \frac{3\rho}{19L_f^2\eta}, \frac{\rho\eta}{8}, \frac{9\rho\eta\mu\lambda}{400\kappa^2})$ and $0 < \lambda = \lambda_t \leq \min(\frac{1}{6L_f}, \frac{9\mu}{100\eta^2L_f^2})$, we have*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|\mathcal{G}_t\| \leq \frac{4\sqrt{2(\tilde{F}(x_1) - F^*)}}{\sqrt{3T\gamma\rho}} + \frac{2\sqrt{2}\sigma}{\sqrt{\gamma\rho\eta b}L_f}, \tag{22}$$

where $\kappa = L_f/\mu$, $L = L_f(1+\kappa)$ and $\tilde{F}(x) = \Phi(x) + g(x)$.

Remark 3. *Given $0 < \eta \leq 1$, $\lambda = O(\frac{1}{\kappa L_f})$, $\gamma = \min(\frac{3\rho}{4L}, \frac{\eta\mu\lambda\rho}{38L_f^2}, \frac{3\rho}{19L_f^2\eta}, \frac{\rho\eta}{8}, \frac{9\rho\eta\mu\lambda}{400\kappa^2})$ and $\rho = O(L_f)$, we have $\gamma\rho = O(\frac{1}{\kappa^2})$. Thus, our VR-SMDA algorithm has a convergence rate of $O(\sqrt{\frac{\kappa^2}{T}} + \sqrt{\frac{\kappa}{b}})$. Let $b = T/\kappa$ and $\sqrt{\frac{\kappa^2}{T}} \leq \epsilon$, we have $T \geq \kappa^2\epsilon^{-2}$. Further let $T = O(\kappa^2\epsilon^{-2})$, $b_1 = q = O(\kappa\epsilon^{-1})$ and $b = O(\kappa\epsilon^{-2})$. Since Algorithm 2 requires b samples to estimate the stochastic directives v_t and w_t at each iteration when $\mod(t,q) = 0$, otherwise needs b_1 samples, and need T iterations, it has a sample complexity of $b_1T + bT/q = O(\kappa^3\epsilon^{-3})$ for finding an ϵ-stationary point.*

Remark 4. *The above sample complexities are obtained when given $\rho = O(L_f)$, where L_f is the smooth parameter of objective function $f(x,y)$. Although in the objective function $f(x,y)$, L_f may be large, we can easily change the original objective function $f(x,y)$ to a new function $\hat{f}(x,y) = \tau f(x,y),\ 0 < \tau < 1$. Since $\nabla\hat{f}(x,y) = \tau\nabla f(x,y)$, the gradient of function $\hat{f}(x,y)$ is $\hat{L}$-Lipschitz continuous ($\hat{L} = \tau L_f$). Thus, we can choose a suitable hyper-parameter τ to let this new objective function $\hat{f}(x,y)$ satisfy the condition $\rho = O(\hat{L})$.*

6 Novelties of Our Methods

A related the mirror descent method was recently released [17]. To help readers understand the difference between our works and [17], we summarize the novelties of our methods as follows:

1) In our algorithms, we integrate the mirror ascent iteration with the momentum technique to update the variable y (please see the step 8 of Algorithm 1 and the step 13 of Algorithm 2). Under this case, we can obtain the gap of the terms $\|y_{t+1} - y^*(x_{t+1})\|^2$ and $\|y_t - y^*(x_t)\|^2$ (please see the Lemma 6 in Appendix A.1), which can guarantee the convergence of our algorithms. However, [17] only uses the mirror descent iteration to update the variable.
2) We provide a useful convergence analysis framework for our methods, which does not rely on the unbiased stochastic gradient (or partial derivatives). For example, our VR-SMDA algorithm builds on the variance reduce technique SPIDER, which generates biased stochastic gradient. However, the convergence analysis of the methods in [17] rely on the unbiased stochastic gradient. Moreover, in our convergence analysis, we establish a useful and key Lyapunov function $\Omega_t = \Phi(x_t) + g(x_t) + \|y_t - y^*(x_t)\|^2$ for all $t \geq 1$. Overall, our convergence analysis is totally different from the convergence analysis of [17].
3) To the best of our knowledge, this paper is the first one to study the mirror descent ascent methods for solving the nonsmooth nonconvex minimax problems. In particular, we first show that the strongly convex mirror functions can improve the sample complexity of mirror-type algorithms. Specifically, the different parameter ρ affects the sample complexity, which can be seen as a lower bound of the strong convexity of mirror functions $\{\psi_t(x)\}$.

