SAPD+ : An Accelerated Stochastic Method for Nonconvex-Concave Minimax Problems

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a new stochastic method SAPD+ for solving nonconvex-concave minimax problems of the form $\min \max \mathcal{L}(x, y) = f(x) + \Phi(x, y) - g(y)$, where f, g are closed convex and $\Phi(x, y)$ is a smooth function that is weakly convex in x, (strongly) concave in y. For both strongly concave and merely concave settings, SAPD+ achieves the best known oracle complexities of $\mathcal{O}(L\kappa_y\epsilon^{-4})$ and $\mathcal{O}(L^3\epsilon^{-6})$, respectively, without assuming compactness of the problem domain, where κ_y is the condition number and L is the Lipschitz constant. We also propose SAPD+ with variance reduction, which enjoys the best known oracle complexity of $\mathcal{O}(L\kappa_y^2\epsilon^{-3})$ for weakly convex-strongly concave setting. We demonstrate the efficiency of SAPD+ on a distributionally robust learning problem with a weakly convex cost and also on a multi-class classification problem in deep learning.

1 Introduction

We consider the following saddle-point (SP) problem:

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \mathcal{L}(x, y) \triangleq f(x) + \Phi(x, y) - g(y), \tag{1}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are, n and m dimensional Euclidean spaces, the function $\Phi(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is smooth and nonconvex in $x \in \mathcal{X}$ and μ_y-strongly concave in $y \in \mathcal{Y}$ for some $\mu_y \geq 0$ (with the convention that for $\mu_y = 0$, Φ is merely concave (MC) in y), and the functions f and g are closed, convex and possibly nonsmooth. In this paper, we consider a particular case of nonconvexity, i.e., we assume that $\Phi(\cdot, y)$ is weakly convex (WC) for any fixed $y \in \mathbf{dom}\, g \subset \mathcal{Y}$. Weakly convex functions constitute a rich class of non-convex functions and arise naturally in many practical settings for machine learning (ML) applications [8, 35], precise definitions will be given later in Section 2. In practice, WC assumption is widely satisfied, e.g., under smoothness –see remark 1, in fact it can be argued that most of the work in related literature considering nonconvex-(strongly) concave SP problems provide their analyses under the premise of weak convexity. The problem (1) with $\mu_y > 0$ is called a weakly convex-strongly concave (WCSC) saddle-point problem, whereas for $\mu_y = 0$, it is called a weakly convex-merely concave (WCMC) saddle-point problem. Both problems arise frequently in many ML problems including constrained optimization of WC objectives based on Lagrangian duality [21], Generative Adversarial Networks (GAN) (where x denotes the parameters of the generator network whereas y represents the parameters of the discriminator network [12]), distributional robust learning with weakly convex loss functions such as those arising in deep learning [13, 35] and learning problems with non-decomposable losses [35].

There are important two settings for (1): (i) the *deterministic setting*, where the partial gradients of Φ are exactly available, (ii) the *stochastic setting*, where only stochastic estimates of the gradients are available. Although, recent years have witnessed significant advances in the deterministic setting [6, 16, 18, 22, 23, 25, 34, 36, 38]; our focus in this paper will be mainly the *stochastic setting* which is more relevant and more applicable to ML problems. Indeed, due to large-dimensions and the sheer

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

size of the modern datasets, computing gradients exactly is either infeasible or impractical in ML practice and gradients are often estimated stochastically based on mini-batches (randomly sampled subset of data points) as in the case of stochastic gradient-type algorithms.

There is a growing literature on the WCSC and WCMC problems in the stochastic setting. Several metrics for quantifying the quality of an approximate solution to (1) have been proposed in the literature. A common way to assess the performance is to define the *primal function* $\phi(\cdot) \triangleq \max_{y \in \mathcal{Y}} \mathcal{L}(\cdot, y)$ and measure the violation of first-order necessary conditions for the non-convex problem $\min_{x \in \mathcal{X}} \phi(x)$. Given the primal iterate sequence $\{x_k\}_{k \geq 0}$ of a stochastic SP algorithm and a threshold $\epsilon > 0$, a commonly used metric is the *gradient norm of the Moreau envelope* (GNME); indeed, the objective is to provide a bound K_ϵ such that $\mathbb{E}[\|\phi_\lambda(x_k)\|] \leq \epsilon$ for all $k \geq K_\epsilon$, where ϕ_λ denotes the Moreau envelope of the primal function ϕ –see Definitions 3, 4 and 5. Another commonly used natural metric is the *gradient norm of the primal function* $\phi(\cdot)$ [4, 16, 15, 26, 37], abbreviated as GNP, where the aim is to derive K_ϵ such that $\mathbb{E}[\|\nabla \phi(x_k)\|] \leq \epsilon$ for all $k \geq K_\epsilon$. Other metrics such as the notion of ϵ-first-order Nash equilibrium (FNE) and its generalized versions also exist in the literature [33, 34]. When using any of these metrics, the ultimate goal is to establish a bound on the oracle (sampling) complexity, i.e., $\sum_{k=0}^{K_\epsilon} b_k$, where b_k denotes the batch-size for iteration $k \geq 0$. It crucial to note for the WCSC setting that GNME, GNP and FNE metrics are all equivalent in the sense that convergence in either of them implies convergence in the other for WCSC problems [22]. In this paper, for the WCSC setting, we adopt both GNME and GNP as the main performance metrics to analyze our algorithms; indeed, in Theorem 2 we show that, when the non-smooth part $f(\cdot) = 0$, we can convert a GNME guarantee to a GNP guarantee by incurring only little additional cost compared to the cost of obtaining a GNME guarantee such that the overall worst-case complexity (in terms of worst-case dependency to the target accuracy ϵ) remains the same for both metrics. When the non-smooth part $f(\cdot) \neq 0$, we also obtain similar guarantees and show equivalence between the metrics based on GNME and the generalized gradient mapping. On the other hand, for the WCMC setting, we provide our guarantees in GNME metric as ϕ is not necessarily differentiable for this scenario. Moreover, our work accounts for the individual effects of L_{xx}, L_{xy}, L_{yx} and L_{yy}, i.e., the Lipschitz constants of $\nabla_x \Phi(\cdot, y)$, $\nabla_x \Phi(x, \cdot)$, $\nabla_y \Phi(x, \cdot)$ and $\nabla_y \Phi(\cdot, y)$ (see Assumption 2), respectively, instead of using the worst-case parameters $L \triangleq \max\{L_{xx}, L_{xy}, L_{yx}, L_{yy}\}$, while the majority of related work ignore the influence of these block Lipschitz constants to be able to analyze their algorithms. We emphasize that using the worst-case parameters will lead to a theoretically conservative step sizes, and this phenomenon has been validated in the work [42].

Contributions. Table 1 summarizes the relevant existing work for WCSC and WCMC problems closest to our setting and our results. More specifically, in Table 1, for the stochastic setting, we report the (oracle) complexity with respect to the GNP and GNME as the performance metrics for WCSC and WCMC problems, respectively, and the batch-size (number of data points in the mini-batches) required at every iteration. We also report whether the method is based on a variance-reduction (VR) technique. VR-based methods mentioned in Table 1 use a small batch-size b' in general, while using a large batch-size $b \geq b'$ once in every q iterations for reducing the variance where the period q is the number of times small batches are sampled consecutively plus one, is also an algorithm parameter. Therefore, for VR-methods, we report the batch size as a triplet (b', b, q). In the column "Compactness", we list whether achieving the specific complexity requires assuming compactness of the primal and/or dual domains. To make the comparison of our results with the existing work easier, we provide the results in the table for the worst-case setting, where $\kappa_y \triangleq \frac{L}{\mu_y}$, and we report the ϵ-, κ_y- and L-dependency of the complexity results for the existing algorithms. That being said, our results have finer granularity in terms of their dependence to the individual effects of L_{xx}, L_{xy}, L_{yx} and L_{yy} as we mentioned earlier. Our contributions (also summarized in section 1) are as follows:

- We propose a new stochastic method, `SAPD+`, based on the inexact proximal point method (iPPM). In this framework, one inexactly solves strongly convex-strongly concave (SCSC) saddle point sub-problems using an accelerated primal-dual method, SAPD [42]. In Theorem 3, we establish an oracle complexity of $\mathcal{O}(L\kappa_y\epsilon^{-4})$ for WCSC problems, and unlike the majority of existing work we do not require compactness for neither the primal nor the dual domain. To our knowledge, our bound has the best κ dependence in the literature; indeed, prior to this work, without using variance reduction, the best known complexity was $\mathcal{O}(L\kappa^3\epsilon^{-4})$; hence, we establish a $\mathcal{O}(\kappa^2)$ improvement.
- We propose a variance-reduced version of `SAPD+` in Theorem 4. For WCSC setting, `SAPD+` using variance reduction achieves an oracle complexity of $\mathcal{O}(L\kappa_y^2\epsilon^{-3})$ –this bound has the best ϵ-dependency in the literature to our knowledge, and among all the methods with the $\mathcal{O}(\epsilon^{-3})$

Ref.	Complexity	Compactness	VR-based	Batchsize
Weakly Convex-Strongly Concave (WCSC) problems				
*Rafique *et al.* [35]	$\mathcal{O}(\epsilon^{-4}\log(\epsilon^{-1}))$	(n, n)	✗	$\mathcal{O}(1)$
†Yan *et al.* [39]	$\mathcal{O}(\epsilon^{-4}\log(\epsilon^{-1}))$	(y, y)	✗	$\mathcal{O}(1)$
Lin *et al.* [22]	$\mathcal{O}(L\kappa_y^3\epsilon^{-4})$	(n, y)	✗	$\mathcal{O}(\kappa_y\epsilon^{-2})$
Bot and B̈ohm [4]	$\mathcal{O}(L\kappa_y^3\epsilon^{-4})$	(n, n)	✗	$\mathcal{O}(\kappa_y\epsilon^{-2})$
‡Huang *et al.* [16]	$\mathcal{O}(\kappa_y^5\mu_y^{-1}\epsilon^{-3})$	(n, n)	✓	$\mathcal{O}(\kappa_y\epsilon^{-1}),\ \mathcal{O}(\kappa_y^2\epsilon^{-2}),\ \mathcal{O}(\kappa_y\epsilon^{-1})$
§Huang *et al.* [15]	$\mathcal{O}(L^{1.5}\kappa_y^{3.5}\epsilon^{-3})$	(y, y)	✓	$\mathcal{O}(\sqrt{\kappa_y})$
Luo *et al.* [26]	$\mathcal{O}(L\kappa_y^3\epsilon^{-3})$	(y, y)	✓	$\mathcal{O}(\kappa_y\epsilon^{-1}),\ \mathcal{O}(\kappa_y^2\epsilon^{-2}),\ \mathcal{O}(\kappa_y\epsilon^{-1})$
Xu *et al.* [37]	$\mathcal{O}(L\kappa_y^3\epsilon^{-3})$	(y, y)	✓	$\mathcal{O}(\kappa_y\epsilon^{-1}),\ \mathcal{O}(\kappa_y^2\epsilon^{-2}),\ \mathcal{O}(\kappa_y\epsilon^{-1})$
SAPD+, Theorem 3	$\mathcal{O}(L\kappa_y\epsilon^{-4})$	(n, n)	✗	$\mathcal{O}(1)$
SAPD+, Theorem 4	$\mathcal{O}(L\kappa_y^2\epsilon^{-3})$	(n, n)	✓	$\mathcal{O}(\kappa_y\epsilon^{-1}),\ \mathcal{O}(\kappa_y\epsilon^{-2}),\ \mathcal{O}(\epsilon^{-1})$
Weakly Convex-Merely Concave (WCMC) problems				
Rafique *et al.* [35]	$\mathcal{O}(L^3\epsilon^{-6}\log^3(L\epsilon^{-2}))$	(y, y)	-	$\mathcal{O}(1)$
Bot and B̈ohm [4]	$\mathcal{O}(L^5\epsilon^{-8})$	(n, y)	-	$\mathcal{O}(1)$
Lin *et al.* [22]	$\mathcal{O}(L^3\epsilon^{-8})$	(n, y)	-	$\mathcal{O}(1)$
SAPD+, Theorem 5	$\mathcal{O}(L^3\epsilon^{-6})$	(n, y)	-	$\mathcal{O}(1)$

Table 1: Summary of relevant work for WCSC and WCMC problems. For the column "Compactness", we use y,n to represent requiring and not requiring compactness, respectively; the first argument is for primal domain and the second is for dual domain. For batchsize, we use (b', b, q) format for VR-based methods; from left to right, *small batch* (b'), *large batch* (b), *frequency* (q).

Table notes: *For WCSC setting, [35] assumes $\Phi(\cdot, y) \triangleq c^\top(\cdot)y$ is weakly convex and $g(\cdot)$ is strongly convex. † In [39], $\mathcal{L} = \Phi$ and Φ need not be smooth, rather second moment of stochastic subgradients is assumed to be uniformly bounded. If Φ is L-smooth, naively assuming $\Phi(\cdot, y)$ and $\Phi(x, \cdot)$ are L_Φ-Lipschitz, then their result implies $\mathcal{O}(L_\Phi^2\kappa_y^2\epsilon^{-4}\log^2(\sqrt{\kappa_y}L_\Phi/\epsilon))$. ‡,§ The complexity result reported here are different than those in [16, 15]. The issues in their proofs leading to the wrong complexity results are explained in appendix I .

complexity, our approach has the best condition number, κ_y, dependency; indeed, prior to this work, the best known complexity was $\mathcal{O}(L\kappa_y^3\epsilon^{-3})$; hence, we establish $\mathcal{O}(\kappa_y)$ factor improvement.

- For the WCMC case, our proposed algorithm SAPD+ results in $\mathcal{O}(L^3\epsilon^{-6})$ complexity, which is the best to our knowledge, improving the best known complexity by $\log^3(L/\epsilon^2)$ factor.
- Finally, we demonstrate the efficiency of SAPD+ on a distributionally robust learning problem with a weakly convex cost and also on a (worst-case) multi-class classification problem in deep learning.

Notation. Throughout the paper, $\|\cdot\|$ denotes the Euclidean norm. Given $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ a closed convex function, $\mathbf{prox}_{\lambda f}(x) \triangleq \operatorname{argmin}_w f(w) + \frac{1}{2\lambda}\|w - x\|^2$ denotes the proximal map of f. Given random ω, we let $\tilde{\nabla}_x\Phi(x, y; \omega)$ and $\tilde{\nabla}_y\Phi(x, y; \omega)$ to be unbiased estimators of $\nabla\Phi_x(x, y)$ and $\nabla\Phi_y(x, y)$. Moreover, given a random mini-batch $\mathcal{B} = \{\omega_i\}_{i=1}^b$, we let $\tilde{\nabla}_x\Phi_{\mathcal{B}}(x, y) = \frac{1}{b}\sum_{i=1}^b \tilde{\nabla}_x\Phi(x, y; \omega_i)$ to denote the stochastic gradient estimate based on the batch $\mathcal{B}$, and we define $\tilde{\nabla}_y\Phi_{\mathcal{B}}(\cdot, \cdot)$ similarly.

2 Preliminaries

We start with describing the notion of weak convexity.

Definition 1. *$f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is γ-weakly convex if $x \mapsto f(x) + \frac{\gamma}{2}\|x\|^2$ is convex.*

Definition 2. *A differentiable function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is L-smooth if $\exists L > 0$ such that for $\forall x, x' \in \mathbf{dom}\, f$, $\|\nabla f(x) - \nabla f(x')\| \le L\|x - x'\|$.*

Remark 1. *If a function is L-smooth, then it is also L-weakly convex.*

Remark 1 shows that weak convexity is a rich class containing the class of smooth functions. In the rest of the paper, we consider the SP problem in (1). Next, we introduce our assumptions.

