FINDING SECOND-ORDER STATIONARY POINTS OF NONCONVEX—STRONGLY-CONCAVE MINIMAX PROBLEMS UNDER WEAKER SMOOTHNESS ASSUMPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Most existing studies on computing second-order stationary points of minimax problems focus on the nonconvex—strongly-concave setting under higher-order smoothness assumptions. We adopt the notion of Goldstein approximate second-order stationarity and, for the first time, extend the study of second-order stationarity to the weaker setting where the objective is only continuously differentiable with a Lipschitz continuous gradient. Specifically, we propose a first-order method, called Smoothed Minimax Cubic Newton (SMCN), for finding an ε-Goldstein approximate second-order stationary point of the primal function. We show that with high probability, SMCN requires at most $\widetilde{\mathcal{O}}(\kappa^4\ell^4 n\varepsilon^{-5})$ outer iterations and $\widetilde{\mathcal{O}}(\kappa^{8.5}\ell^8 n^2\varepsilon^{-9} + \kappa^{6.5}\ell^6 n^3\varepsilon^{-7})$ first-order oracle calls, where ℓ denotes the Lipschitz constant of the gradient, κ is the condition number, and n is the primal dimension. We conduct numerical experiments to further demonstrate the effectiveness of the proposed method.

1 INTRODUCTION

In this paper, we consider the nonconvex–strongly-concave (NC–SC) minimax problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n}\max_{\boldsymbol{y}\in\mathbb{R}^m} f(\boldsymbol{x},\boldsymbol{y}), \tag{1.1}$$

where $f:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is smooth, and possibly nonconvex with respect to $\boldsymbol{x}$ and strongly concave with respect to $\boldsymbol{y}$. This formulation covers many applications in practice, including generative adversarial networks (Arjovsky et al., 2017; Goodfellow et al., 2014), adversarial training (Ganin et al., 2016; Sinha et al., 2017), and distributionally robust optimization (Gao & Kleywegt, 2023; Shafieezadeh-Abadeh et al., 2015). Problem (1.1) can be reformulated as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} P(\boldsymbol{x}), \qquad \text{where} \qquad P(\boldsymbol{x}) := \max_{\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}). \tag{1.2}$$

We refer to P as the primal function. First-order methods for solving (1.1) have been extensively studied. Existing approaches include the gradient descent–ascent (GDA) method (Lin et al., 2020), multi-step GDA (Nouiehed et al., 2019), and other variants (Zheng et al., 2025; 2023; Zhang et al., 2020; Mahdavinia et al., 2022; Jiang et al., 2024; Li et al., 2026). These methods can find an approximate first-order stationary point (FOSP) of the objective function f or the primal function P. However, without additional assumptions, the primal function P could be nonconvex, so an FOSP is not enough to ensure local optimality.

Most existing studies on computing second-order stationary points (SOSPs) for minimax problems focus on finding such points for the primal function P in the NC–SC setting. A predominant approach is to adopt a nested scheme, in which the inner maximization is approximately solved by a first-order method, while the outer minimization is handled by a classical second-order method, such as cubic regularization (Griewank, 1981; Nesterov & Polyak, 2006; Doikov et al., 2023; Cartis et al., 2011), trust-region methods (Yuan, 2015; Jiang et al., 2026; Curtis et al., 2021), or Levenberg–Marquardt methods (Marumo et al., 2025; Aravkin et al., 2024). To justify the outer second-order

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

model and control the errors induced by the approximate inner solutions, existing analyses typically require both the gradient and Hessian of f to be Lipschitz continuous (Chen et al., 2021; Luo et al., 2022; Wang & Xu, 2025; Chen et al., 2026). Recent work has relaxed these global Lipschitz assumptions through generalized smoothness conditions (Zhu et al., 2025). Nevertheless, the Hessian is still required to exist and have suitably controlled variation. Meanwhile, a recently proposed single-loop method eliminates the need to repeatedly solve the inner maximization problem, but it imposes the stronger requirement that f be thrice continuously differentiable (Ma et al., 2026).

These higher-order smoothness assumptions can be restrictive in applications. Once twice continuous differentiability is no longer imposed, the picture changes substantially. In the NC–SC setting, the first-order structure remains well behaved: Strong concavity of $f(\boldsymbol{x}, \cdot)$ and Lipschitz continuity of ∇f ensure that the primal function P has a Lipschitz continuous gradient. Such regularity, however, does not automatically extend to the second order, since P need not be twice differentiable. Consequently, Hessian-based notions of second-order stationarity and standard second-order methods are not directly available. Thus, a natural question arises:

> *Can we obtain a meaningful second-order stationarity concept for NC–SC minimax problems, without assuming that f is twice continuously differentiable?*

In this paper, we provide a first affirmative answer to the above question. We adopt the notion of Goldstein approximate second-order stationary points, previously studied in smooth nonconvex minimization (Guan & So, 2026a; Lei et al., 2026). We extend this line of work to NC–SC minimax problems and seek such points for the primal function P. Building on the framework of Guan & So (2026a), we develop a generic double-loop first-order method for computing these points. For smooth NC–SC minimax problems, our method finds with high probability an ε-Goldstein approximate SOSP of the primal function P in $\widetilde{\mathcal{O}}(\kappa^4\ell^4 n\varepsilon^{-5})$ iterations and $\widetilde{\mathcal{O}}(\kappa^{8.5}\ell^8 n^2\varepsilon^{-9}+\kappa^{6.5}\ell^6 n^3\varepsilon^{-7})$ first-order oracle calls, where ℓ, κ, and n denote the Lipschitz constant of ∇f, the condition number, and the primal dimension, respectively. Notably, this result does not require f to be twice continuously differentiable. Table 1 compares our results with closely related work.

Table 1: Comparison with related works in NC–SC minimax problems.