7 Experiments

In this section, we perform the numerical experiments to validate the efficiency of our algorithms. Specifically, we conduct two tasks: fair classifier and robust neural network training. In the experiments, we conduct these tasks on the Fashion-MNIST dataset as in [25] as well MNIST dataset and CIFAR-10 dataset. Fashion-MNIST dataset and MNIST dataset consist of 28×28 arrays of grayscale pixel images classified into 10 categories, and includes $60,000$ training images and $10,000$ testing images. CIFAR-10 dataset includes $60,000$ 32×32 colour images ($50,000$ training images and $10,000$ testing images). We compare our algorithms (MDA, SMDA and VR-SMDA) with the existing proximal gradient descent ascent algorithms (MAPGDA [2], PAGDA [4] and PASGDA[4]) for solving these nonsmooth nonconvex minimax problems. Note that the Proximal-GDA algorithm in [5] only is a non-accelerated version of MAPGDA algorithm [2], so we omit it in the comparison methods. The experiments are run on CPU machines with 2.3 GHz Intel Core i9 as well as NVIDIA Tesla P40 GPU.

7.1 Fair Classifier

The first task is to train a fair classifier to minimize the maximum loss over categories. Here, we use a nonconvex Convolutional Neural Network (CNN) model as classifier. Similar to [25], we limit our

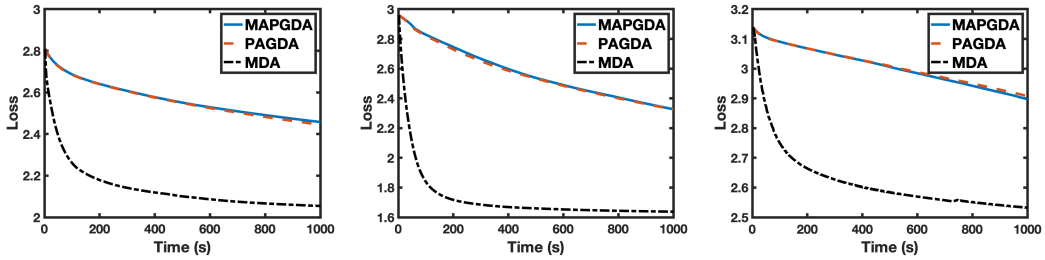
experiment to the three categories. To be precise, the Fashion-MNIST dataset is limited to T-shirt/top, Coat and Shirt categories. The MNIST dataset is limited to digital numbers $\{0, 2, 3\}$, and CIFAR10 dataset is limited to airplane, automobile and bird. Specifically, we will solve the following minimax problem:

$$\min_{w} \max_{y \in \mathcal{Y}} \Big\{ \sum_{i=1}^{3} y_i \mathcal{L}_i(w) + g(w) - h(y) \Big\} \quad \text{s.t. } \mathcal{Y} = \Big\{ y \mid y_i \geq 0, \sum_{i=1}^{3} y_i = 1 \Big\}, \tag{23}$$

where w denotes the CNN model parameters; and $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are the loss functions corresponding to the samples in three different categories. Here we let $g(w) = \nu_1 \|w\|_1$ and $h(y) = \nu_2 \|y\|_2^2$, where $\nu_1 > 0$ and $\nu_2 > 0$. Clearly the inner maximization problem is strongly concave, and the outer minimization problem is non-convex non-smooth. Thus our theory can be applied.

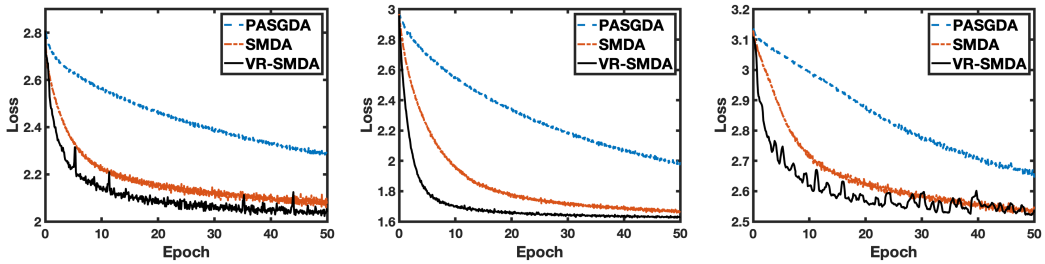
In the experiment, we let $\nu_1 = 0.001$ and $\nu_2 = 0.1$ in the above problem (23). For fair comparison, we use the same step size for all methods. Specifically, step-size for w is 0.001 and step-size for y is 0.00001. We apply xavier normal initialization to CNN layer. In our algorithms, we choose the mirror functions $\psi_t(w) = \frac{1}{2} w^T H_t w$ and $\phi_t(y) = \frac{1}{2} y^T G_t y$ for all $t \geq 1$, where H_t and G_t are generated from (8) and (9) respectively, given $\alpha = 0.1$ and $\rho = 0.00005$. We set $\eta = 1$ in our algorithms. We run all deterministic algorithms for 1000 seconds and all stochastic algorithms for 50 epochs. Then we record the loss value. For stochastic methods, batch sizes of PASGDA and SMDA are 3000. For our VR-SMDA, we set the large batch size $b = 60000$ and the mini-batch size $b_1 = q = 3000$.