Assumption 1. *$f : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ and $g : \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ are proper, closed, convex functions. Let $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be such that (i) for any $y \in \mathbf{dom}\, g \subset \mathcal{Y}$, $\Phi(\cdot, y)$ is γ-weakly convex and bounded from below; (ii) for any $x \in \mathbf{dom}\, f \subset \mathcal{X}$, $\Phi(x, \cdot)$ is μ_y-strongly concave for some $\mu_y \ge 0$; (iii) Φ is differentiable on an open set containing $\mathbf{dom}\, f \times \mathbf{dom}\, g$.*

Assumption 2. *There exist $L_{xx}, L_{yy} \ge 0$, $L_{xy}, L_{yx} > 0$ such that $\|\nabla_x\Phi(x, y) - \nabla_x\Phi(\bar{x}, \bar{y})\| \le L_{xx}\|x - \bar{x}\| + L_{xy}\|y - \bar{y}\|$, and $\|\nabla_y\Phi(x, y) - \nabla_y\Phi(\bar{x}, \bar{y})\| \le L_{yx}\|x - \bar{x}\| + L_{yy}\|y - \bar{y}\|$ for all $x, \bar{x} \in \mathbf{dom}\, f \subset \mathcal{X}$, and $y, \bar{y} \in \mathbf{dom}\, g \subset \mathcal{Y}$.*

Assumption 1 allows non-convexity in x while requiring (strong) concavity in the y variable. Assumption 2 is standard in the analysis of first-order methods for solving SP problems. It should be noticed that when $L_{yx} = L_{xy} = 0$, the problem in (1) can be solved separately for the primal and dual variables; hence, it is natural to assume $L_{yx}, L_{xy} > 0$.

Suppose that we implement SAPD, stated in Algorithm 1, on the SCSC problem

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \mathcal{L}(x,y) + \frac{\mu_x+\gamma}{2}\|x-\bar{x}\|^2 \quad (2)$$

for some given $\mu_x > 0$ and $\bar{x} \in \mathcal{X}$ –strong convexity follows from $\mathcal{L}(\cdot, y)$ being γ-weakly convex. We make the following assumption on the statistical nature of the gradient noise as in, e.g., [5, 10, 42].

Algorithm 1 SAPD Algorithm

Input: $\tau, \sigma, \theta, \mu_x, x_0, y_0, N$
$\tilde{q}_0 \leftarrow 0$
for $k \geq 0$ **do**
  $\tilde{s}_k \leftarrow \tilde{\nabla}_y \Phi(x_k, y_k; \omega_k^y) + \theta \tilde{q}_k$
  $y_{k+1} \leftarrow \mathbf{prox}_{\sigma g}(y_k + \sigma \tilde{s}_k)$
  $x_{k+1} \leftarrow \mathbf{prox}_{\tau f}(x_k - \tau \tilde{\nabla}_x \Phi(x_k, y_{k+1}; \omega_k^x))$
  $\tilde{q}_{k+1} \leftarrow \tilde{\nabla}_y \Phi(x_{k+1}, y_{k+1}; \omega_{k+1}^y) - \tilde{\nabla}_y \Phi(x_k, y_k; \omega_k^y)$
end for
Output: $(\bar{x}_N, \bar{y}_N) = \frac{1}{K_N(\theta)} \sum_{i=0}^{N-1} \theta^{-k}(x_{k+1}, y_{k+1})$

Assumption 3. *Given arbitrary $\bar{x} \in \mathcal{X}$ and $\mu_x > 0$, let $\{(x_k, y_k)\}$ be generated by SAPD running on* (2)*. There exist $\delta_x, \delta_y \geq 0$ such that for all $k \geq 0$, the stochastic gradients $\tilde{\nabla}_x \Phi(x_k, y_{k+1}; \omega_k^x)$, $\tilde{\nabla}_y \Phi(x_k, y_k; \omega_k^y)$ and random sequences $\{\omega_k^x\}_k$, $\{\omega_k^y\}_k$ satisfy the following conditions:*

(i) $\{\omega_k^x\}_k$ *and* $\{\omega_k^y\}_k$ *are independent sequences, also independent from each other;*
(ii) $\mathbb{E}[\tilde{\nabla}_x \Phi(x_k, y_{k+1}; \omega_k^x) | x_k, y_{k+1}] = \nabla_x \Phi(x_k, y_{k+1})$*;*
(iii) $\mathbb{E}[\tilde{\nabla}_y \Phi(x_k, y_k; \omega_k^y) | x_k, y_k] = \nabla_y \Phi(x_k, y_k)$*;*
(iv) $\mathbb{E}[\|\tilde{\nabla}_x \Phi(x_k, y_{k+1}; \omega_k^x) - \nabla_x \Phi(x_k, y_{k+1})\|^2 | x_k, y_{k+1}] \leq \delta_x^2$*;*
(v) $\mathbb{E}[\|\tilde{\nabla}_y \Phi(x_k, y_k; \omega_k^y) - \nabla_y \Phi(x_k, y_k)\|^2 | x_k, y_k] \leq \delta_y^2$*.*

Assumption 3 says that the gradient noise conditioned on the iterates is unbiased with a finite variance, such assumptions are common in the literature, e.g., [5, 10, 42], and can be satisfied when gradients are estimated from randomly sampled data points with replacement. For WCSC minimax problems, a commonly adopted definition for ϵ-stationary is based on Moreau envelope, e.g., see [22, 39]. It is inspired by Davis and Drusvyatskiy's work [8] for solving weakly convex minimization problems. For the sake of completeness, we briefly review this idea below.

Definition 3. *Let $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be γ-weakly convex. Then, for any $\lambda \in (0, \gamma^{-1})$, Moreau envelope of ϕ is defined as $\phi_\lambda : \mathbb{R}^d \to \mathbb{R}$ such that $\phi_\lambda(x) \triangleq \min_{w\in\mathcal{X}} \phi(w) + \frac{1}{2\lambda}\|w - x\|^2$.*

Lemma 1. *Let $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a γ-weakly convex function. For any given $\lambda \in (0, \gamma^{-1})$, $\phi_\lambda(\cdot)$ is well-defined on $\mathcal{X}$. Moreover, $\nabla \phi_\lambda(x) = \lambda^{-1}(x - \mathbf{prox}_{\lambda\phi}(x))$ for $x \in \mathcal{X}$; hence, ϕ_λ is λ^{-1}-smooth, where $\mathbf{prox}_{\lambda\phi}(x) = \operatorname{argmin}_{w\in\mathcal{X}}\{\phi(w) + \frac{1}{2\lambda}\|w - x\|^2\}$.*

Definition 4. *Under Assumption 1, define $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ such that $\phi(x) \triangleq \max_{y\in\mathcal{Y}} \mathcal{L}(x, y)$ for $x \in \mathcal{X}$, i.e., $\phi(x) = f(x) + \phi^s(x)$, where $\phi^s(x) \triangleq \max_{y\in\mathcal{Y}} \Phi(x, y) - g(y)$ for $x \in \mathbf{dom}\, f$.*

Remark 2. *Under Assumption 1, since $\Phi(\cdot, y)$ is γ-weakly convex for any $y \in \mathbf{dom}\, g$, ϕ^s is γ-weakly convex*[1]*; hence, ϕ is also γ-weakly convex. Note that*

$$\mathbf{prox}_{\lambda\phi}(x) = \operatorname*{argmin}_{w\in\mathcal{X}}\{\phi(w) + \tfrac{1}{2\lambda}\|w - x\|^2\} = \operatorname*{argmin}_{w\in\mathcal{X}} \max_{y\in\mathcal{Y}} \mathcal{L}(w, y) + \tfrac{1}{2\lambda}\|w - x\|^2. \quad (3)$$

Furthermore, when $\mu_y > 0$, ϕ^s is differentiable on $\mathbf{dom}\, f$.

In the following definition, we introduce the notion of ϵ-stationary with respect to the GNME metric.

Definition 5. *A point x_ϵ is an ϵ-stationary point of a γ-weakly convex function ϕ if $\|\nabla \phi_\lambda(x_\epsilon)\| \leq \epsilon$ for some $\lambda \in (0, \gamma^{-1})$. If $\epsilon = 0$, then x_ϵ is a stationary point of ϕ.*

Thus, from Lemma 1, computing an ϵ-stationary point x_ϵ for ϕ is equivalent to searching for x_ϵ such that $\|x_\epsilon - \mathbf{prox}_{\lambda\phi}(x_\epsilon)\|$ is small. Recall that for any $\lambda \in (0, \gamma^{-1})$, $\mathbf{prox}_{\lambda\phi}(x)$ is well-defined and unique. We also observe from (3) that $\mathbf{prox}_{\lambda\phi}(\cdot)$ computation is indeed an SCSC SP problem. To compute x_ϵ such that $\|x_\epsilon - \mathbf{prox}_{\lambda\phi}(x_\epsilon)\|$ is small, it is natural to consider the iPPM algorithm (see

[1] One can argue that $\phi^s(\cdot) + \frac{\gamma}{2}\|\cdot\|^2$ is a pointwise supremum of convex functions.

e.g. [17]), which computes $x_0^{t+1} \approx \mathbf{prox}_{\lambda\phi}(x_0^t)$, i.e. proximal steps are "inexactly" computed for $t \geq 0$, starting from an arbitrary given point $x_0^0 \in \mathcal{X}$. Next, we describe SAPD+, an iPPM algorithm employing SAPD to *inexactly* solve the SCSC subproblems arising in the iPPM iterations.

