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

Xuan Zhang
Department of Industrial and Manufacturing Engineering
Pennsylvania State University
University Park, PA, USA.
xxz358@psu.edu

Necdet Serhat Aybat
Department of Industrial and Manufacturing Engineering
Pennsylvania State University
University Park, PA, USA.
nsa10@psu.edu

Mert Gürbüzbalaban
Department of Management Science and Information Systems
Rutgers University
Piscataway, NJ, USA
mg1366@rutgers.edu

Abstract

We propose a new stochastic method SAPD+ for solving nonconvex-concave minimax problems of the form $\min_{x} \max_{y} \mathcal{L}(x, y) = f(x) + \Phi(x, y) - g(y)$, 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 $O(L\kappa_y^2 \epsilon^{-3})$ and $O(L^3 \epsilon^{-6})$, respectively, without assuming compactness of the problem domain, where $\kappa_y$ 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 $O(L\kappa_y^2 \epsilon^{-3})$ for weakly convex-strongly concave setting. We demonstrate the efficiency of SAPD+ on a distributionally robust learning problem with a nonconvex regularizer and also on a multi-class classification problem in deep learning.

1 Introduction

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

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

where $\mathcal{X}$ and $\mathcal{Y}$ are, $n$ and $m$ dimensional Euclidean spaces, the function $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is smooth and possibly nonconvex in $x \in \mathcal{X}$ and $\mu_y$-strongly concave in $y \in \mathcal{Y}$ for some $\mu_y \geq 0$—with the convention that for $\mu_y = 0$, $\Phi$ 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 $y \in \text{dom} \ g \subset \mathcal{Y}$. Weakly convex functions constitute a rich class of non-convex functions and arise naturally in many practical settings for machine learning (ML) applications [9][15]. Precise definitions will be given later in Section 2. In practice, WC assumption is widely satisfied, e.g., under smoothness—in remark [1] 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 settings including constrained optimization of WC objectives based on Lagrangian duality \([22]\).
Generative Adversarial Networks (GAN) (where \(z\) denotes the parameters of the generator network
whereas \(y\) represents the parameters of the discriminator network \([13]\)), distributional robust learning
with weakly convex loss functions such as those arising in deep learning \([14, 35]\) and learning
problems with non-decomposable losses \([35]\).

There are two important settings for \((1)\): (i) the deterministic setting, where the partial gradients of \(\Phi\)
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
\([6, 17, 12, 23, 24, 25, 35, 36, 38]\); our focus in this paper will be mainly on the stochastic setting,
which is more relevant and more applicable to ML problems. Indeed, due to large-dimensions and
the sheer 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 \(\phi(\cdot) \equiv \max_{y \in Y} \mathcal{L}(:, y)\)
and measure the violation of first-order necessary conditions for the non-convex problem \(\min_{x \in X} \phi(x)\).
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 \(K_\epsilon\) such that \(E[\|\nabla \phi(x_k)\|] \leq \epsilon \) for all \(k \geq K_\epsilon\), where \(\phi_\lambda\) denotes the Moreau
envelope of the primal function \(\phi\) – see Definitions \([3, 4]\) and \([5]\). Another commonly used natural metric
is the gradient norm of the primal function \(\phi(\cdot) \equiv \max_{y \in Y} \mathcal{L}(:, y)\), abbreviated as GNP, where the
aim is to derive \(K_\epsilon\) such that \(E[\|\nabla \phi(x_k)\|] \leq \epsilon \) for all \(k \geq K_\epsilon\). Other metrics such as the notion of
\(\epsilon\)-first-order Nash equilibrium (FNE) and its generalized versions also exist in the literature \([32, 33]\).

When using any of the aforementioned 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 \(k \geq 0\). For the
WCSC setting, it is crucial to note that GNME, GNP and FNE metrics are all equivalent in the sense that
convergence in either of them implies convergence in the other two metric for WCSC problems \([23]\).
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 computational cost required for the GNME guarantee, and the overall worst-case
complexity (in terms of worst-case dependency to the target accuracy \(\epsilon\)) 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 \(\phi\) is not necessarily differentiable
for this scenario. Moreover, our work accounts for the individual effects of \(L_{xx}, L_{xy}, L_{gx}\) and \(L_{gy}\),
respectively, instead of using the worst-case parameters \(L \equiv \max\{L_{xx}, L_{xy}, L_{gx}, L_{gy}\}\), while the
majority of related work ignore the influence of these block Lipschitz constants in their analyses. 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 \([43]\).

Contributions. Table \(1\) summarizes the relevant existing work for WCSC and WCMC problems
closest to our setting. 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 \(b'\) all iterations except for few,
where they need a large batch-size \(b \geq b'\) once in every \(q\) iterations. The period \(q\) is equal to
the number of times small batches are sampled consecutively plus one, and it 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.
VR-based methods to state small batch
To make the comparison of our results with the existing work easier, we provide the results in the Table 1:

<table>
<thead>
<tr>
<th>Reference</th>
<th>Complexity</th>
<th>Compactness</th>
<th>VR-based</th>
<th>Batchsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Ratcliffe et al. [35]</td>
<td>$O(e^{-t} \log(e^{-t}))$</td>
<td>(n, n)</td>
<td>x</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>† Yan et al. [39]</td>
<td>$O(e^{-t} \log(e^{-t}))$</td>
<td>(y, y)</td>
<td>x</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>† Yang et al. [41]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(n, n)</td>
<td>x</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Lin et al. [23]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(y, y)</td>
<td>x</td>
<td>$O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>Bot and Bohm [4]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(n, n)</td>
<td>x</td>
<td>$O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>† Huang et al. [17]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(n, n)</td>
<td>✓</td>
<td>$O(\kappa_y^{-3}), O(\kappa_y^{-3}), O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>† Huang et al. [16]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(y, y)</td>
<td>✓</td>
<td>$O(\kappa_y^{-3}), O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>Luo et al. [26]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(y, y)</td>
<td>✓</td>
<td>$O(\kappa_y^{-3}), O(\kappa_y^{-3}), O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>· Xu et al. [31]</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(y, y)</td>
<td>✓</td>
<td>$O(\kappa_y^{-3}), O(\kappa_y^{-3}), O(\kappa_y^{-3})$</td>
</tr>
<tr>
<td>· SAPD+ Theorem 7</td>
<td>$O(L \kappa_y^{-3})$</td>
<td>(n, n)</td>
<td>x</td>
<td>$O(\kappa_y^{-3}), O(\kappa_y^{-3}), O(\kappa_y^{-3})$</td>
</tr>
</tbody>
</table>