Method	**Smoothness of f**	**Stationary point**
Cubic-GDA (Chen et al., 2021)	Smooth with Lipschitz Hessian	SOSP of P
MCN (Luo et al., 2022)	Smooth with Lipschitz Hessian	SOSP of P
GRTR (Wang & Xu, 2025)	Smooth with Lipschitz Hessian	SOSP of P
HSDA (Chen et al., 2026)	Smooth with Lipschitz Hessian	SOSP of P
ANCGDA (Zhu et al., 2025)	Generalized smoothness*	SOSP of P
ACQRN (Ma et al., 2026)	C^3; smooth with Lipschitz Hessian	SOSP of P†
Theorem 4.3	**Smooth**	**GSOSP of P**

"smooth" means that ∇f is Lipschitz, and GSOSP denotes a Goldstein approximate SOSP.

* Both first-order and second-order generalized smoothness conditions are imposed on f.

† ACQRN finds an SOSP of $h_\beta := f(\boldsymbol{x}, \boldsymbol{y}) + \frac{\beta}{2}\|\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})\|^2$, which is transferred to P.

More broadly, the existing NC–SC theory reveals a clear hierarchy. At the nonsmooth end, Shi & Luo (2026) established Goldstein approximate FOSP guarantees for Lipschitz NC–SC minimax problems. For smooth NC–SC minimax problems, the theory of classical approximate FOSPs is already well developed (Lin et al., 2020). Under higher-order smoothness assumptions, existing methods provide classical approximate SOSP guarantees (Chen et al., 2021; Luo et al., 2022; Wang & Xu, 2025; Chen et al., 2026; Jiang et al., 2026; Zhu et al., 2025). This leaves a gap between the first-order theory for smooth problems and the second-order theory under stronger smoothness assumptions. Our result helps bridge this gap and provides a more complete picture of the computational landscape of different stationarity concepts for NC–SC minimax problems across different levels of smoothness.

2 PRELIMINARIES

Notation. We use the Euclidean norm for vectors and the spectral norm for matrices. The unit ball and unit sphere in $\mathbb{R}^n$ are denoted by $\mathbb{B}^n := \{\boldsymbol{x} : \|\boldsymbol{x}\| \le 1\}$ and $\mathbb{S}^n := \{\boldsymbol{x} : \|\boldsymbol{x}\| = 1\}$, respectively. The uniform distribution on $\mathcal{S}$ is denoted by $\mathrm{Unif}(\mathcal{S})$. We write $\mathrm{conv}(\mathcal{S})$ for the convex hull of $\mathcal{S}$ and $\mathrm{Sym}(A) := (A + A^\top)/2$ for the symmetric part of a square matrix. A function is L-smooth if its gradient is L-Lipschitz. We use $\mathrm{AGD}(\varphi, \boldsymbol{y}^0, T)$ to denote the output after applying T iterations of Nesterov's accelerated gradient descent method (AGD) to φ, initialized at $\boldsymbol{y}^0$.

2.1 SECOND-ORDER STATIONARY POINTS

First, we give the definition of approximate SOSPs for a twice continuously differentiable function.

Definition 2.1. (Luo et al., 2022, Definition 3) Given a twice continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ and constants $\varepsilon, \delta \ge 0$, we call $\boldsymbol{x}$ an (ε, δ)-SOSP of h if

$$\|\nabla h(\boldsymbol{x})\| \le \varepsilon \quad \text{and} \quad \nabla^2 h(\boldsymbol{x}) \succeq -\delta \boldsymbol{I}.$$

As $\varepsilon, \delta \downarrow 0$, the above condition recovers the second-order necessary conditions.

For an L-smooth function, by Rademacher's Theorem (Rockafellar & Wets, 1998, Theorem 9.60), its gradient is differentiable almost everywhere. Next, we introduce the Clarke generalized Hessian and Goldstein generalized Hessian of such a function.

Definition 2.2. Given an L-smooth function $h : \mathbb{R}^n \to \mathbb{R}$ and a point $\boldsymbol{x} \in \mathbb{R}^n$:

- The Clarke generalized Hessian of h at $\boldsymbol{x}$ is defined as

$$\partial_C^2 h(\boldsymbol{x}) := \mathrm{conv}\left(\limsup_{\boldsymbol{y} \to \boldsymbol{x},\, \boldsymbol{y} \in \mathbb{D}_h} \{\nabla^2 h(\boldsymbol{y})\} \right),$$

  where $\mathbb{D}_h$ consists of points at which h is twice differentiable; see (Hiriart-Urruty et al., 1984, Definition 2.1). By construction, the generalized Hessian is a nonempty compact convex set of symmetric matrices; see (Rockafellar & Wets, 1998, Theorem 13.52).
- Given a constant $\delta > 0$, the Goldstein generalized Hessian of h at $\boldsymbol{x}$ is defined as

$$\partial_\delta^2 h(\boldsymbol{x}) := \mathrm{conv}\left(\bigcup_{\boldsymbol{y} \in \boldsymbol{x} + \delta \mathbb{B}^n} \partial_C^2 h(\boldsymbol{y}) \right);$$

  see (Guan & So, 2026b, Definition 2.5).

Building on the following necessary optimality conditions, we introduce the corresponding notion of Goldstein approximate second-order stationary points.