Figure 1 shows the loss vs time of different deterministic methods. Figure 2 plots the loss vs epoch of different stochastic methods. From these results, we can find that our algorithms consistently outperform the other algorithms with a great margin.



(a) Fashion-MNIST (b) MNIST (c) CIFAR-10

Figure 1: Results of different deterministic methods on the fair classifier task.



(a) Fashion-MNIST (b) MNIST (c) CIFAR-10

Figure 2: Results of different stochastic methods on the fair classifier task.

7.2 Robust Neural Network Training

The second task is to train robust Neural Network (NN). Although NNs have been widely used in many applications such as image classification, they are vulnerable to adversarial attacks such as Fast Gradient Sign Method (FGSM) [10] and Projected Gradient Descent (PGD) attack [15]. In other word, a small perturbation in the data input of NN can significantly change its output. Thus, we try to train a robust neural network against these adversarial attacks, which generally reformulate this robust training into the following minimax problem:

$$\min_{w} \sum_{i=1}^{n} \max_{y_i \in \mathcal{Y}} \mathcal{L}\big(f(a_i + y_i; w), b_i\big), \quad \mathcal{Y} = \{ y \mid \|y_i\|_\infty \leq \varepsilon,\ i \in [n] \} \tag{24}$$

where (a_i, b_i) denotes the i-th data point, and w is the parameter of NN, and y_i denotes is the perturbation added to the i-th data point. Following [25], we approximate the inner maximization problem of the above minimax problem (24) with the following finite max problem

$$\min_{w} \sum_{i=1}^{n} \max \ \big\{\mathcal{L}\big(f(\hat{a}_{i,0}(w); w), b_i\big), \cdots, \mathcal{L}\big(f(\hat{a}_{i,9}(w); w), b_i\big)\big\}, \tag{25}$$

where $\hat{a}_{i,j}(w)$ is the result of a targeted attack on data point a_i that is changed the output of the network to label j. Following [25], we can obtain $\hat{a}_{i,j}(w)$ by using the following procedure: In the last layer of the NN architecture for learning classification on MNIST (Fashion-MNIST) we have 10 different neurons, each corresponding with one category of classification. For any sample (a_i, b_i) in the dataset and starting from $a_{i,j}^0 = a_i$ for any $j = 1, 2, \cdots, 9$, we run projected gradient ascent to obtain the following chain of points:

$$a_{i,j}^{k+1} \ = \ \text{Proj}_{\mathcal{Y}}\big[a_{i,j}^k + \mu \nabla_a \big(Z_j(a_{i,j}^k, w) - Z_{b_i}(a_{i,j}^k, w)\big)\big], \ k = 0, 1, \cdots, K-1 \tag{26}$$

where $\mu > 0$ is a stepsize, and Z_j is the network logit before softmax corresponding to label j. Finally, we can set $\hat{a}_{i,j}(w) = a_{i,j}^K$ in the above minimax problem (25).

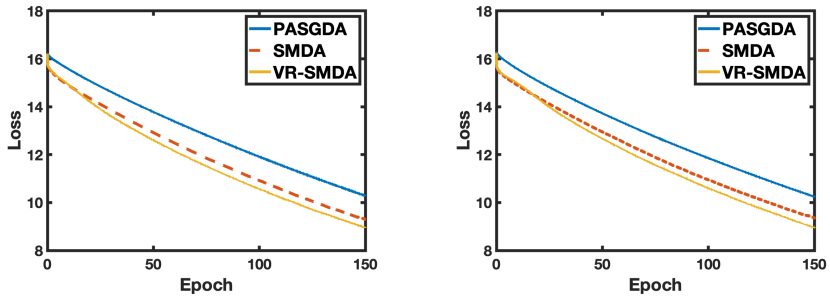
Next, we can replace the above problem (25) with the following nonconvex nonsmooth problem:

$$\min_{w} \sum_{i=1}^{n} \max_{u \in U} \big\{ \sum_{i=0}^{9} u_i \ \mathcal{L}\big(f(a_{i,j}^K; w), b_i\big) + \nu_1 \|w\|_1 - \nu_2 \|u\|_2^2 \big\}, \tag{27}$$

$$\text{s.t. } U = \big\{u \in \mathbb{R}^{10} \mid u_i \geq 0, \ \sum_{i=0}^{9} u_i = 1\big\},$$

where $\nu_1 > 0$ and $\nu_2 > 0$. Clearly, the inner maximization problem in (27) is strongly-concave, and its outer minimization problem is non-convex non-smooth. Thus, our theory can be applied to the above minimax problem (27).