2.1 The proposed algorithm SAPD+ and its analysis

The convergence and robustness properties of SAPD for SCSC SP problems are analyzed in [42]. For the WCSC SP problems, as we explained in the previous subsection, the main idea is to apply the iPPM framework as stated in SAPD+ (see algorithm 2) which requires successively solving SCSC SP problems. In the rest, the counter for iPPM is denoted with $t \in \mathbb{Z}_+$. At each outer iteration $t \geq 1$, we inexactly compute the prox map, i.e., $x_0^{t+1} \approx \mathbf{prox}_{\lambda\phi}(x_0^t)$, which is well-defined for $\lambda \in (0, \gamma^{-1})$; hence, to derive our preliminary results, we fix $\lambda = (\mu_x + \gamma)^{-1}$ for some given $\mu_x > 0$ – thus, $\mathcal{L}(x,y) + \frac{\mu_x+\gamma}{2}\|x - x_0^t\|^2$ is SCSC in (x,y) with moduli (μ_x, μ_y) and has a unique saddle point. To summarize, we will construct $\{x_0^t\}_{t=1}^T \subset \mathbf{dom}\, f$ by inexactly solving the following SCSC SP problem in (4) at each outer iteration $t \in \mathbb{Z}_+$ through running SAPD for $N_t \in \mathbb{Z}_+$ iterations:

Algorithm 2 SAPD+ Algorithm

1: **Input:** $\{\tau, \sigma, \theta, \mu_x\}$, $(x_0^0, y_0^0) \in \mathcal{X} \times \mathcal{Y}$, $\{N_t\}_{t\geq 0} \in \mathbb{Z}^+$
2: **for** $t = 0, 1, 2, ..., T$ **do**
3: **if** VR-flag == **false then**
4: $(x_0^{t+1}, y_0^{t+1}) \leftarrow$ SAPD$(\tau, \sigma, \theta, \mu_x, x_0^t, y_0^t, N_t)$
5: **else**
6: $(x_0^{t+1}, y_0^{t+1}) \leftarrow$ VR-SAPD$(\tau, \sigma, \theta, \mu_x, x_0^t, y_0^t, N_t)$
7: **end if**
8: **end for**

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \mathcal{L}^t(x,y) \triangleq f(x) + \Phi^t(x,y) - g(y), \text{ where } \Phi^t(x,y) = \Phi(x,y) + \frac{\mu_x+\gamma}{2}\|x - x_0^t\|^2, \quad (4)$$

where we will specify $N_t \in \mathbb{Z}_+$ later. Hence, Φ^t is an SCSC function with moduli (μ_x, μ_y).

Next, we briefly explain the main step of SAPD+ with VR-flag=**false**. The statement $(x_0^{t+1}, y_0^{t+1}) \leftarrow$ SAPD$(\tau, \sigma, \theta, x_0^t, y_0^t, N_t)$ in line 4 of algorithm 2 means that (x_0^{t+1}, y_0^{t+1}) is generated using SAPD, as stated in algorithm 1 –indeed, SAPD is run on (4) for N_t iterations with SAPD parameters (τ, σ, θ) and starting from the initial point (x_0^t, y_0^t).

To analyze the convergence of SAPD+, we first define the gap function $\mathcal{G}^t$ for t-th SAPD+ iteration:

$$\mathcal{G}^t(x,y) \triangleq \max_{y'\in\mathcal{Y}} \mathcal{L}^t(x,y') - \min_{x'\in\mathcal{X}} \mathcal{L}^t(x',y). \quad (5)$$

Recall that $\mathcal{L}^t$ is an SCSC function; therefore, *i)* it has a unique saddle point denoted by (x_*^t, y_*^t), and it is important to note that $x_*^t = \mathbf{prox}_{\lambda\phi}(x_0^t)$ for $\phi(x) = \max_{y\in\mathcal{Y}} \mathcal{L}(x,y)$ and $\lambda = (\gamma + \mu_x)^{-1}$; *ii)* for any $(x,y) \in \mathbf{dom}\, f \times \mathbf{dom}\, g$, the following quantities are well-defined:

$$x_*^t(y) \triangleq \operatorname*{argmin}_{x'\in\mathcal{X}} \mathcal{L}^t(x',y), \quad y_*(x) \triangleq \operatorname*{argmax}_{y'\in\mathcal{Y}} \mathcal{L}^t(x,y') = \operatorname*{argmax}_{y'\in\mathcal{Y}} \mathcal{L}(x,y'). \quad (6)$$

Thus, it follows that $\mathcal{G}^t(x,y) = \mathcal{L}^t(x, y_*(x)) - \mathcal{L}^t(x_*^t(y), y)$. Moreover, we also define $\mathcal{G}(x,y) \triangleq \sup_{y'\in\mathcal{Y}} \mathcal{L}(x,y') - \inf_{x'\in\mathcal{X}} \mathcal{L}(x',y)$. Assumption 1 ensures that $\mathcal{G}$ is well defined. We first provide our oracle complexity in the GNME metric under the compactness assumption of the primal-dual domains; later, in section 2.2, we show that under a particular subdifferentiability assumption compactness requirement can be avoided.

Assumption 4. $\mathbf{dom}\, f$ *and* $\mathbf{dom}\, g$ *are compact sets.*

Theorem 1. *Suppose Assumption 1, 2, 3, and 4 hold. Let* $\mu_x = \gamma$, $\{\tau, \sigma, \theta\}$ *and* N *be chosen as*

$$N = \frac{\ln(265)}{\ln(\frac{1}{\theta})} + 1, \quad \tau = \frac{1-\theta}{\gamma}, \quad \sigma = \frac{1-\theta}{\mu_y\theta}, \quad \theta \geq \max\{\bar{\theta}(\beta),\, \bar{\bar{\theta}}_1,\, \bar{\bar{\theta}}_2\}, \quad (7)$$

for $\beta = \min\{\frac{1}{2},\, \frac{\mu_y}{4\gamma},\, \frac{\gamma}{4\mu_y},\, \frac{L_{yx}}{2L_{xy}}\}$, *where* $\bar{\theta}(\beta) \triangleq \max\{\bar{\theta}_1(\beta),\, \bar{\theta}_2(\beta)\} \in (0,1)$ *such that*

$$\bar{\theta}_1(\beta) \triangleq 1 - \frac{\beta\mu_y L'_{xx}}{2L_{yx}^2}\Big(\sqrt{1 + \frac{4L_{yx}^2\mu_x}{\beta {L'_{xx}}^2 \mu_y}} - 1\Big), \quad \bar{\theta}_2(\beta) \triangleq \begin{cases} 1 - \frac{(1-\beta)^2}{8}\frac{\mu_y^2}{L_{yy}^2}\Big(\sqrt{1 + \frac{16L_{yy}^2}{(1-\beta)^2\mu_y^2}} - 1\Big) & L_{yy} > 0 \\ 0 & L_{yy} = 0, \end{cases}$$

$$\bar{\bar{\theta}}_1 = \max\Big\{0, 1 - \frac{1}{384}\cdot\frac{\epsilon^2}{\delta_x^2}\Big\}, \qquad \bar{\bar{\theta}}_2 = \Big(1 + \frac{1}{384}\cdot\frac{\mu_y}{10\gamma}\frac{\epsilon^2}{\delta_y^2}\Big)^{-1}. \quad (8)$$

For $\epsilon > 0$, when `VR-flag=`***false***, `SAPD+` *iterate x_0^T is ϵ-stationary, i.e., $\mathbb{E}\left[\|\nabla\phi_\lambda(x_0^T)\|\right] \leq \epsilon$, for $T \geq 96\mathcal{G}(x_0^0, y_0^0) \cdot \frac{\gamma}{\epsilon^2} + 1$, which requires C_ϵ oracle calls in total where*

$$C_\epsilon = \mathcal{O}\Big(\Big(\tfrac{\max\{L_{xx},L_{yx}\}}{\gamma} + \tfrac{\max\{L_{yx},L_{xy}\}}{\sqrt{\gamma\mu_y}} + \tfrac{\max\{L_{yy},L_{yx}\}}{\mu_y}\Big)\gamma \cdot \epsilon^{-2} + \Big(\tfrac{\delta_x^2}{\gamma} + \tfrac{\delta_y^2}{\mu_y}\Big)\gamma^2 \cdot \epsilon^{-4}\Big)\mathcal{G}(x_0^0, y_0^0).$$