Lemma 2.1. *(Hiriart-Urruty et al., 1984, Theorem 3.1) Let $\boldsymbol{x}$ be a local minimum of the L-smooth function $h : \mathbb{R}^n \to \mathbb{R}$. Then, for each $\boldsymbol{d} \in \mathbb{R}^n$, there exists a matrix $\boldsymbol{A} \in \partial_C^2 h(\boldsymbol{x})$ such that $\langle \boldsymbol{A}\boldsymbol{d}, \boldsymbol{d} \rangle \ge 0$. Equivalently, this condition can be written as*

$$\nabla h(\boldsymbol{x}) = 0 \quad \textit{and} \quad \min_{\boldsymbol{d} \in \mathbb{S}^n} \max_{\boldsymbol{A} \in \partial_C^2 h(\boldsymbol{x})} \langle \boldsymbol{A}\boldsymbol{d}, \boldsymbol{d} \rangle \ge 0.$$

Definition 2.3. (Guan & So, 2026a, Definition 2.5) Given an L-smooth function $h : \mathbb{R}^n \to \mathbb{R}$ and constants $\varepsilon_1, \varepsilon_2, \delta \ge 0$, we call $\boldsymbol{x}$ an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of h if

$$\|\nabla h(\boldsymbol{x})\| \le \varepsilon_1 \quad \text{and} \quad \min_{\boldsymbol{d} \in \mathbb{S}^n} \max_{\boldsymbol{A} \in \partial_\delta^2 h(\boldsymbol{x})} \langle \boldsymbol{A}\boldsymbol{d}, \boldsymbol{d} \rangle \ge -\varepsilon_2.$$

If $\varepsilon_1 = \varepsilon_2 = \delta = \varepsilon$ for some $\varepsilon \ge 0$, then we simply call $\boldsymbol{x}$ an ε-Goldstein approximate SOSP of h.

2.2 RANDOMIZED SMOOTHING

Randomized smoothing can smooth a Lipschitz function into a smooth function (Lin et al., 2022), and an L-smooth function into a function with a Lipschitz continuous Hessian (Guan & So, 2026a). Moreover, an SOSP of the smoothed function can be transformed into a Goldstein approximate SOSP of the original function. We present the results as follows.

Theorem 2.1. *(Guan & So, 2026a, Theorem 3.1, Proposition 2.5) Let $L, \sigma > 0$ be arbitrary and $h : \mathbb{R}^n \to \mathbb{R}$ be L-smooth. For $h_\sigma : \mathbb{R}^n \to \mathbb{R}$ given by*

$$h_\sigma(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{u}\sim \mathrm{Unif}(\mathbb{B}^n)}[h(\boldsymbol{x} + \sigma \boldsymbol{u})] \quad \text{for all } \boldsymbol{x} \in \mathbb{R}^n,$$

the following statements hold.

(1) $h_\sigma : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable.

(2) h_σ has an L-Lipschitz continuous gradient and

$$\nabla h_\sigma(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim \mathrm{Unif}(\mathbb{B}^n)}[\nabla h(\boldsymbol{x} + \sigma \boldsymbol{u})].$$

(3) h_σ has a $\frac{cL\sqrt{n}}{\sigma}$-Lipschitz continuous Hessian and

$$\nabla^2 h_\sigma(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim \mathrm{Unif}(\mathbb{B}^n)}[\nabla^2 h(\boldsymbol{x} + \sigma \boldsymbol{u})],$$

where $c > 0$ is some fixed constant.

(4) Let $\varepsilon_1, \varepsilon_2, \delta > 0$ be arbitrary. If $\boldsymbol{x}$ is an $(\varepsilon_1/3, \varepsilon_2)$-SOSP of h_σ, then $\boldsymbol{x}$ is an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of h, where $\sigma = \min\{\delta, \varepsilon_1/(3L)\}$.

3 Algorithmic Framework

In this section, given $\varepsilon_1, \varepsilon_2, \delta > 0$, we propose a first-order method for finding an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of (1.2). The proposed method extends the first-order framework of Guan and So (2026a) to smooth NC–SC minimax problems, with inexact gradient information of the primal function. First, we state the assumptions used throughout this paper.

Assumption 3.1.

(1) The function $(\boldsymbol{x}, \boldsymbol{y}) \mapsto f(\boldsymbol{x}, \boldsymbol{y})$ is μ-strongly concave in $\boldsymbol{y}$, i.e., there exists a constant $\mu > 0$ such that, for any $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^m$,

$$f(\boldsymbol{x}, \boldsymbol{y}) \le f(\boldsymbol{x}, \boldsymbol{y}') + \nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}')^\top (\boldsymbol{y} - \boldsymbol{y}') - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2.$$

(2) The function $(\boldsymbol{x}, \boldsymbol{y}) \mapsto f(\boldsymbol{x}, \boldsymbol{y})$ has an ℓ-Lipschitz continuous gradient, i.e., there exists a constant $\ell > 0$ such that, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ and $\boldsymbol{y}, \boldsymbol{y}' \in \mathbb{R}^m$,

$$\|\nabla f(\boldsymbol{x}, \boldsymbol{y}) - \nabla f(\boldsymbol{x}', \boldsymbol{y}')\|^2 \le \ell^2 \left(\|\boldsymbol{x} - \boldsymbol{x}'\|^2 + \|\boldsymbol{y} - \boldsymbol{y}'\|^2\right).$$

(3) For every $\boldsymbol{x} \in \mathbb{R}^n$, the inner maximum in (1.2) satisfies $P(\boldsymbol{x}) < +\infty$. Moreover, $P^\star := \inf_{\boldsymbol{x}\in\mathbb{R}^n} P(\boldsymbol{x}) > -\infty$.

Under Assumption 3.1, the primal function P has the following properties.

Lemma 3.1. *(Lin et al., 2020, Lemma 4.3) Suppose that Assumption 3.1 holds. Then, the following hold.*

(1) The solution map $\boldsymbol{x} \mapsto \boldsymbol{y}^\star(\boldsymbol{x})$ given by

$$\boldsymbol{y}^\star(\boldsymbol{x}) := \arg\max_{\boldsymbol{y}\in\mathbb{R}^m} f(\boldsymbol{x}, \boldsymbol{y}) \tag{3.1}$$

is uniquely defined and κ-Lipschitz continuous, where $\kappa := \ell/\mu$.