Table 1: Summary of relevant work for WCSC and WCMC problems. For the column “Compactness”, we use y and n to indicate when the results require compactness and when do not require it, 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 to state small batch ($b$), large batch ($b$), and frequency ($q$) employed within the algorithm.

Table notes: * For WCSC setting, [35] assumes $\Phi(\cdot, y) \triangleq c^1(\cdot)y$ is weakly convex and $g(\cdot)$ is strongly convex. † In [35], $L = \Phi$ and $L = \Phi$ need not be smooth, rather second moment of stochastic subgradients is assumed to be uniformly bounded. When $\Phi$ is $L$-smooth, $\Phi(\cdot, y)$ and $\Phi(x, \cdot)$ are $L_0$-Lipschitz, the results in [35] imply $O(L_0^2 \kappa_y^{-3} \log^3(\sqrt{\kappa_y} L_0/\epsilon))$ complexity. ‡ The complexity results reported here are different than those in [17], [16]. The issues in their proofs leading to the wrong complexity results are explained in Appendix I. The notation $O$ ignores logarithmic factors.

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 $\epsilon$, $\kappa_y$, 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 subproblems using an accelerated primal-dual method, SAPD [43]. In Theorem 3, we establish an oracle complexity of $O(L \kappa_y \epsilon^{-4})$ 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 $\kappa_y$ dependence in the literature; indeed, prior to this work, without using variance reduction, the best known complexity was $O(L \kappa_y^2 \epsilon^{-4})$ shown in [41]; hence, we establish a $O(\kappa_y)$ improvement.

- We propose a variance-reduced version of SAPD+ in Theorem 4. For WCSC setting, SAPD+ using variance reduction achieves an oracle complexity of $O(L \kappa_y^2 \epsilon^{-3})$ –this bound has the best $\epsilon$-dependency in the literature to our knowledge, and among all the methods with the $O(\epsilon^{-3})$ complexity, our approach has the best condition number, $\kappa_y$, dependence; indeed, prior to this work, the best known complexity was $O(L \kappa_y^2 \epsilon^{-4})$; hence, we establish $O(\kappa_y)$ factor improvement.

- For the WCMC case, our proposed algorithm SAPD+ results in $O(L^3 \epsilon^{-6})$ 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 and also on a worst-case multi-class classification problem in deep learning.

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

We start with describing the notion of weak convexity.

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

**Definition 2.** A differentiable function \( h : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\} \) is \( L \)-smooth if \( \exists L > 0 \) such that for all \( x, x' \in \text{dom } h \), \( \| \nabla h(x) - \nabla h(x') \| \leq L \| x - x' \| \).

**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 \( (i) \). Next, we introduce our assumptions.

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

**Assumption 2.** There exist \( L_{xx}, L_{xy} \geq 0, L_{yx}, L_{yy} > 0 \) such that \( \| \nabla_x \Phi(x, y) - \nabla_x \Phi(x', y) \| \leq L_{xx} \| x - x' \| + L_{xy} \| y - y' \| \), and \( \| \nabla_y \Phi(x, y) - \nabla_y \Phi(x', y') \| \leq L_{yx} \| x - x' \| + L_{yy} \| y - y' \| \) for all \( x, x' \in \mathcal{X} \), and \( y, y' \in \text{dom } g \subset \mathcal{Y} \).

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 \( (i) \) 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

\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \mathcal{L}(x, y) + \frac{\mu_x + \gamma}{2} \| x - x_0 \|^2
\]

for some given \( \mu_x > 0 \) and \( x_0 \in \mathcal{X} \)–strong convexity follows from \( \mathcal{L}(\cdot, y) \) being \( \gamma \)-weakly convex.

We make the following assumption on the statistical nature of the gradient noise as in, e.g., [8][11][43].

**Assumption 3.** Given arbitrary \( x_0 \in \mathcal{X} \) and \( \mu_x > 0 \), let \( \{x_k, y_k\} \) sequence be generated by SAPD, stated in Algorithm 1 running on \( (i) \). There exist \( \delta_x, \delta_y \geq 0 \) such that for all \( k \geq 0 \), the stochastic gradients \( \tilde{\nabla}_x \Phi(x_k, y_{k+1}; \omega^x_k) \), \( \tilde{\nabla}_y \Phi(x_k, y_k; \omega^y_k) \) and random sequences \( \{\omega^x_k\}, \{\omega^y_k\}_k \) satisfy the conditions:

1. \( E[\nabla_x \Phi(x_k, y_{k+1}; \omega^x_k)|x_k, y_{k+1}] = \nabla_x \Phi(x_k, y_{k+1}) \);
2. \( E[\nabla_y \Phi(x_k, y_k; \omega^y_k)|x_k, y_k] = \nabla_y \Phi(x_k, y_k) \);
3. \( \| \nabla_x \Phi(x_k, y_{k+1}; \omega^x_k) - \nabla_x \Phi(x_k, y_{k+1}) \|^2 \leq \delta^2_x \);
4. \( \| \nabla_y \Phi(x_k, y_k; \omega^y_k) - \nabla_y \Phi(x_k, y_k) \|^2 \leq \delta^2_y \).

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., [6][11][43], and are satisfied when gradients are estimated from randomly sampled data points with replacement.