Remark 3. *For any $y \in \mathbf{dom}\, g$, since $\Phi(\cdot, y)$ L_{xx}-smooth, it is necessarily L_{xx}-weakly convex; hence, $\gamma \leq L_{xx}$. To get a worst-case complexity, let*

$$L \triangleq \max\{L_{xy}, L_{yx}, L_{xx}, L_{yy}\},\ \kappa_y \triangleq L/\mu_y,\ \delta \triangleq \max\{\delta_x, \delta_y\},\ \gamma = L. \tag{9}$$

Our oracle complexity C_ϵ in Theorem 1 can be simplified as $C_\epsilon = \mathcal{O}\left(\max\{1, \frac{\delta^2}{\epsilon^2}\}\frac{\kappa_y L\mathcal{G}(x_0^0, y_0^0)}{\epsilon^2}\right)$.

In fact, Li et al. [20] (see also [41]) provide a lower complexity bound for a class of first-order stochastic algorithms that do not use variance reduction. The lower bound for finding ϵ-stationary points of smooth WCSC problems in GNP metric is $\Omega(L\Delta_\phi(\sqrt{\kappa_y}\epsilon^{-2} + \kappa_y^{\frac{1}{3}}\epsilon^{-4}))$, where $\Delta_\phi = \phi(x_0) - \min_{x\in\mathcal{X}} \phi(x)$ with x_0 the initial point.

Consider $\phi = f + \phi^s$ as given in definition 4. For $\lambda > 0$, the map $G_\lambda : \mathbb{R}^d \to \mathbb{R}^d$ defined as

$$G_\lambda(\tilde{x}) \triangleq \frac{1}{\lambda}[\tilde{x} - \mathbf{prox}_{\lambda f}(\tilde{x} - \lambda\nabla\phi^s(\tilde{x}))]$$

is called the *generalized gradient mapping* and its norm is frequently used in optimization for assessing stationarity (see e.g. [9]). Theorem 1 provides guarantees in the GNME metric. Theorem 2 shows that given x_ϵ, an ϵ-stationary point in GNME metric (see definition 5), we can generate $\tilde{x}$ such that $\|G_\lambda(\tilde{x})\| \leq \epsilon$ for some $\lambda > 0$, i.e., an ϵ-stationary point in *generalized gradient mapping* metric, within $\tilde{\mathcal{O}}(1/\epsilon^2)$ `SAPD` iterations. When $f(\cdot) = 0$, this metric becomes equivalent to the GNP metric.

Theorem 2. *Suppose Assumptions 1, 2, 3 hold. Given x_ϵ, an ϵ-stationary point for the γ-weakly convex function $\phi(\cdot) = \max_{y\in\mathcal{Y}} \mathcal{L}(\cdot, y)$, i.e., $\|\nabla\phi_\lambda(x_\epsilon)\| \leq \frac{\epsilon}{2}$ for some fixed $\lambda \in (0, \gamma^{-1})$. Let $\phi^s(\cdot) = \max_{y\in\mathcal{Y}} \Phi(\cdot, y) - g(y)$ so that $\phi = f + \phi^s$. Then, initialized from x_ϵ, within $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ stochastic first-order oracle calls,* `SAPD`*, stated in algorithm 1, can generate $\tilde{x}$ such that $\mathbb{E}\big[\|G_\lambda(\tilde{x})\| \leq \epsilon$.*

Remark 4. *Suppose* `SAPD+` *requires T_ϵ iterations to output x_ϵ such that $\|\nabla\phi_\lambda(x_\epsilon)\| \leq \frac{\epsilon}{2}$. Then generating $\tilde{x}$ from x_ϵ such that $\mathbb{E}\big[\|G_\lambda(\tilde{x})\| \leq \epsilon$ requires an additional* `SAPD+` *iteration, i.e., $T_\epsilon + 1$ iterations in total, where in the last iteration we run* `SAPD` *for $\tilde{\mathcal{O}}(\epsilon^{-2})$ iterations rather than N.*

2.2 Relaxing the compactness assumption

In the previous section, we assume that $\mathbf{dom}\, f$ and $\mathbf{dom}\, g$ are compact sets, e.g., $f(\cdot) = \mathbb{1}_X(\cdot)$ and $g(\cdot) = \mathbb{1}_Y(\cdot)$, where $X \subset \mathcal{X}$ and $Y \subset \mathcal{Y}$ are compact convex sets. In this section, we show that `SAPD+` can also handle unbounded domains under the following assumption.

Assumption 5. *For f and g closed convex, suppose $\exists B_f,\ B_g > 0$ such that $\inf\{\|s_f\| :\ s_f \in \partial f(x)\} \leq B_f$ for all $x \in \mathbf{dom}\, f$ and $\inf\{\|s_g\| :\ s_g \in \partial g(y)\} \leq B_g$ for all $y \in \mathbf{dom}\, g$.*

Remark 5. *Assumption 5 holds when f is an indicator function of a closed convex set (not necessarily bounded) or for $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ such that $\mathbf{dom}\, f$ is open and f is Lipschitz. Two important examples for this scenario: (i) $f(\cdot) = 0$, (ii) f is a norm, e.g., ℓ_1-, ℓ_2-, or the Nuclear norms.*

The existing work based on iPPM framework either require compactness, e.g., [39], or some special structure on $\mathcal{L}$, e.g., [35]. This is also true for VR-based methods, e.g.,[15, 26, 37]. To our knowledge, ours is the first one to overcome this difficulty and strictly improve the complexity bounds for WCSC and WCMC settings without compactness assumption; moreover, the same idea also works simultaneously with a variance reduction technique that will be discussed later (see section 3). Finally, same trick for removing compactness assumption for WCSC setting also helps removing compactness assumption for the primal domain in WCMC setting as well (see section 4).

Remark 6. *In [22], when $f = g = 0$, boundedness of dual space is required while Assumption 5 is a weaker requirement. Furthermore, based on the discussion with the authors of [39], one also needs compactness of the domain for their proof to hold. In [16], the authors require the sub-level set $\{x : \phi(x) + f(x) \leq \alpha\}$ to be compact for all $\alpha > 0$. There are simple convex functions that do not satisfy this condition such as $f(x) = \max\{0, x\}$. Bot and B̈ohm [4] use milder assumptions than [22] without requiring compactness; however, their complexity is the same as the complexity of [22].*

Theorem 3. *The result of Theorem 1 continues to hold, if one replaces the compact domain assumption, i.e., Assumption 4, with Assumption 5.*

Finally, it is crucial to note that for smooth problems, i.e., $f(\cdot) = g(\cdot) = 0$, the variance term for `SAPD` error bounds can be greatly simplified. Due to limited space, we provide the details in appendix E.

3 Variance reduction

In this section, we further strength our algorithm with SPIDER variance reduction technique [11], a variant of SARAH [31, 32], as stated in algorithm 3. Indeed, we use a large batchsize of b in every q iterations and use small batchsizes of b_x and b_y for the rest. We prove that `SAPD+` using variance reduction, i.e., with `VR-flag`=**true**, achieves an oracle complexity of $\mathcal{O}(L\kappa_y^2\epsilon^{-3})$; hence, we provide an $\mathcal{O}(\kappa_y)$ factor improvement over the best known complexity in the literature to our knowledge.