(2) The primal function P has a $(\kappa+1)\ell$-Lipschitz continuous gradient and

$$\nabla P(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f\big(\boldsymbol{x}, \boldsymbol{y}^\star(\boldsymbol{x})\big).$$

Consequently, by Theorem 2.1, randomized smoothing transforms the $(\kappa+1)\ell$-smooth function P into a twice differentiable function P_σ whose Hessian is M-Lipschitz continuous, where

$$M = \frac{c(\kappa+1)\ell\sqrt{n}}{\sigma}$$

for some $\sigma > 0$ and a fixed constant $c > 0$. On a high level, our proposed method has the following double-loop structure.

- **Inner loop:** For a given primal point $\boldsymbol{x}$, we apply AGD to $-f(\boldsymbol{x}, \cdot)$ to compute an approximate maximizer of the inner problem (3.1).
- **Outer loop:** We minimize the smoothed primal function P_σ using the cubic Newton method with inexact gradient and Hessian estimators. By finding an $(\varepsilon_g, \varepsilon_H)$-SOSP of P_σ with
$$\varepsilon_g = \frac{\varepsilon_1}{3}, \qquad \varepsilon_H = \varepsilon_2, \qquad \sigma = \min\left\{\delta, \frac{\varepsilon_1}{3(\kappa+1)\ell}\right\},$$
we obtain an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of P.

The main difficulty in the algorithmic implementation lies in constructing estimators of ∇P_σ and $\nabla^2 P_\sigma$. Although these quantities can be estimated using gradient evaluations of P, the gradient ∇P is not directly available, since $\nabla P(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f\big(\boldsymbol{x}, \boldsymbol{y}^\star(\boldsymbol{x})\big)$, where $\boldsymbol{y}^\star(\boldsymbol{x})$ is the exact solution of the inner maximization problem. In practice, we only obtain an approximate maximizer, and hence the resulting evaluations of ∇P are inexact. Therefore, the estimators of ∇P_σ and $\nabla^2 P_\sigma$ involve two sources of error: the randomized sampling error and the error caused by the inexact solution of the inner maximization problem.

To control the error, we employ a two-stage warm-start strategy. At the k-th iteration, we first initialize AGD at $\boldsymbol{y}_{k-1}$ and approximately solve the inner maximization problem at $\boldsymbol{x}_k$, obtaining $\boldsymbol{y}_k \approx \boldsymbol{y}^\star(\boldsymbol{x}_k)$. Then, for each randomly perturbed point $\boldsymbol{x}_k \pm \sigma \boldsymbol{u}_j$ and $\boldsymbol{x}_k \pm \sigma \boldsymbol{v}_j$, we initialize AGD at $\boldsymbol{y}_k$ and compute an approximate maximizer at the corresponding perturbed point. These approximate maximizers are subsequently used to evaluate $\nabla_{\boldsymbol{x}} f$ and construct the estimators of ∇P_σ and $\nabla^2 P_\sigma$.

The complete procedure is presented in Algorithm 1. In particular, we set
$$r_\star = \min\left\{\sqrt{\frac{16\varepsilon_g}{17M}}, \frac{24\varepsilon_H}{37M}\right\}.$$

We remark that the cubic subproblem (3.2) in Algorithm 1 can be solved globally using established methods. Cartis et al. (2011, Section 6.1) described an approach based on matrix factorization and scalar root finding. Practical solvers are available in the GALAHAD package RQS, with their numerical effectiveness demonstrated by Gould et al. (2010).

4 Convergence Results

4.1 Conditional Stationarity and Finite Termination

For a prescribed number of outer iterations K, define the estimator-accuracy event
$$\mathcal{E}_K := \bigcap_{k=0}^{K-1} \Big\{ \|\boldsymbol{g}(\boldsymbol{x}_k) - \nabla P_\sigma(\boldsymbol{x}_k)\| \le C_g,\ \|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \le C_H \Big\}, \tag{4.1}$$
where $C_g := Mr_\star^2/48$, and $C_H := Mr_\star/24$. Intuitively, $\mathcal{E}_K$ is the event that the gradient and Hessian estimators for P_σ remain sufficiently close to their exact counterparts throughout the first K outer iterations.

We first show that on the event $\mathcal{E}_K$, the stopping criterion in line 17 of Algorithm 1 yields the desired stationarity guarantee.

Theorem 4.1. *Suppose that Assumption 3.1 holds. On the event $\mathcal{E}_K$, if Algorithm 1 terminates, then its output is an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of P.*

Next, we show that Algorithm 1 terminates within a finite number of iterations.

Theorem 4.2. *Under Assumption 3.1, on the event $\mathcal{E}_{K_\star}$, where*
$$K_\star := \left\lceil \frac{24}{Mr_\star^3}\left(P_\sigma(\boldsymbol{x}_0) - \min_{\boldsymbol{x}\in\mathbb{R}^n} P_\sigma(\boldsymbol{x})\right)\right\rceil,$$
Algorithm 1 terminates after at most $K_\star$ iterations.