For WCSC minimax problems, a commonly adopted definition for \( \epsilon \)-stationary is based on Moreau envelope, e.g., see [23][39]. It is inspired by Davis and Drusvyatskiy’s work [9] for solving weakly convex minimization problems. For the sake of completeness, we briefly review this idea below.

**Definition 3.** Let \( \phi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\} \) be \( \gamma \)-weakly convex. Then, for any \( \lambda \in (0, \gamma^{-1}) \), Moreau envelope of \( \phi \) is defined as \( \phi_\lambda : \mathbb{R}^d \rightarrow \mathbb{R} \) such that \( \phi_\lambda(x) = \max_{y \in \mathcal{Y}} \phi(y) + \frac{1}{2\lambda} \| y - x \|^2 \).
Lemma 1. Let \( \phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\} \) be a \( \gamma \)-weakly convex function. For any given \( \lambda \in (0, \gamma^{-1}) \), \( \phi_\lambda(\cdot) \) is well-defined on \( \mathcal{X} \). Moreover, \( \nabla \phi_\lambda(x) = \lambda^{-1}(x - \text{prox}_{\lambda \phi}(x)) \) for \( x \in \mathcal{X} \); hence, \( \phi_\lambda \) is \( \lambda^{-1} \)-smooth, where \( \text{prox}_{\lambda \phi}(x) \triangleq \arg\min_{w \in \mathcal{X}} \{ \phi(w) + \frac{1}{2\lambda} \| w - x \|^2 \} \).

Definition 4. Under Assumption \( \mathbf{2} \) let \( \phi, \phi^* : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\} \) such that \( \phi(x) \triangleq \max_{y \in \mathcal{Y}} \mathcal{L}(x, y) \) and \( \phi^*(x) = \phi(x) - f(x) \) for \( x \in \text{dom} \ f \), i.e., \( \phi^*(x) \triangleq \max_{y \in \mathcal{Y}} \Phi(x, y) - g(y) \) for \( x \in \text{dom} \ f \).

Remark 2. Under Assumption \( \mathbf{2} \) since \( \Phi(\cdot, y) \) is \( \gamma \)-weakly convex for any \( y \in \text{dom} \ g \), \( \phi^* \) is \( \gamma \)-weakly convex; hence, \( \phi \) is also \( \gamma \)-weakly convex. Note that
\[
\text{prox}_{\lambda \phi}(x) = \arg\min_{w \in \mathcal{X}} \{ \phi(w) + \frac{1}{2\lambda} \| w - x \|^2 \} = \arg\max_{y \in \mathcal{Y}} \mathcal{L}(w, y) + \frac{1}{2\lambda} \| w - x \|^2. \tag{3}
\]
Furthermore, when \( \mu_y > 0 \), \( \phi^* \) is differentiable on \( \text{dom} \ f \).

In the following definition, we introduce the notion of \( \epsilon \)-stationary with respect to the GNME metric.

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

Thus, from Lemma 1 computing an \( \epsilon \)-stationary point \( x_\epsilon \) for \( \phi \) is equivalent to searching for \( x_\epsilon \) such that \( \| x_\epsilon - \text{prox}_{\lambda \phi}(x_\epsilon) \| \) is small. Recall that for any \( \lambda \in (0, \gamma^{-1}) \), \( \text{prox}_{\lambda \phi}(x) \) is well-defined and unique. We also observe from (3) that \( \text{prox}_{\lambda \phi}(\cdot) \) computation is indeed an SCSC problem. To compute \( x_\epsilon \) such that \( \| x_\epsilon - \text{prox}_{\lambda \phi}(x_\epsilon) \| \) is small, it is natural to consider the iPPM algorithm -- e.g., see [13]. A generic iPPM generates \( \{x_{\epsilon t}^i\}_{t \geq 0} \) such that \( x_{\epsilon t}^i \approx \text{prox}_{\lambda \phi}(x_{\epsilon t}^{i-1}) \), i.e., proximal steps are “inexactly” computed for \( t \geq 0 \), starting from an arbitrary given point \( x_{\epsilon 0}^0 \in \mathcal{X} \).

In the next section, we describe the proposed SAPD+ method, an iPPM algorithm employing SAPD to inexactly solve the SCSC subproblems arising in the iPPM iterations.

3 The proposed algorithm SAPD+ and its analysis

The convergence and robustness properties of SAPD for SCSC SP problems are analyzed in [43]. For the WCSC SP problems, as we explained in the previous section, 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 outer iterations is denoted with \( t \in \mathbb{Z}_+ \). At each outer iteration \( t \geq 1 \), we inexactly compute the prox map, i.e., \( x_{\epsilon t}^{i+1} \approx \text{prox}_{\lambda \phi}(x_{\epsilon t}^i) \), which is well-defined for \( \lambda \in (0, \gamma^{-1}) \); hence, to derive our preliminary results, we fix \( \lambda = (\mu_\nu + \gamma)^{-1} \) for some given \( \mu_\nu > 0 \) -- thus, \( \mathcal{L}(x, y) + \frac{4\mu_\nu + \gamma}{\lambda} \| x - x_{\epsilon t}^i \|^2 \) is SCSC in \( (x, y) \) with moduli \( \mu_x, \mu_y \) and has a unique saddle point. Consider the following SCSC SP problem:
\[
\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \mathcal{L}^i(x, y) \triangleq f(x) + \Phi^i(x, y) - g(y), \quad \text{where} \quad \Phi^i(x, y) \triangleq \Phi(x, y) + \frac{\mu_x + \gamma}{2} \| x - x_{\epsilon t}^i \|^2. \tag{4}
\]