Algorithm 3 `VR-SAPD` Algorithm

Input: $\tau, \sigma, \theta, \mu_x, x_0, y_0, N, b, b_x, b_y, q$
Let $\mathcal{B}_0$ be random mini-batch sample with $|\mathcal{B}_0| = b$
$\tilde{w}_0 \leftarrow \tilde{\nabla}_y \Phi_{\mathcal{B}_0}(x_0, y_0), \quad \tilde{s}_0 \leftarrow w_0$
for $k \geq 0$ **do**
  $y_{k+1} \leftarrow \mathbf{prox}_{\sigma g}(y_k + \sigma \tilde{s}_k)$
  if $\mathrm{mod}(k, q) == 0$ **then**
    $v_k \leftarrow \tilde{\nabla}_x \Phi_{\mathcal{B}_k}(x_k, y_{k+1})$
  else
    Let $\mathcal{I}_k^x$ be random mini-batch sample with $|\mathcal{I}_k^x| = b_x$
    $v_k \leftarrow \tilde{\nabla}_x \Phi_{\mathcal{I}_k^x}(x_k, y_{k+1}) - \tilde{\nabla}_x \Phi_{\mathcal{I}_k^x}(x_{k-1}, y_k) + v_{k-1}$
  end if
  $x_{k+1} \leftarrow \mathbf{prox}_{\tau f}(x_k - \tau v_k)$
  Let $\mathcal{B}_{k+1}$ be mini-batch sample with $|\mathcal{B}_{k+1}| = b$
  if $\mathrm{mod}(k+1, q) == 0$ **then**
    $w_{k+1} \leftarrow \tilde{\nabla}_y \Phi_{\mathcal{B}_{k+1}}(x_{k+1}, y_{k+1})$
  else
    Let $\mathcal{I}_{k+1}^y$ be mini-batch sample with $|\mathcal{I}_{k+1}^y| = b_y$
    $\tilde{q}_{k+1} \leftarrow \tilde{\nabla}_y \Phi_{\mathcal{I}_{k+1}^y}(x_{k+1}, y_{k+1}) - \nabla_y \tilde{\Phi}_{\mathcal{I}_{k+1}^y}(x_k, y_k)$
    $w_{k+1} \leftarrow w_k + \tilde{q}_{k+1}$
  end if
  $\tilde{s}_{k+1} \leftarrow (1+\theta) w_{k+1} - \theta w_k$
end for
Output: $(\bar{x}_N, \bar{y}_N) = \frac{1}{K_N(\theta)} \sum_{i=0}^{N-1} \theta^{-k} (x_{k+1}, y_{k+1})$

Here, we use $\tilde{\nabla}_y \Phi^t_{\mathcal{B}_k}(x_k, y_k)$ to represent $\frac{1}{|\mathcal{B}_k|} \sum_{\omega_k^i \in \mathcal{B}_k} \tilde{\nabla}_y \Phi(x_k, y_y; \omega_k^i)$, where $\mathcal{B}_k = \{\omega_k^i\}_{i=1}^b$ is the mini-batch with $|\mathcal{B}_k| = b$ and we define $\tilde{\nabla}_x \Phi^t_{\mathcal{B}_k}(x_k, y_{k+1})$ similarly; in addition, $\mathcal{I}_k^x = \{\omega_k^{x,i}\}$ and $\mathcal{I}_k^y = \{\omega_k^{y,i}\}$ with $|\mathcal{I}_k^x| = b_x$ and $|\mathcal{I}_k^y| = b_y$ denote the small mini-batches we use for generating $\tilde{\nabla}_y \Phi^t_{\mathcal{I}_k^y}(x_k, y_k)$ and $\tilde{\nabla}_x \Phi^t_{\mathcal{I}_k^x}(x_k, y_{k+1})$. Moreover, Throughout this section we make the following continuity assumption on the stochastic oracle similar to [16, 15, 26, 37].

Assumption 6. *$\exists L_{xx}, L_{xy}, L_{yx}, L_{yy} \geq 0$ such that $\forall x, \bar{x} \in \mathbf{dom}\, f \subset \mathcal{X}$ and $\forall y, \bar{y} \in \mathbf{dom}\, g \subset \mathcal{Y}$,*

$$\begin{aligned} \|\tilde{\nabla}_y \Phi(x, y; \omega) - \tilde{\nabla}_y \Phi(\bar{x}, \bar{y}; \omega)\| &\leq L_{yx}\|x - \bar{x}\| + L_{yy}\|y - \bar{y}\|, \quad w.p.\ 1, \\ \|\tilde{\nabla}_x \Phi(x, y; \omega) - \tilde{\nabla}_x \Phi(\bar{x}, \bar{y}; \omega)\| &\leq L_{xx}\|x - \bar{x}\| + L_{xy}\|y - \bar{y}\|, \quad w.p.\ 1. \end{aligned} \tag{10}$$

Assumption 7. *Consider `SAPD+` with `VR-flag` = **true**. For any $k \geq 0$, $\mathcal{I}_k^x$ is independent of (x_i, y_{i+1}) and $\mathcal{I}_k^y$ is independent of (x_i, y_i) for $i \in \{k-1, k\}$.*

Remark 7. *For finite-sum type problems of the form $\min_x \max_y \frac{1}{n} \sum_{i=1}^n \Phi_i(x, y)$, we can set the stochastic gradient according to $\tilde{\nabla}_x \Phi(x, y; \omega) = \nabla_x \Phi_\omega(x, y)$ and $\tilde{\nabla}_y \Phi(x, y; \omega) = \nabla_y \Phi_\omega(x, y)$ where ω is uniformly drawn at random from $\{1, \ldots, n\}$. Therefore, if batch samples $\mathcal{I}_k^x, \mathcal{I}_k^y$ are drawn from $\{1, \ldots, n\}$ uniformly at random with replacement; batches will be independent of the past iterates satisfying Assumption 7.*

Theorem 4. *Suppose Assumption 1,3,6 and 7 hold. Moreover, either Assumption 4 or Assumption 5 holds. Fix $\theta = 1$, and let τ, σ, b and N be chosen as follows:*

$$\tau = \left(L_{yx} + L'_{xx} + 2(q-1)\Big(\frac{{L'_{xx}}^2}{\gamma b_x} + \frac{10 L_{yx}^2}{\mu_y b_y} \Big) \right)^{-1}, \ \sigma = \left(2L_{yy} + L_{yx} + 2(q-1)\Big(\frac{L_{xy}^2}{\gamma b_x} + \frac{10 L_{yy}^2}{\mu_y b_y} \Big) \right)^{-1},$$

$$N = 2(1+\zeta) \max\Big\{ \frac{1}{\gamma\tau} - 1, \ \frac{1}{\mu_y \sigma} \Big\}, \quad b \geq \Big\lceil \max\Big\{ 144\delta_x^2, \ 360\delta_y^2 \frac{\gamma}{\mu_y} \Big\} \frac{1}{\epsilon^2} \Big\rceil, \tag{11}$$

*where $L'_{xx} = L_{xx} + 2\gamma$. For $\epsilon > 0$, and any $b_x, b_y, q \in \mathbb{N}^+$, x_0^T generated by `SAPD+` with `VR-flag` = **true** is ϵ-stationary, i.e., $\mathbb{E}\left[\|\nabla \phi_\lambda(x_0^T)\|\right] \leq \epsilon$, for $T \geq 288\mathcal{G}(x_0^0, y_0^0) \cdot \frac{\gamma}{\epsilon^2}$, which requires $T(Nb/q + N(b_x + b_y))$ oracle calls in total, where*

$$N = \mathcal{O}\Big(\max\Big\{ \frac{L_{yx} + L_{xx}}{\gamma} + \frac{q}{b_x}\frac{L_{xx}^2}{\gamma^2} + \frac{q}{b_y}\frac{L_{yx}^2}{\gamma\mu_y}, \ \frac{L_{yy} + L_{yx}}{\mu_y} + \frac{q}{b_y}\frac{L_{yy}^2}{\mu_y^2} + \frac{q}{b_x}\frac{L_{xy}^2}{\gamma\mu_y} \Big\} \Big). \tag{12}$$

Remark 8. *For any $y \in \mathbf{dom}\, g$, since $\Phi(\cdot, y)$ L_{xx}-smooth, it is necessarily L_{xx}-weakly convex; hence, $\gamma \leq L_{xx}$. To get a worst-case complexity, consider the setting in (9), and let $b_x = b_y = b'$. Then, Theorem 4 implies that setting $b = \mathcal{O}\Big(\kappa_y \frac{\delta^2}{\epsilon^2}\Big)$, $N = \mathcal{O}\big(\kappa_y + \kappa_y^2 \frac{q}{b'}\big)$, and $T = \mathcal{O}\Big(\frac{L\mathcal{G}(x_0^0, y_0^0)}{\epsilon^2}\Big)$ leads to $Nb/q + Nb' = \mathcal{O}\Big(\kappa_y \frac{b}{q} + \kappa_y^2 \frac{b}{b'} + \kappa_y b' + \kappa_y^2 q\Big)$. Thus, setting $q = \sqrt{\frac{b}{\kappa_y}}$ and $b' = \sqrt{b\kappa_y}$ leads to the oracle complexity of $T(Nb/q + Nb') = \mathcal{O}\Big(\max\Big\{1, \frac{\delta}{\epsilon}\Big\} \kappa_y^2 \cdot \frac{L\mathcal{G}(x_0^0, y_0^0)}{\epsilon^2} \Big)$.*