In the experiment, we set $\nu_1 = 0.001$ and $\nu_2 = 0.1$ in the above problem (27). The Fashion-MNIST and MNIST datasets are tested in this experiment. In the above problem (26), we set $K = 5$. For fair comparison, we use the same step size for all methods. Specifically, step-size for w is 0.001 and step-size for u is 0.00001. We set $\eta = 1$ in our algorithms. For our algorithms, we choose the mirror functions $\psi_t(w) = \frac{1}{2} w^T H_t w$ and $\phi_t(u) = \frac{1}{2} u^T G_t u$ for all $t \geq 1$, where H_t and G_t are generated from (8) and (9) respectively, given $\alpha = 0.1$ and $\rho = 0.00005$. Here we only conduct experiments with stochastic methods, and batch-sizes of PASGDA and SMDA are 600. For our VR-SMDA, we set $b = 1200$ and $b_1 = q = 600$. Following [25], we set $\varepsilon = 0.4$ in the above problem (24). Figure 3 shows the loss vs epoch of different stochastic methods. From these results, we can find that our algorithms outperform the other algorithms, and the VR-SMDA consistently outperforms the SMDA.



(a) Fashion-MNIST (b) MNIST

Figure 3: Results of different stochastic methods on the robust neural network training task.

8 Conclusions

In the paper, we studied the nonsmooth nonconvex minimax optimization, and proposed a class of novel adaptive mirror descent ascent methods to solve these minimax problems. Moreover, we provided an effective convergence analysis framework for our methods. In particular, this paper is the first one to discover that the strongly mirror function can improve the sample complexity of mirror-type algorithms. The experimental results on fair classifier and robust neural network training tasks verify that our new algorithms consistently outperform the related algorithms.

References

[1] R. Babanezhad and S. Lacoste-Julien. Geometry-aware universal mirror-prox. *arXiv preprint arXiv:2011.11203*, 2020.

[2] B. Barazandeh and M. Razaviyayn. Solving non-convex non-differentiable min-max games using proximal gradient method. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3162–3166, 2020.

[3] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[4] R. I. Boţ and A. Böhm. Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems. *arXiv preprint arXiv:2007.13605*, 2020.

[5] Z. Chen, Y. Zhou, T. Xu, and Y. Liang. Proximal gradient descent-ascent: Variable convergence under kl geometry. *arXiv preprint arXiv:2102.04653*, 2021.

[6] Y. Deng, M. M. Kamani, and M. Mahdavi. Distributionally robust federated averaging. *arXiv preprint arXiv:2102.12660*, 2021.

[7] J. Diakonikolas, C. Daskalakis, and M. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2746–2754. PMLR, 2021.

[8] C. Fang, C. J. Li, Z. Lin, and T. Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[11] Z. Guo, Y. Xu, W. Yin, R. Jin, and T. Yang. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 2021.

[12] Z. Guo, Z. Yuan, Y. Yan, and T. Yang. Fast objective and duality gap convergence for non-convex strongly-concave min-max problems. *arXiv preprint arXiv:2006.06889*, 2020.

[13] F. Huang, S. Gao, J. Pei, and H. Huang. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[16] L. Lei and M. I. Jordan. On the adaptivity of stochastic gradient-based optimization. *SIAM Journal on Optimization*, 30(2):1473–1500, 2020.

[17] W. Li, Z. Wang, Y. Zhang, and G. Cheng. Variance reduction on adaptive stochastic mirror descent. *arXiv preprint arXiv:2012.13760*, 2020.

[18] T. Liang and J. Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915. PMLR, 2019.

[19] T. Lin, C. Jin, and M. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

[20] L. Luo, H. Ye, Z. Huang, and T. Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

[21] D. Madras, E. Creager, T. Pitassi, and R. Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[23] A. Mokhtari, A. Ozdaglar, and S. Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020.

[24] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621. PMLR, 2017.

[25] M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

[26] H. Rafique, M. Liu, Q. Lin, and T. Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

[27] M. Sanjabi, M. Razaviyayn, J. Ba, and J. D. Lee. On the convergence and robustness of training gans with regularized optimal transport. *Advances in Neural Information Processing Systems*, 2018:7091–7101, 2018.

[28] P. Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995.

[29] H.-T. Wai, M. Hong, Z. Yang, Z. Wang, and K. Tang. Variance reduced policy evaluation with smooth function approximation. *Advances in Neural Information Processing Systems*, 32:5784–5795, 2019.

[30] Z. Xu, H. Zhang, Y. Xu, and G. Lan. A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. *arXiv preprint arXiv:2006.02032*, 2020.

[31] J. Yang, N. Kiyavash, and N. He. Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the supplemental material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]