We will construct \( \{x_{\epsilon t}^i\}_{t=1}^T \subset \text{dom} \ f \) by inexactly solving (4) at each outer iteration \( t \in \mathbb{Z}_+ \) through running SAPD for \( N_t \in \mathbb{Z}_+ \) iterations --we will specify \( N_t \in \mathbb{Z}_+ \) later. Next, we briefly explain the main step of SAPD+ with VR-flag=false. The statement in line 4 of Algorithm 2 means that \( (x_{\epsilon t}^{i+1}, y_{\epsilon t}^{i+1}) \) is generated using SAPD, where is displayed in Algorithm 1; indeed, SAPD is run on \( \{x_{\epsilon t}^i\} \) for \( N_t \) iterations with SAPD parameters \( \{\tau, \sigma, \theta, \mu_x\} \) and starting from the initial point \( (x_{\epsilon 0}^0, y_{\epsilon 0}^0) \). To analyze the convergence of SAPD+, we first define the gap function \( G^i \) for \( t \)-th SAPD+ iteration:
\[
G^i(x, y) \triangleq \max_{y' \in \mathcal{Y}} \mathcal{L}^i(x, y') - \min_{x' \in \mathcal{X}} \mathcal{L}^i(x', y). \tag{5}
\]

Recall that \( \mathcal{L}^i \) is an SCSC function; therefore, \( i \) it has a unique saddle point denoted by \( (x_{\epsilon t}^i, y_{\epsilon t}^i) \), and it is important to note that \( x_{\epsilon t}^i = \text{prox}_{\lambda \phi}(x_{\epsilon t}^i) \) for \( \phi(x) = \max_{y \in \mathcal{Y}} \mathcal{L}(x, y) \) and \( \lambda = (\gamma + \mu_x)^{-1} \); \( ii \)

One can argue that \( \phi^*(\cdot) + \frac{\mu_x + \gamma}{2} \| \cdot \|^2 \) is a pointwise supremum of convex functions.
for any $(x, y) \in \text{dom } f \times \text{dom } g$, the following quantities are well-defined:

$$x^*_t(y) \triangleq \arg\min_{x' \in X} L'(x', y), \quad y^*_t(x) \triangleq \arg\max_{y' \in Y} L'(x, y') = \arg\max_{y' \in Y} L(x, y'). \quad (6)$$

Thus, it follows that $G^t(x, y) = L'(x, y_t(x)) - L'(x^*_t(y), y)$. Moreover, for $(x, y) \in \text{dom } f \times \text{dom } g$, we also define $G(x,y) \triangleq \sup_{y' \in Y} L(x, y'') - \inf_{y' \in Y} L(x', y')$. Assumption 1 ensures that $G$ is well defined.

Next, we first provide our oracle complexity in the GNE metric under the compactness assumption of the primal-dual domains; later, in section 3.1, we show that under a particular subdifferentiability assumption compactness requirement can be avoided.

**Assumption 4.** $f$ and $g$ are compact sets.

**Theorem 1.** Suppose Assumptions 1, 2, 3 hold. Let $\mu_\gamma = 2, \tau = 1, \sigma$ and $N$ be chosen as

$$N = 33 \max \left\{ \frac{1}{\tau \eta_x}, \frac{1}{\tau \mu_y}, \frac{1}{\tau \mu_y \delta}, \frac{1}{\tau \mu_x \gamma}, \frac{1}{\tau \mu_x \delta}, \frac{1}{\tau \mu_x \gamma} \right\}, \quad \tau = \min \left\{ \frac{1}{\tau \mu_x}, \frac{1}{1 + \tau \mu_y \delta}, \frac{1}{\tau \mu_y \gamma} \right\}, \quad \sigma = \min \left\{ \frac{1}{\tau \mu_x + \tau \mu_y \delta}, \frac{1}{\tau \mu_y + \tau \mu_x \gamma} \right\}. \quad (7)$$

Then, for any $\epsilon > 0$, when $\text{VR-flag} = \text{false}$, SAPD+ guarantees $\epsilon$-stationary, $\min_{t=0,\ldots,T} \mathbb{E} \left[ \| \nabla \phi_\lambda (x^*_t) \| \right] \leq \epsilon$, for $T \geq 96 \mathcal{G}(x_0^*, y_0^*) \cdot \frac{1}{\gamma^2}$, which requires $C_\epsilon$ stochastic first-order oracle calls in total where

$$C_\epsilon = \mathcal{O} \left( \left( \frac{\max \{ L_{xx}, L_{yx}, L_{yx} \} + \max \{ \beta L_{xx}, L_{yx} \} }{\gamma} \right) \cdot \epsilon^{-2} + \left( \frac{\beta^2}{\gamma} + \frac{\gamma^2}{\mu_y} \right) \cdot \epsilon^{-4} \right) \mathcal{G}(x_0^*, y_0^*).$$

**Proof.** See appendix B for the proof.

**Remark 3.** Since $\mathbb{E} \left[ \min_{t=0,\ldots,T} \| \nabla \phi_\lambda (x^*_t) \| \right] \leq \min_{t=0,\ldots,T} \mathbb{E} \left[ \| \nabla \phi_\lambda (x^*_t) \| \right]$, the guarantees given in Theorem 1 also hold for achieving $\mathbb{E} \left[ \min_{t=0,\ldots,T} \| \nabla \phi_\lambda (x^*_t) \| \right] \leq \epsilon$.

**Remark 4.** For any $y \in \text{dom } g$, 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

$$L \triangleq \max \{ L_{xx}, L_{yx}, L_{yx}, L_{yy} \}, \quad \kappa_y \triangleq \frac{L}{\mu_y}, \quad \delta \triangleq \max \{ \delta_x, \delta_y \}, \quad \gamma = \frac{L}{\mu_y}. \quad (8)$$

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

In fact, Li et al. [21] (see also [42]) provide a lower complexity bound for a class of first-order stochastic algorithms that do not use variance reduction. The lower bound for finding $\epsilon$-stationary points of smooth WSCC problems in GNP metric is $\Omega (L_\Delta \phi (\sqrt{\epsilon^2 + \gamma^2} \epsilon^{-4}))$, where $\Delta_\phi \triangleq \phi(x_0) - \min_{x \in X} \phi(x)$ and $x_0$ is an arbitrary initial point.