4 Weakly convex-merely concave (WCMC) problems

In this section, we state the convergence guarantees of SAPD+ for solving WCMC problems. In particular, we will consider (1) such that $f(\cdot) = 0$ and $\mu_y = 0$, i.e., $\Phi(x, \cdot)$ is merely concave for all $x \in \mathcal{X}$. Instead of directly solving (1) in WCMC setting, we will solve an approximate model obtained by smoothing the primal problem in a similar spirit to the technique in [30]. More precisely, we approximate (1) with the following WCSC problem: given an arbitrary $\hat{y} \in \mathbf{dom}\, g$, consider

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \hat{\mathcal{L}}(x,y) \triangleq \hat{\Phi}(x,y) - g(y), \quad \text{where} \quad \hat{\Phi}(x,y) \triangleq \Phi(x,y) - \frac{\hat{\mu}_y}{2}\|y-\hat{y}\|^2. \tag{13}$$

Theorem 5. *Under Assumptions 1, 2, 3, consider* (1) *such that* $f(\cdot) \equiv 0$, $\mu_y = 0$, *and* $\mathcal{D}_{\mathcal{Y}} \triangleq \sup_{y_1,y_2\in\mathbf{dom}\, g} \|y_1 - y_2\| < \infty$. *When either Assumption 4 or Assumption 5 holds, for any given* $\epsilon > 0$, *SAPD+ with VR-flag = **false**, applied to* (13) *with* $\hat{\mu}_y = \Theta(\epsilon^2/(L\mathcal{D}_y^2))$, *is guaranteed to generate* $x_\epsilon \in \mathcal{X}$ *such that* $\mathbb{E}\left[\|\nabla\phi_\lambda(x_\epsilon)\|\right] \le \epsilon$ *for* $\lambda = 1/(2\gamma)$ *within* $\mathcal{O}(L^3\epsilon^{-6})$ *stochastic first-order oracle calls.*

5 Numerical experiments

The experiments are conducted on a PC with 3.6 GHz Intel Core i7 CPU and NVIDIA RTX2070 GPU. We consider distributionally robust optimization and fair classification. In the rest, n and d represent the number of samples in the dataset and the dimension of each data point, respectively.

Distributionally Robust Optimization (DRO). First, we consider nonconvex-regularized variant of DRO problem [1, 28, 19, 26, 42, 40] which arises in distributionally robust learning. Let $\{\mathbf{a}_i, b_i\}_{i=1}^n$ be the dataset where $\mathbf{a}_i \in \mathbb{R}^d$ are the features and $b_i \in \{-1, 1\}$ are labels. The DRO problem is

$$\text{(DRO):} \min_{x\in\mathcal{R}^d} \max_{y\in Y} \frac{1}{n}\sum_{i=1}^n y_i \ell_i(x) + f(x) - g(y), \tag{14}$$

where $\ell_i(x) = \log(1 + \exp -b_i \mathbf{a}_i^\top \mathbf{x})$, $f(x) = \eta_1 \sum_{i=1}^d \frac{\alpha x_i^2}{1+\alpha x_i^2}$ is a nonconvex regularizer [2], $g(y) = \frac{1}{2}\eta_2\|ny - \mathbf{1}\|^2$, and $Y \triangleq \{y \in \mathbb{R}_+^d : \mathbf{1}^\top y = 1\}$ – here, $\mathbf{1}$ denotes the vector with all entries equal to one. This problem can be viewed as a robust formulation of empirical risk minimization where the weights y_i are allowed to deviate from $1/n$; and the aim is to minimize the worst-case empirical risk. We perform experiments on three data sets: $i)$ a9a with $n = 32561$; $d = 123$; $ii)$ gisette with $n = 6000$; $d = 5000$; $iii)$ sido0 with $n = 12678$; $d = 4932$. The dataset sido0 is obtained from Causality Workbench[2] while the others can be downloaded from LIBSVM repository[3].

Parameter tuning. We set the parameters according to [40, 26, 19], i.e., , $\alpha = 10$, $\eta_1 = 10^{-3}$, $\eta_2 = 1/n^2$. We compare SAPD, SAPD-VR against PASGDA [4], SREDA [26], SMDA, SMDA-VR [16] algorithms. As suggested in [26], we tune the primal stepsizes of all the algorithms based on a grid-search over the set $\{10^{-3}, 10^{-2}, 10^{-1}\}$ and the ratio of the primal stepsize versus dual stepsize is varied to take values in the set $\{10, 10^2, 10^3, 10^4\}$. For all variance reduction-based algorithms, i.e., for SAPD-VR, SREDA, SMDA-VR, we tune the large batch size $b \triangleq |\mathcal{B}|$ from the set $\{3000, 6000\}$, and the small batch size $b' \triangleq |I|$ from grid search over the set $\{10, 100, 200\}$. For the frequency parameter q, we let $q = b' = |I|$ for SAPD-VR and SMDA-VR (as suggested in [16]); for SREDA, when we set q and m (SREDA's inner loop iteration number) to $\mathcal{O}(n/|I|)$ as suggested in [26], we noticed that SREDA does not perform well against SAPD-VR and SMDA-VR. Therefore, to optimize the performance of SREDA further, we tune q, m from a grid search over $\{10, 100, 200\}$. For methods without variance reduction, i.e., for SAPD, SMDA and PASGDA, we tune the batch size from $\{10, 100, 200\}$ as well. For SAPD and SAPD-VR, we tune the momentum θ from $\{0.8, 0.85, 0.9\}$ and let the inner iteration numbers $N = \frac{\ln(265)}{\ln(\frac{1}{\theta})}$ according to eq. (7).

Results. In fig. 1, we plot the average loss against the epoch number based on 30 simulations (runs). The standard deviations of the runs are also illustrated around the average in lighter color as shaded regions. We observe that SAPD and SAPD-VR consistently outperforms over other algorithms. For a9a, gisette, sido0 datasets, the average training accuracy of SAPD are 84.06%, 95.41%, 96.43%, and of SAPD-VR are 84.33%, 97.69%, 97.46%, respectively. The best performance for a9a, gisette, sido0 among all the other algorithms are 75.92%, 93.07%, 96.43%, respectively. More importantly, we observe that as an accelerated method, SAPD-VR enjoys fast convergence properties while still being robust to gradient noise.

[2] http://www.causality.inf.ethz.ch/challenge.php?page=datasets

[3] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html

Figure 1: Comparison of SAPD, SAPD-VR against PASGDA [4], SREDA [26], SMDA, SMDA-VR [16] on real-data for solving eq. (14) with 30 times simulation.

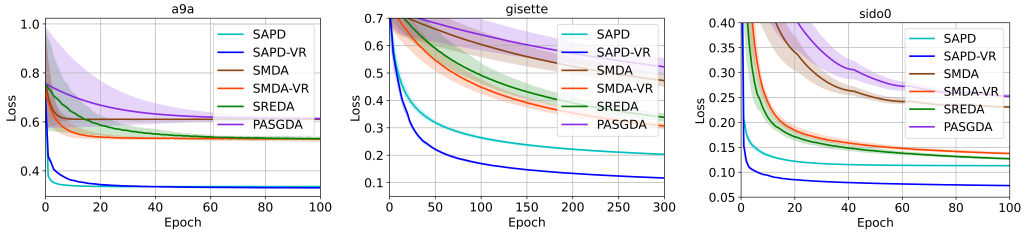
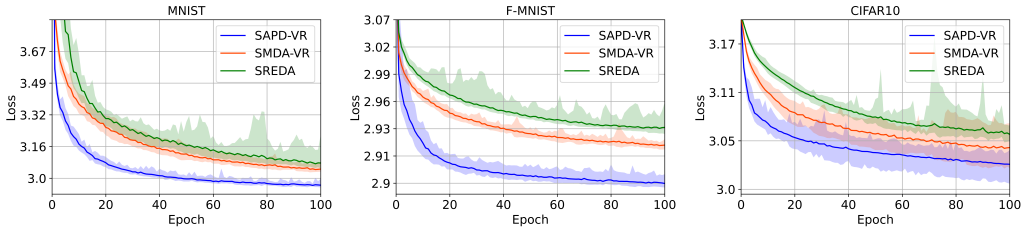


Figure 2: Comparison of SAPD-VR against other Variance Reduction algorithms, SREDA [26], SMDA-VR [16] on real-data for solving eq. (15) with 30 times simulation.