Algorithm 1: Smoothed Minimax Cubic Newton (SMCN) for NC–SC minimax problems

Input: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\boldsymbol{y}_{-1} = \mathbf{0}$, $\{T_k\}$, T_g, T_H, b_g, b_H
for $k = 0, 1, 2, \ldots$
  $\boldsymbol{y}_k = \mathrm{AGD}\big(-f(\boldsymbol{x}_k, \cdot), \boldsymbol{y}_{k-1}, T_k\big)$
  Sample $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{b_g} \sim \mathrm{Unif}(\mathbb{B}^n)$ independently
  for $j = 1, 2, \cdots, b_g$
    $\widetilde{\boldsymbol{y}}_{k_j}^{+} = \mathrm{AGD}\big(-f(\boldsymbol{x}_k + \sigma \boldsymbol{u}_j, \cdot), \boldsymbol{y}_k, T_g\big)$
    $\widetilde{\boldsymbol{y}}_{k_j}^{-} = \mathrm{AGD}\big(-f(\boldsymbol{x}_k - \sigma \boldsymbol{u}_j, \cdot), \boldsymbol{y}_k, T_g\big)$
  end for
  $\boldsymbol{g}(\boldsymbol{x}_k) = \dfrac{1}{2b_g} \displaystyle\sum_{j=1}^{b_g} \Big[\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k + \sigma \boldsymbol{u}_j, \widetilde{\boldsymbol{y}}_{k_j}^{+}) + \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k - \sigma \boldsymbol{u}_j, \widetilde{\boldsymbol{y}}_{k_j}^{-})\Big]$
  Sample $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{b_H} \sim \mathrm{Unif}(\mathbb{S}^n)$ independently
  for $j = 1, 2, \cdots, b_H$
    $\widehat{\boldsymbol{y}}_{k_j}^{+} = \mathrm{AGD}\big(-f(\boldsymbol{x}_k + \sigma \boldsymbol{v}_j, \cdot), \boldsymbol{y}_k, T_H\big)$
    $\widehat{\boldsymbol{y}}_{k_j}^{-} = \mathrm{AGD}\big(-f(\boldsymbol{x}_k - \sigma \boldsymbol{v}_j, \cdot), \boldsymbol{y}_k, T_H\big)$
  end for
  $\boldsymbol{H}(\boldsymbol{x}_k) = \dfrac{1}{b_H} \displaystyle\sum_{j=1}^{b_H} \mathrm{Sym}\Big[\frac{n}{2\sigma}\big(\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k + \sigma \boldsymbol{v}_j, \widehat{\boldsymbol{y}}_{k_j}^{+}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k - \sigma \boldsymbol{v}_j, \widehat{\boldsymbol{y}}_{k_j}^{-})\big)\boldsymbol{v}_j^{\top}\Big]$
  $\boldsymbol{d}_k = \arg\min_{\boldsymbol{d} \in \mathbb{R}^n} \Big\{\boldsymbol{g}(\boldsymbol{x}_k)^{\top}\boldsymbol{d} + \dfrac{1}{2}\boldsymbol{d}^{\top}\boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{d} + \dfrac{M}{6}\|\boldsymbol{d}\|_2^3\Big\}$ (3.2)
  if $\|\boldsymbol{d}_k\|_2 \le r_\star$ **then**
    $\widehat{\boldsymbol{x}} = \boldsymbol{x}_k + \boldsymbol{d}_k$
    break
  end if
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$
end for
return $\widehat{\boldsymbol{x}}$

Proof sketch. We prove Theorem 4.1 by contraposition. We first relate the step norm $\|\boldsymbol{d}_k\|_2$ to the gradient norm and smallest Hessian eigenvalue of P_σ. Then, these relations show that if the updated point is not an approximate SOSP, then $\|\boldsymbol{d}_k\|_2 > r_\star$. For Theorem 4.2, we argue by contradiction. We first establish the decrease in P_σ at each nonterminal iteration, and then use the fact that P_σ cannot fall below its minimum value to show that the algorithm must terminate within $K_\star$ iterations. The complete proofs are provided in Appendix B.

4.2 Inner Iteration Counts

Since AGD is invoked at three different stages of Algorithm 1, we first establish three corresponding lemmas that quantify the approximation accuracy achieved after a prescribed number of inner AGD iterations.

We first control the accuracy of the approximate maximizer at the current iterate.

Lemma 4.1. *Given $\varepsilon_{\boldsymbol{y}} > 0$, we run Algorithm 1 with*

$$T_0 = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\varepsilon_{\boldsymbol{y}}}\|\boldsymbol{y}^\star(\boldsymbol{x}_0)\|_2\right)\right\rceil$$

and, for $k \ge 1$,

$$T_k = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\varepsilon_{\boldsymbol{y}}}\left(\varepsilon_{\boldsymbol{y}} + \kappa\|\boldsymbol{d}_{k-1}\|_2\right)\right)\right\rceil.$$

Then, for $k \ge 0$,

$$\|\boldsymbol{y}_k - \boldsymbol{y}^\star(\boldsymbol{x}_k)\|_2 \le \varepsilon_{\boldsymbol{y}}.$$

Then, we establish the corresponding accuracy guarantees at the randomly perturbed points used in the gradient and Hessian estimators.

Lemma 4.2. *Letting $\widetilde{\varepsilon}_{\boldsymbol{y}} = \frac{Mr_\star^2}{432}$, we run Algorithm 1 with $T_g = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\ell\sqrt{\kappa+1}}{\widetilde{\varepsilon}_{\boldsymbol{y}}}(\varepsilon_{\boldsymbol{y}} + \kappa\sigma)\right)\right\rceil$, for $k \geq 0$. Then, for $k \geq 0$, we have*

$$\|\widetilde{\boldsymbol{y}}_{k_j}^{+} - \boldsymbol{y}^\star(\boldsymbol{x}_k + \sigma\boldsymbol{u}_j)\|_2 \leq \frac{\widetilde{\varepsilon}_{\boldsymbol{y}}}{\ell}, \qquad \|\widetilde{\boldsymbol{y}}_{k_j}^{-} - \boldsymbol{y}^\star(\boldsymbol{x}_k - \sigma\boldsymbol{u}_j)\|_2 \leq \frac{\widetilde{\varepsilon}_{\boldsymbol{y}}}{\ell}.$$

In the same way, we can derive the following result.