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

$$G_\lambda (\hat{x}) \triangleq \frac{1}{\lambda} [\hat{x} - \text{prox}_{\lambda f}(\hat{x} - \lambda \nabla \phi^*(\hat{x}))] \quad (9)$$

called the generalized gradient mapping and its norm is frequently used in optimization for assessing stationarity (see e.g. [10]). Theorem 1 provides guarantees in the GNE metric. Theorem 2 shows that given $x_e$, an $\epsilon$-stationary point in GNE metric (see definition 5) in expectation, we can generate $\hat{x}$ such that $\mathbb{E} \| G_\lambda (\hat{x}) \| \leq \epsilon$ for some $\lambda > 0$, i.e., an $\epsilon$-stationary point in generalized gradient mapping metric, within $\mathcal{O}(1/\epsilon^2)$ SAPD iterations. Indeed, when $f(\cdot) = 0$, this metric and the GNP metric are the same.

**Theorem 2.** Suppose Assumptions 1, 2, 3 hold, and $x_e$, an $\epsilon$-stationary point for the $\gamma$-weakly convex function $\phi(\cdot) = \max_{y \in Y} \mathcal{L}(\cdot, y)$ in expectation, i.e., $\mathbb{E} \| \nabla \phi_\lambda (x_e) \| \leq \frac{\gamma}{\epsilon}$ for some fixed $\lambda \in (0, \gamma^{-1})$ is given. Then, there exists some $\tau, \sigma, \theta \geq 1$ such that initialized from $x_e$, SAPD, stated in Algorithm 1 can generate $\hat{x}$ such that $\mathbb{E} \| G_\lambda (\hat{x}) \| \leq \epsilon$ within $\mathcal{O}(1/\epsilon^2)$ stochastic first-order oracle calls, where $\phi^*(\cdot) = \max_{y \in \mathcal{Y}} \phi(\cdot, y) - g(\cdot)$ so that $\phi = f + \phi^*$ as in Definition 4.

**Proof.** See appendix B for the proof.

**Remark 5.** Based on Remark 3, the random vector $x_e$ in Theorem 2 can be chosen as $x_e^* \triangleq \arg\min_{0 \leq t \leq \mathcal{T}} \| \nabla \phi_\lambda (x_e) \|$. However, since $t_e$ can not be computed in practice, we provide an alternative method in the appendix to generate a point $x_e$ such that $\mathbb{E} \| \nabla \phi_\lambda (x_e) \| \leq \epsilon$ within $\mathcal{O} \left( \frac{L_{xx} \mathcal{G}(x_0^*, y_0^*)}{\epsilon^2} + \frac{L_{xx} \mathcal{G}(x_0^*, y_0^*)}{\epsilon^2} \right)$ stochastic first-order oracle calls – see Theorem 1 in appendix D.
3.1 Relaxing the compactness assumption

In Theorem 1 we assume that dom f and dom g are compact sets, e.g., f(·) = 1χ(·) and g(·) = 1Y(·), where X ⊂ X and Y ⊂ 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 ∃Bf, Bg > 0 such that ∩f(x) ≤ Bf for all x ∈ dom f and ∩f(y) ≤ Bg for all y ∈ dom g.

Remark 6. Assumption 5 holds when f is an indicator function of a closed convex set (not necessarily bounded) or for f : R^d → R ∪ {+∞} such that dom f is open and f is Lipschitz. Two important examples for this scenario are: (i) f(·) = 0, (ii) f is a norm, e.g., κ_1, κ_2, or the Nuclear norms.

The existing work based on iPPM framework either require compactness, e.g., [39], or some special structure on L, e.g., [33]. This is also true for VR-based methods, e.g., [16, 26, 37]. To our knowledge, ours is the first one to overcome this difficulty and strictly improve the best known complexity bound for the WSCC setting without compactness assumption; moreover, the same idea also works simultaneously with a variance reduction technique that will be discussed later (see section 4). Finally, the same trick for removing compactness assumption for the WSCC setting also helps removing the compactness assumption for the primal domain in WCMC setting and we still improve the best known complexity for this setting as well (see section 4).

Remark 7. In [23], 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], compactness of the domain is needed for their proof to hold. In [17], the sub-level set \{x : φ(x) + f(x) ≤ α\} is required to be compact for all α > 0. There are simple convex functions that do not satisfy this condition such as f(x) = max\{0, x\}. Bot and Bohm [4] use milder assumptions than [23] without requiring compactness; however, their complexity is the same as the complexity of [23].

Theorem 3. The result of Theorem 1 continues to hold, if one replaces the compact domain assumption, i.e., Assumption 2 with Assumption 5.

Proof. See appendix E for the proof.

4 Variance reduction

Variance reduction techniques have been found useful for solving SCSC problems in finite sum form, e.g., [34]—see also [5] using Richardson-Romberg extrapolation in solving SCSC problems with noisy gradients to obtain improved practical performance.

In this section, we equip SAPD+ with SPIDER variance reduction technique [12], a variant of SARAH [31] More precisely, for exactly solving SCSC subproblems given in [4], we propose using VR-SAPD as stated in Algorithm 3. Note VR-SAPD employs a large batchsize of b in every q iterations and use small batchsizes of b_ε and b_ε' for the rest. We prove that SAPD+ using variance reduction, i.e., with VR-flag=true, achieves an oracle complexity of \(O(L\kappa_k^2\epsilon^{-3})\); hence, we show an \(O(\kappa_k)\) factor improvement over the best known complexity in the literature to our knowledge.