Fair Classification. In the context of multi-class classification, Mohri *et al.* [27] propose training a fair classifier thorough minimizing the worst-case loss over the classification categories. In the spirit of [33, 16], we adopt a nonconvex convolutional neural network (CNN) model as a classifier and set the number of categories to 3, resulting in a minimax problem of the form:

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \sum_{i=1}^{3} y_i \ell_i(x) - g(y), \tag{15}$$

where $x \in \mathbb{R}^p$ represents the parameters of the CNN, and ℓ_1, ℓ_2, ℓ_3 correspond to the loss of three categories, $g(y) = \frac{\eta}{2}\|y\|_2^2$ is a regularizer with $\eta > 0$. We train (15) on the datasets to classify: $i)$ gray-scale hand-written digits $\{0, 2, 3\}$ from MNIST; $ii)$ fashion images with target classes {T-shirt/top, Sandal, Ankle boot} from F-MNIST; $iii)$ RBG colored images with target classes {Plane, Truck, Deer} from CIFAR10. For both MNIST and F-MNIST $p = 43831$, $n = 18000$ and $d = 28 \times 28 \times 1$, and for CIFAR10 $p = 61411$, $n = 15000$, and $d = 32 \times 32 \times 3$.

We let the regularization parameter $\eta = 0.1$ as suggested in [16]. We compare SAPD-VR against the other VR-based algorithms SREDA and SMDA-VR over 30 runs.We tune the primal stepsizes of SAPD-VR and SREDA by a grid search over the set $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}\}$ and the ratio of primal to dual stepsizes, i.e., τ/σ, is chosen from $\{10, 10^2, 5 \times 10^2, 10^3\}$. For SMDA-VR, the primal and dual stepsizes are 10^{-3} and 10^{-5} as suggested in [16] –we also tried stepsizes bigger than the suggested; but, it caused convergence issues in the experiments. We set the large batchsize $|\mathcal{B}| = 3000$ and the small batchsize $|\mathcal{I}| = 200$ for all algorithms and data sets; the frequency $q = 200$ is used for SAPD-VR and SMDA-VR, and we tune q for SREDA taking values from $\{10, 50, 100, 200\}$. The momentum θ for SAPD-VR is tuned taking values from $\{0.8, 0.85, 0.9\}$ and inner iteration number $N = \frac{\ln(265)}{\ln(\frac{1}{\theta})}$ is set according to eq. (7). For SREDA, we tune the inner loop iteration from $\{10, 50, 100\}$. Fig. 2 shows that SAPD-VR outperforms the other VR-based algorithms clearly in terms of both the average loss and the standard deviation of the loss.

6 Conclusion

In this paper, we considered both WCSC and WCMC saddle-point problems assuming we only have an access to an unbiased stochastic first-oracle with a finite variance. This setting arises in many applications ranging from distributionally robust learning to GANs. We proposed a new method `SAPD+`, which achieves an improved complexity in terms of target accuracy ϵ for both WCSC and WCMC problems; moreover, our bound for `SAPD+` has a better dependency to the condition number κ_y for the WCSC scenario. We also showed that our algorithm `SAPD+` can also support the SPIDER variance-reduction technique. Finally, we provided numerical experiments demonstrating that `SAPD+` can achieve a state-of-the-art performance on distributionally robust learning and on multi-class classification problems arising in ML.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Anestis Antoniadis, Irène Gijbels, and Mila Nikolova. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63(3):585–615, 2011.

[3] Amir Beck. *First-order methods in optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

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

[5] Bugra Can, Mert Gurbuzbalaban, and Necdet Serhat Aybat. A Variance-Reduced Stochastic Accelerated Primal Dual Algorithm. *arXiv e-prints*, page arXiv:2202.09688, February 2022.

[6] Ziyi Chen, Shaocong Ma, and Yi Zhou. Accelerated proximal alternating gradient-descent-ascent for nonconvex minimax machine learning. *arXiv preprint arXiv:2112.11663*, 2021.

[7] Ziyi Chen, Yi Zhou, Tengyu Xu, and Yingbin Liang. Proximal gradient descent-ascent: Variable convergence under KL geometry. *arXiv preprint arXiv:2102.04653*, 2021.

[8] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[9] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.

[10] Alireza Fallah, Asuman Ozdaglar, and Sarath Pattathil. An optimal multistage stochastic gradient method for minimax problems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3573–3579. IEEE, 2020.

[11] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[13] Mert Gürbüzbalaban, Andrzej Ruszczyński, and Landi Zhu. A stochastic subgradient method for distributionally robust non-convex learning. *arXiv preprint arXiv:2006.04873*, 2020.

[14] Erfan Yazdandoost Hamedani and Necdet Serhat Aybat. A primal-dual algorithm for general convex-concave saddle point problems. *arXiv preprint arXiv:1803.01401*, 2018.

[15] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *Journal of Machine Learning Research*, 23(36):1–70, 2022.

[16] Feihu Huang, Xidong Wu, and Heng Huang. Efficient mirror descent ascent methods for nonsmooth minimax problems. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Alfredo N Iusem, Teemu Pennanen, and Benar Fux Svaiter. Inexact variants of the proximal point algorithm without monotonicity. *SIAM Journal on Optimization*, 13(4):1080–1097, 2003.

[18] Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, pages 4880–4889. PMLR, 2020.

[19] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pages 1895–1904. PMLR, 2017.

[20] Haochuan Li, Yi Tian, Jingzhao Zhang, and Ali Jadbabaie. Complexity lower bounds for nonconvex-strongly-concave min-max optimization. *arXiv preprint arXiv:2104.08708*, 2021.

[21] Zichong Li and Yangyang Xu. Augmented lagrangian–based first-order methods for convex-constrained programs with weakly convex objective. *INFORMS Journal on Optimization*, 3(4):373–397, 2021.

[22] Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

[23] Tianyi Lin, Chi Jin, and Michael. I. Jordan. Near-Optimal Algorithms for Minimax Optimization. *arXiv e-prints*, page arXiv:2002.02417, February 2020.

[24] Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020.

[25] Songtao Lu, Ioannis Tsaknakis, Mingyi Hong, and Yongxin Chen. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*, 68:3676–3691, 2020.

[26] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33:20566–20577, 2020.

[27] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[28] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in NeurIPS*, pages 2208–2216, 2016.

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

[30] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

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

[32] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017.

[33] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*, 2019.

[34] Dmitrii M Ostrovskii, Andrew Lowy, and Meisam Razaviyayn. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems. *SIAM Journal on Optimization*, 31(4):2508–2538, 2021.

[35] H Rafique, M Liu, Q Lin, and T Yang. Non-convex min–max optimization: provable algorithms and applications in machine learning (2018). *arXiv preprint arXiv:1810.02060*, 1810.

[36] Kiran Koshy Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Efficient algorithms for smooth minimax optimization. *arXiv preprint arXiv:1907.01543*, 2019.

[37] Tengyu Xu, Zhe Wang, Yingbin Liang, and H Vincent Poor. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. 2020.

[38] Zi Xu, Huiling Zhang, Yang Xu, and Guanghui Lan. A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. *arXiv preprint arXiv:2006.02032*, 2020.

[39] Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Optimal epoch stochastic gradient descent ascent methods for min-max optimization. *Advances in Neural Information Processing Systems*, 33:5789–5800, 2020.

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

[41] Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. In *Uncertainty in Artificial Intelligence*, pages 482–492. PMLR, 2021.

[42] Xuan Zhang, Necdet Serhat Aybat, and Mert Gürbüzbalaban. Robust accelerated primal-dual methods for computing saddle points. *arXiv preprint arXiv:2111.12743*, 2021.

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? [Yes] Although we improve the existing best known bounds, there is still room to improve κ_y dependence based on the existing lower complexity bounds [20, 41], unless the lower complexity bounds are loose.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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)? [Yes]
   (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)? [Yes]
   (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]