Lemma 4.3. *Letting $\widehat{\varepsilon}_{\boldsymbol{y}} = \frac{Mr_\star}{216}$, we run Algorithm 1 with $T_H = \left\lceil 2\sqrt{\kappa}\log\left(\frac{n\ell\sqrt{\kappa+1}}{\sigma\widehat{\varepsilon}_{\boldsymbol{y}}}(\varepsilon_{\boldsymbol{y}} + \kappa\sigma)\right)\right\rceil$, for $k \geq 0$. Then, for $k \geq 0$, we have*

$$\|\widehat{\boldsymbol{y}}_{k_j}^{+} - \boldsymbol{y}^\star(\boldsymbol{x}_k + \sigma\boldsymbol{v}_j)\|_2 \leq \frac{\sigma\widehat{\varepsilon}_{\boldsymbol{y}}}{n\ell}, \qquad \|\widehat{\boldsymbol{y}}_{k_j}^{-} - \boldsymbol{y}^\star(\boldsymbol{x}_k - \sigma\boldsymbol{v}_j)\|_2 \leq \frac{\sigma\widehat{\varepsilon}_{\boldsymbol{y}}}{n\ell}.$$

The proofs are provided in Appendix C.

4.3 High-Probability Convergence and Oracle Complexity

In view of Theorems 4.1 and 4.2, a key step is to ensure that $\mathcal{E}_{K_\star}$ holds with sufficiently high probability. Towards that end, we first establish the following error bounds for the gradient and Hessian estimators for P_σ.

Lemma 4.4. *With $\widetilde{\varepsilon}_{\boldsymbol{y}}$ as defined in Lemma 4.2, $\boldsymbol{g}(\boldsymbol{x}_k)$ as constructed in Algorithm 1, and $t > \widetilde{\varepsilon}_{\boldsymbol{y}}$, we have*

$$\mathbb{P}\left\{\|\boldsymbol{g}(\boldsymbol{x}_k) - \nabla P_\sigma(\boldsymbol{x}_k)\|_2 \geq t\right\} \leq 2\exp\left(-\frac{b_g(t-\widetilde{\varepsilon}_{\boldsymbol{y}})^2}{2(\kappa+1)^2\ell^2\sigma^2 + \frac{4}{3}(\kappa+1)\ell\sigma(t-\widetilde{\varepsilon}_{\boldsymbol{y}})}\right).$$

Lemma 4.5. *With $\widehat{\varepsilon}_{\boldsymbol{y}}$ as defined in Lemma 4.3, $\boldsymbol{H}(\boldsymbol{x}_k)$ as constructed in Algorithm 1, and $t > \widehat{\varepsilon}_{\boldsymbol{y}}$, we have*

$$\mathbb{P}\left\{\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \geq t\right\} \leq 2n\exp\left(-\frac{b_H(t-\widehat{\varepsilon}_{\boldsymbol{y}})^2/2}{C_{n\kappa\ell}^2 + C_{n\kappa\ell}(t-\widehat{\varepsilon}_{\boldsymbol{y}})/3}\right),$$

where $C_{n\kappa\ell} := (n+1)(\kappa+1)\ell$.

Next, we show that by choosing the sample sizes b_g and b_H judiciously, the event $\mathcal{E}_{K_\star}$ holds with high probability.

Lemma 4.6. *Given $\eta \in (0,1)$, set*

$$b_g = \left\lceil\left(\frac{5832(\kappa+1)^2\ell^2\sigma^2}{M^2r_\star^4} + \frac{72(\kappa+1)\ell\sigma}{Mr_\star^2}\right)\log\left(\frac{4K_\star}{\eta}\right)\right\rceil,$$
$$b_H = \left\lceil\left(\frac{1458C_{n\kappa\ell}^2}{M^2r_\star^2} + \frac{18C_{n\kappa\ell}}{Mr_\star}\right)\log\left(\frac{4nK_\star}{\eta}\right)\right\rceil,$$

in Algorithm 1, where $C_{n\kappa\ell} := (n+1)(\kappa+1)\ell$. Then, we have $\mathbb{P}\{\mathcal{E}_{K_\star}\} \geq 1-\eta$.

The proofs are provided in Appendix D.

Combining this high-probability guarantee with Theorems 4.1 and 4.2, we obtain the convergence guarantee of Algorithm 1.

Theorem 4.3. *Suppose that Assumption 3.1 holds. Run Algorithm 1 with T_k, T_g, and T_H chosen according to Lemmas 4.1, 4.2, and 4.3, respectively, and with sample sizes b_g, b_H chosen according to Lemma 4.6. Then, with probability at least $1-\eta$, Algorithm 1 returns an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of P in at most $K_\star$ outer iterations, where $K_\star$ is given in Theorem 4.2.*

We further derive the corresponding iteration and first-order oracle complexities when ε_1, ε_2, and δ are of the same order and $\kappa \gg 1$.

Corollary 4.1. *Run Algorithm 1 as described in Theorem 4.3 to get an $(\varepsilon_1, \varepsilon_2, \delta)$-approximate Goldstein SOSP of P, and suppose that $\varepsilon_1 = \varepsilon_2 = \delta = \varepsilon$ for some sufficiently small $\varepsilon > 0$. Then, the following hold.*

(1) The number of outer iterations is at most $\widetilde{\mathcal{O}}(\kappa^4 \ell^4 n \varepsilon^{-5})$.

(2) The number of gradient oracle calls is at most $\widetilde{\mathcal{O}}(\kappa^{8.5} \ell^8 n^2 \varepsilon^{-9} + \kappa^{6.5} \ell^6 n^3 \varepsilon^{-7})$.

Here, $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic terms in the complexity and $\Theta(\cdot)$ hides constant factors.

The complexity derivation is provided in Appendix E.