Here, we use \(\nabla_y\Phi_k^{\eta}(x_k, y_k)\) to represent \(\frac{1}{|B_k^\eta|} \sum_{y \in B_k^\eta} \nabla_y\Phi(x_k, y_k; \eta_k^{\eta})\), where \(B_k^\eta = \{\eta_k^{\eta}\}_{\eta_k^{\eta}=1}^b\) is the mini-batch with \(|B_k^\eta| = b\) and we define \(\nabla_y\Phi_k^{\eta}(x_k, y_{k+1})\) similarly. In addition, \(I_k^\eta = \{\eta_k^{\eta}\}\) and

<table>
<thead>
<tr>
<th>Algorithm 3 VR-SAPD Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Input: τ, σ, β, ζ, x_0, y_0, N, b, b_ε', b_ε, g</td>
</tr>
<tr>
<td>2: (\Phi(x, y) = \Phi(x, y) + \frac{2\epsilon^2}{x-x_0}</td>
</tr>
<tr>
<td>3: Let (B_k^\eta, B_k^{\eta+1}) be random mini-batch samples with (</td>
</tr>
<tr>
<td>4: (w_0 = \nabla_y\Phi_k^0(x_0, y_0)), (v_0 = w_0)</td>
</tr>
<tr>
<td>5: for (k \geq 0) do</td>
</tr>
<tr>
<td>6: (y_{k+1} = \text{prox}_{\sigma g}(y_k + \sigma s_k))</td>
</tr>
<tr>
<td>7: if mod(k, q) == 0 then</td>
</tr>
<tr>
<td>8: (v_k = \nabla_x\Phi_k(x_k, y_{k+1}))</td>
</tr>
<tr>
<td>9: else</td>
</tr>
<tr>
<td>10: Let (T_k^\eta) be random mini-batch sample with (</td>
</tr>
<tr>
<td>11: (v_k = \nabla_x\Phi_k(x_k, y_{k+1}) - \nabla_x\Phi_k(x_{k-1}, y_k) + v_{k-1})</td>
</tr>
<tr>
<td>12: end if</td>
</tr>
<tr>
<td>13: (x_{k+1} = \text{prox}_{\sigma f}(x_k - \tau v_k))</td>
</tr>
<tr>
<td>14: Let (B_k^{\eta+1}, B_k^{\eta+1}) be random mini-batch samples with (</td>
</tr>
<tr>
<td>15: if mod(k + 1, q) == 0 then</td>
</tr>
<tr>
<td>16: (w_{k+1} = \nabla_y\Phi_k^{\eta}(x_{k+1}, y_{k+1}))</td>
</tr>
<tr>
<td>17: else</td>
</tr>
<tr>
<td>18: Let (T_k^{\eta+1}) be mini-batch sample with (</td>
</tr>
<tr>
<td>19: (q_{k+1} = \nabla_y\Phi_k^{\eta}(x_{k+1}, y_{k+1}) - \nabla_y\Phi_k^{\eta}(x_k, y_k))</td>
</tr>
<tr>
<td>20: (w_{k+1} = w_k + q_{k+1})</td>
</tr>
<tr>
<td>21: end if</td>
</tr>
<tr>
<td>22: (s_{k+1} = (1 + \theta)w_{k+1} - \theta w_k)</td>
</tr>
<tr>
<td>23: end for</td>
</tr>
<tr>
<td>24: Output: ((\bar{x}<em>N, \bar{y}<em>N) = \frac{1}{N} \sum</em>{k=0}^{N-1} (x</em>{k+1}; y_{k+1}))</td>
</tr>
</tbody>
</table>
\( T_k^b = \{ \omega_k^{b_i} \} \) with \( |T_k^b| = b'_i \) and \( |T_k^g| = b'_g \) denote the small mini-batches for generating \( \nabla_y \Phi^1(x_k, y_k) \) and \( \nabla_y \Phi^g(x_{k+1}) \). When we run VR-\textsc{SAPD} on a generic subproblem as in (2), we use the convention that \( \nabla_y \Phi^1(x_k, y_{k+1}) \triangleq \nabla_y \Phi^1(x_k, y_{k+1}) + (\mu_x + \gamma)(x_k - x_0) \).

Throughout this section we make a continuity assumption on the stochastic first-order oracles similar to [17, 16, 26, 37].

**Assumption 6.** \( \exists L_{xxxx}, L_{xxxy}, L_{xxxy}, L_{yyyy} \geq 0 \) such that \( \forall x, \bar{x} \in \text{dom } f \subset \mathcal{X} \) and \( \forall y, \bar{y} \in \text{dom } g \subset \mathcal{Y}, \)
\[
\begin{align*}
    & \| \nabla_y \Phi(x, y; \omega) - \nabla_y \Phi(\bar{x}, \bar{y}; \omega) \| \leq L_{xxxx} \| x - \bar{x} \| + L_{xxxy} \| y - \bar{y} \|, \\    & \text{w.p. 1,} \\    & \| \nabla_y \Phi(x, y; \omega) - \nabla_y \Phi(\bar{x}, \bar{y}; \omega) \| \leq L_{xxxy} \| x - \bar{x} \| + L_{xxxy} \| y - \bar{y} \|, \\    & \text{w.p. 1.}
\end{align*}
\]

**Assumption 7.** Consider \textsc{SAPD}+ with \( VR\text{-flag} = \text{true} \). We assume (i) for any \( k \geq 0 \), the random mini-batches \( B^k, B^g \), \( T_k^b \) and \( T_k^g \) consist of independent elements, and \( B^k \) is independent from \( B^g \); (ii) for \( i \in \{ k - 1, k \} \), \( B^k \) is independent of \( (x_i, y_{i+1}) \), and \( B^g \), \( T_k^g \) are independent of \( (x_i, y_i) \).

**Remark 8.** 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 \( \nabla_x \Phi(x, y; \omega) = \nabla_x \Phi(x, y) \) and \( \nabla_y \Phi(x, y; \omega) = \nabla_y \Phi(x, y) \) where \( \omega \in \text{dom } f \) is uniformly drawn at random from \( \{1, \ldots, n\} \). Therefore, if mini-batch samples 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 Assumptions 1, 6 and 7 hold. Moreover, either Assumption 4 or Assumption 5 holds. Let \( \mu_x = \gamma, \theta = 1, \) and \( \tau, \sigma, b \) and \( N \) be chosen as follows:
\[
\tau = \left( L_{xxxx} + 2 \gamma + 2(q - 1) \left( \frac{L_{xxxx} + 2 \gamma^2}{\gamma b'_y} + \frac{10 L_{xxxy}^2}{\mu_x b'_y} \right) \right)^{-1}, \quad N = 2(1 + \zeta) \max\left\{ \frac{\gamma}{\gamma - 1}, \frac{1}{\mu_x \gamma^2} \right\},
\]
\[
\sigma = \left( 2 L_{xxxy} + 2 \gamma + 2(q - 1) \left( \frac{L_{xxxy}^2}{\gamma b'_x} + \frac{10 L_{xxxy}^2}{\mu_x b'_y} \right) \right)^{-1}, \quad b \geq \max\left\{ \frac{144 \sigma^2}{\gamma}, \frac{360 \sigma^2}{\gamma} \right\}.
\]

For any \( \epsilon > 0 \) and parameters \( b'_x, b'_y, q \in \mathbb{N} \), when \( VR\text{-flag} = \text{true} \), \textsc{SAPD}+ guarantees \( c \)-stationary, mini\(=0, \ldots, T \) \( \mathbb{E} \| \nabla \phi(x_T^n) \| \leq \epsilon, \) for \( T \geq 288 q \left( \frac{b'_x}{b'_y} \right)^2 \cdot \frac{\epsilon^2}{\gamma} \), which requires \( T \left( \frac{N b}{q} + N \left( b'_x + b'_y \right) \right) \) stochastic first-order oracle calls in total, where
\[
N = O\left( \max\left\{ \frac{L_{xxxx} + 2L_{xxxx}}{\gamma b'_y \gamma}, \frac{q L_{xxxy}^2}{\gamma b'_y \mu_x}, \frac{L_{xxxy} + L_{xxxx}}{\mu_y} + \frac{q L_{xxxy}^2}{b'_y \mu_y} + \frac{q L_{xxxy}^2}{b'_y \gamma \mu_y} \right\} \right).
\]

**Proof.** See appendix E for the proof. \( \square \)

**Remark 9.** For any \( y \in \text{dom } g \), since \( \Phi(\cdot, y) \) \( L_{xxxx} \)-smooth, it is necessarily \( L_{xxxx} \)-weakly convex; hence, \( \gamma \leq L_{xxxx} \). To get a worst-case complexity, consider the setting in (8), and let \( b'_y = b'_y = b' \).

Then, Theorem 4 implies that setting \( b = O\left( \frac{\kappa_y^2 \sigma^2}{\gamma} \right), \) \( N = O\left( \kappa_y^2 + \kappa_y^2 q \right), \) and \( T = O\left( \frac{L_{xxxx}(\gamma)^2}{\sigma^2} \right) \) leads to \( N b/q + N b' = O\left( \kappa_y \frac{b}{q} + \kappa_y^2 q + \kappa_y b' + \kappa_y^2 q \right) \). Thus, setting \( q = \sigma^2 \gamma \) and \( b' = \sqrt{\frac{b}{\sigma \gamma}} \) leads to the oracle complexity of \( T \left( \frac{N b}{q} + N b' \right) = O\left( \frac{\kappa_y^2 b}{q} + \frac{L_{xxxx}(\gamma)^2}{\sigma^2} \right) \).

**Remark 10.** The results in Theorem 4 continues to hold under a weaker form of Assumption 6 as in [26, 37], i.e., we replace eq. (10) with
\[
\begin{align*}
    \mathbb{E} \left[ \| \nabla_y \Phi(x, y; \omega) - \nabla_y \Phi(\bar{x}, \bar{y}; \omega) \|^2 \right] & \leq 2 L_{xxxx} \| x - \bar{x} \|^2 + 2 L_{xxxy} \| y - \bar{y} \|^2, \\    \mathbb{E} \left[ \| \nabla_y \Phi(x, y; \omega) - \nabla_y \Phi(\bar{x}, \bar{y}; \omega) \|^2 \right] & \leq 2 L_{xxxy} \| x - \bar{x} \|^2 + 2 L_{xxxy} \| y - \bar{y} \|^2.
\end{align*}
\]

### 5 Weakly convex-momarily concave (WCMC) problems

In this section, we state the convergence guarantees of \textsc{SAPD}+ for solving WCMC problems. In particular, we will consider (1) such that \( \tilde{f}(\cdot) = 0 \) 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 \( \tilde{y} \in \text{dom } g \), consider
\[
\begin{align*}
    \min_x \max_y \tilde{L}(x, y) & \triangleq \Phi(x, y) - g(y), \quad \text{where } \tilde{\Phi}(x, y) \triangleq \Phi(x, y) - \frac{\mu_y}{2} \| y - \tilde{y} \|^2.
\end{align*}
\]

8
Theorem 5. Under Assumptions 1, 2, 3 consider (1) such that \( f(x) \equiv 0 \), \( \mu_y = 0 \), and \( D_y \triangleq \sup_{y_1, y_2 \in \text{dom } g} \| y_1 - y_2 \| \leq \infty \). When either Assumption 3 or Assumption 5 holds, for any given \( \epsilon > 0 \), SAPD+ with VR-flag = false, applied to (13) with \( \mu_y = \Theta((\epsilon^2/|L|)^2) \), is guaranteed to generate \( x_\epsilon \in X \) such that \( \mathbb{E}[\| \nabla \phi_\lambda(x_\epsilon) \|] \leq \epsilon \) for \( \lambda = 1/(2\gamma) \) within \( O(L^3\epsilon^{-5}) \) stochastic first-order oracle calls.

Proof. See appendix G for the proof.

6 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. In this section, SAPD+ means calling SAPD+ with VR-flag=false, and SAPD+VR means calling SAPD+ with VR-flag=true.