5 Numerical Experiments

We compare SMCN (Algorithm 1) with GDA (Lin et al., 2020) on a synthetic NC–SC minimax problem. Since both methods use only first-order information, GDA serves as a baseline for assessing the benefit of incorporating curvature estimates in escaping non-minimizing stationary points.

Problem and initialization. We consider

$$\min_{\boldsymbol{x} \in \mathbb{R}^3} \max_{\boldsymbol{y} \in \mathbb{R}^2} \omega(\boldsymbol{x}_3) + \boldsymbol{x}_1 \boldsymbol{y}_1 + \boldsymbol{x}_2 \boldsymbol{y}_2 - \frac{1}{40} \boldsymbol{y}_1^2 - \frac{5}{2} \boldsymbol{y}_2^2,$$

whose primal function is $P(\boldsymbol{x}) = \omega(\boldsymbol{x}_3) + 10\boldsymbol{x}_1^2 + 0.1\boldsymbol{x}_2^2$. The definition of ω, the reference value $P^\star$, and the experiment parameters are given in Appendix F. We should emphasize that ω is not twice continuously differentiable. Thus, existing second-order methods for NC-SC minimax problems cannot be applied directly. We use two initializations:

$$(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1) = ([10^{-5}, 10^{-5}, 10^{-5}]^\top, [0, 0]^\top),$$
$$(\widetilde{\boldsymbol{x}}_2, \widetilde{\boldsymbol{y}}_2) = ([10^{-5}, 10^{-5}, 0.8 - 10^{-5}]^\top, [0, 0]^\top).$$

The first lies near $\boldsymbol{x}_3 = 0$, a strict local maximizer of ω; the corresponding stationary point of P has a smallest Hessian eigenvalue of -0.1. The second lies just inside the boundary $\boldsymbol{x}_3 = 0.8$ of the flat region, a non-strict local maximizer of ω. At this boundary, P is not twice differentiable: the curvature along $\boldsymbol{x}_3$ is -0.005 on the inner side and zero on the outer side. Thus, the two initializations allow us to examine escape from a stationary point with negative curvature and from a non-minimizing stationary point where the Hessian does not exist, respectively.

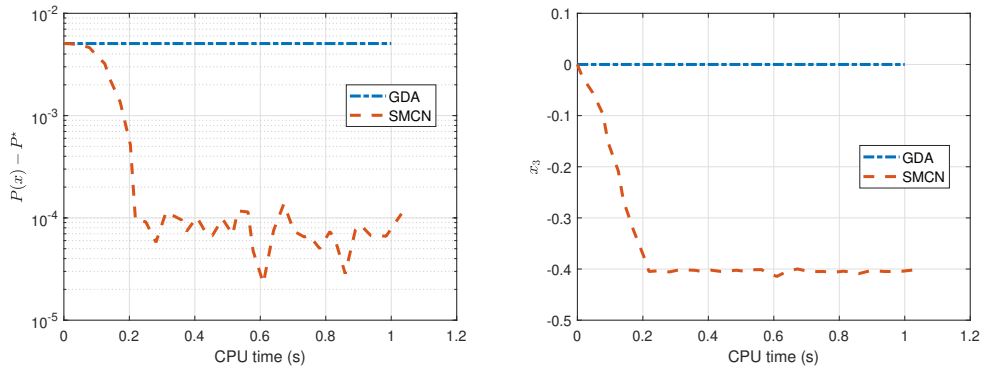

Figure 1: Performance comparison of GDA and SMCN when starting from $(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)$. (Left) Evolution of the optimality gap $P(\boldsymbol{x}) - P^\star$ over CPU time. (Right) Trajectory of the variable $\boldsymbol{x}_3$ over CPU time.

Results. We plot the gap $P(\boldsymbol{x}) - P^\star$ and the coordinate $\boldsymbol{x}_3$ against CPU time in Figures 1 and 2, corresponding to the two initializations. In both cases, GDA moves slowly because the gradient along $\boldsymbol{x}_3$ is small near the initial point. SMCN leaves these neighborhoods and substantially

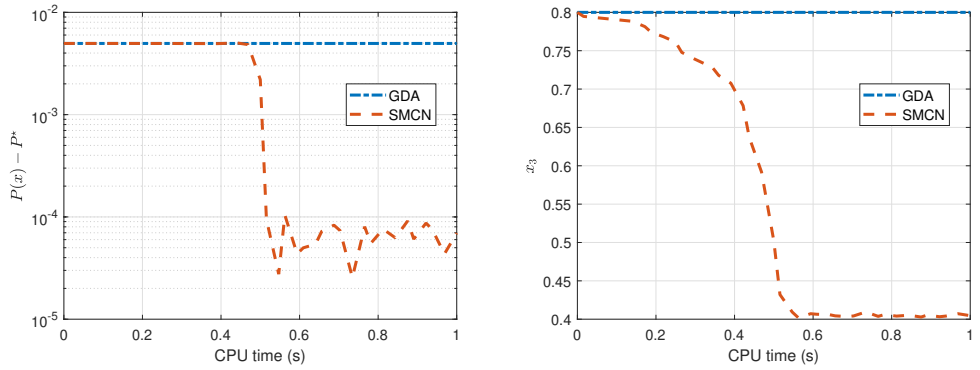

Figure 2: Performance comparison of GDA and SMCN when starting from $(\widetilde{\boldsymbol{x}}_2, \widetilde{\boldsymbol{y}}_2)$. (Left) Evolution of the optimality gap $P(\boldsymbol{x}) - P^\star$ over CPU time. (Right) Trajectory of the variable $\boldsymbol{x}_3$ over CPU time.

reduces the gap over the same time interval. Since the nonconvexity of P comes entirely from $\omega(\boldsymbol{x}_3)$, Figure 3 shows the corresponding trajectories on the landscape of ω: SMCN approaches a strict local minimizer, while GDA remains near the initial stationary region. SMCN takes longer to escape in the second case, consistent with the weaker negative curvature near the boundary of the flat region. These observations illustrate the effectiveness of SMCN in escaping non-minimizing stationary regions, even near a point where classical Hessian-based second-order conditions are not defined.