Distributionally Robust Optimization (DRO). First, we consider nonconvex-regularized variant of DRO problem \( \mathbf{1}_{23} \) which arises in distributionally robust learning. Let \( \{a_i, b_i\}_{i=1}^n \) be the dataset where \( a_i \in \mathbb{R}^d \) are the features and \( b_i \in \{-1, 1\} \) are labels. The DRO problem is

\[
\min_{x \in \mathbb{R}^d} \max_{y \in Y} \frac{1}{n} \sum_{i=1}^n y_i \ell_i(x) + f(x) - g(y),
\]

(14)

where \( \ell_i(x) = \log(1 + \exp(-b_i a_i^\top x)) \) is the logistic loss, \( f(x) = \eta_1 \sum_{i=1}^d \alpha x_i^2 \) is a nonconvex regularizer \( \mathbf{2} \), \( Y \triangleq \{y \in \mathbb{R}_2^d : 1^\top y = 1\} \). Theorem 5. Under Assumptions 1, 2, 3 consider (1) such that \( f(x) \equiv 0 \), \( \mu_y = 0 \), and \( D_y \triangleq \sup_{y_1, y_2 \in \text{dom } g} \| y_1 - y_2 \| \leq \infty \). When either Assumption 3 or Assumption 5 holds, for any given \( \epsilon > 0 \), SAPD+ with VR-flag = false, applied to (13) with \( \mu_y = \Theta((\epsilon^2/|L|)^2) \), is guaranteed to generate \( x_\epsilon \in X \) such that \( \mathbb{E}[\| \nabla \phi_\lambda(x_\epsilon) \|] \leq \epsilon \) for \( \lambda = 1/(2\gamma) \) within \( O(L^3\epsilon^{-5}) \) stochastic first-order oracle calls.

Proof. See appendix G for the proof.

Parameter tuning. We set the parameters according to [40, 26, 20], i.e., \( \alpha = 10 \), \( \eta_1 = 10^{-3} \), \( \eta_2 = 1/n^2 \). We compare SAPD+ and SAPD+VR against PASGDA [4], SREDA [26], SMDA, SMDA-VR [17] 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 stepsizes to dual stepsize, i.e., \( \tau/\sigma \), is varied to take values from 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 \( b \triangleq |B| \) 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 [17]); for SREDA, when we set \( q \) and \( m \) (SREDA’s inner loop iteration number) to \( O(n/|I|) \) as suggested in [26], we noticed that SREDA does not perform well against SAPD+ 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 also use mini-batch to estimate the gradients and tune the batch size from \{10, 100, 200\} as well. For SAPD+ and SAPD+VR, we tune the momentum \( \theta \) from \{0.8, 0.85, 0.9\} and the inner iteration number from \( N = \{10, 50, 100\} \).

Results. To fairly compare the performances of algorithms using different batch sizes, we plot loss against epochs in x-axis. \(^3\) 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.

\(^3\)http://www.causality.inf.ethz.ch/challenge.php?page=datasets
\(^4\)https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html
\(^5\)an epoch is completed whenever an algorithm does one pass over the whole data set through sampling mini-bathes without replacement.
SAPD+ WCMC problems; moreover, our bound for We let the regularization parameter \( \eta \) CIFAR10 with target classes \{Plane, Truck, Deer\} from \( x \) where (15) on the datasets to classify: SAPD+ variance-reduction technique. Finally, we provided numerical experiments demonstrating that classification problems arising in ML. 

\( S^\text{APD+VR} \) = \( N \) for \( S^\text{APD+VR} \) and \( S^\text{MDA-VR} \) small batchsize \( |I| \) but, it caused convergence issues in the experiments. We set the large batchsize \( \tau/\sigma \) dual stepsizes, i.e., \( S^\text{APD+VR} \) and \( S^\text{MDA-VR} \) stepsizes are \( 10 \) for all algorithms and data sets; the frequency \( q = 200 \) is used for \( S^\text{APD+VR} \) and \( S^\text{MDA-VR} \), and we tune \( q \) for \( S^\text{REDA} \) taking values from \{10, 50, 100, 200\}. The momentum \( \theta \) for \( S^\text{APD+VR} \) is tuned taking values from \{0.8, 0.85, 0.9\} and inner iteration number is tuned from \( N = \{10, 50, 100\} \). For \( S^\text{REDA} \), we tune the inner loop iteration from \{10, 50, 100\}. Fig. 2 shows that \( S^\text{APD+VR} \) outperforms the other VR-based algorithms clearly in terms of both the average loss and the standard deviation of the loss.

7 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 \( \epsilon \) for both WCSC and WCMC problems; moreover, our bound for SAPD+ has a better dependency to the condition number \( \kappa_y \) for the WCSC scenario. We also showed that our algorithm SAPD+ can 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.

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 [32, 17], 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:

\[
\min_{x \in \mathbb{R}^p} \max_{y \in \mathbb{Y}} \sum_{i=1}^{3} y_i \ell_i(x) - g(y), \quad \text{s.t.} \quad \sum_{i=1}^{3} y_i = 1, \quad y_i \geq 0, \quad \forall \ i
\]

(15)

where \( x \in \mathbb{R}^p \) represents the parameters of the CNN, and \( \ell_1, \ell_2, \ell_3 \) correspond to the loss of three categories whose details are given in appendix [1]. \( g(y) = \frac{\eta}{2} \|y\|_2^2 \) 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 [17]. 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 \{0.1^{-2}, 5 \times 10^{-3}, 10^{-3}\} and the ratio of primal to dual stepsizes, i.e., \( \tau/\sigma \), 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 [17] —we also tried stepsizes bigger than the suggested; but, it caused convergence issues in the experiments. We set the large batchsize \( |B| = 3000 \) and the small batchsize \( |z| = 200 \) 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 \( \theta \) for SAPD+VR is tuned taking values from \{0.8, 0.85, 0.9\} and inner iteration number is tuned from \( N = \{10, 50, 100\} \). 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.
References


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 $\kappa_y$ dependence based on the existing lower complexity bounds [21][42], 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]