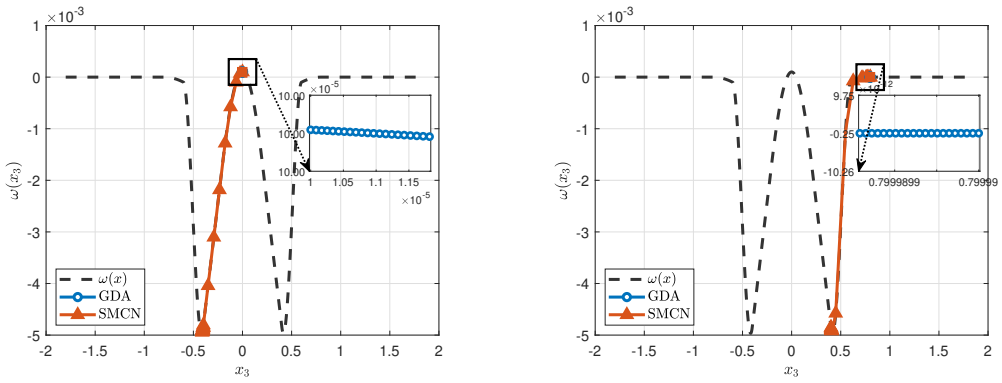

Figure 3: Trajectories of $\boldsymbol{x}_3$ on the landscape of ω when starting from $(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)$ (left) and $(\widetilde{\boldsymbol{x}}_2, \widetilde{\boldsymbol{y}}_2)$ (right).

6 Conclusion

Our results show for the first time that meaningful second-order stationarity guarantees remain attainable for NC–SC minimax problems even when the objective function is not twice continuously differentiable. Under Lipschitz continuity of the gradient, the proposed SMCN method computes a Goldstein approximate SOSP of the primal function using only first-order information of the minimax objective. By controlling both randomized estimation errors and errors introduced by inexact inner maximization, we established high-probability convergence together with explicit iteration and first-order oracle complexity bounds. Our numerical results further illustrate the ability of SMCN to leave neighborhoods of non-minimizing stationary points. These results help bridge the gap between the first-order theory for smooth NC–SC minimax problems and the second-order theory under higher-order smoothness assumptions, thus providing a more complete picture of the computational landscape of different stationarity concepts in NC–SC minimax optimization.

AI USE STATEMENT

In this work, we used generative AI tools for adjusting the manuscript to comply with formatting requirements and checking for typos, formatting issues, and grammatical errors. We have not used generative AI tools for any tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

REFERENCES

A. Y. Aravkin, R. Baraldi, and D. Orban. A Levenberg–Marquardt method for nonsmooth regularized least squares. *SIAM Journal on Scientific Computing*, 46(4):A2557–A2581, 2024.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, 2017.

C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

J. Chen, Z. Xu, and H. Zhang. A homogeneous second-order descent ascent algorithm for nonconvex-strongly concave minimax problems. *arXiv preprint arXiv:2602.14058*, 2026.

Z. Chen, Q. Li, and Y. Zhou. Escaping saddle points in nonconvex minimax optimization via cubic-regularized gradient descent-ascent. *arXiv preprint arXiv: 2110.07098*, 2021.

F. E. Curtis, D. P. Robinson, C. W. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM Journal on Optimization*, 31(1):518–544, 2021.

N. Doikov, E. M. Chayti, and M. Jaggi. Second-order optimization with lazy Hessians. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 8138–8161, 2023.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

R. Gao and A. Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

N. I. Gould, D. P. Robinson, and H. S. Thorne. On solving trust-region and other regularised subproblems in optimization. *Mathematical Programming Computation*, 2(1):21–57, 2010.

A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 1981.

J. Guan and A. M.-C. So. On computing Goldstein approximate second-order stationary points of structured nonsmooth nonconvex programs. *arXiv preprint arXiv:2607.24122*, 2026a.

J. Guan and A. M.-C. So. On the hardness of deterministic second-order optimization of functions with Lipschitz gradients. *arXiv preprint arXiv:2607.24120*, 2026b.

J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen. Generalized Hessian matrix and second-order optimality conditions for problems with $c^{1,1}$ data. *Applied Mathematics and Optimization*, 11(1): 43–56, 1984.

X. Jiang, L. Zhu, A. M.-C. So, S. Cui, and J. Sun. Shuffling gradient descent-ascent with variance reduction for nonconvex–strongly concave smooth minimax problems. *arXiv preprint arXiv:2410.04761*, 2024.

Y. Jiang, C. He, C. Zhang, D. Ge, B. Jiang, and Y. Ye. Beyond nonconvexity: A universal trust-region method with new analyses. *Journal of Scientific Computing*, 106(1):28, 2026.

M. Lei, T. K. Pong, M.C. Yue, L. Zeng, and H. Zhang. A Gaussian smoothing-based zeroth-order method for Goldstein second-order stationarity. *arXiv preprint arXiv:2607.21258*, 2026.

J. Li, M. Nagarajan, S. Pan, and N. Zhang. Smoothing meets perturbation: Unified and tight analysis for nonconvex-concave minimax optimization. *arXiv preprint arXiv:2602.14185*, 2026.

T. Lin, C. Jin, and M. I. Jordan. On gradient descent ascent for nonconvex–concave minimax problems. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6083–6093, 2020.

T. Lin, Z. Zheng, and M. I. Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26160–26175, 2022.

L. Luo, Y. Li, and C. Chen. Finding second-order stationary points in nonconvex–strongly-concave minimax optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36667–36679, 2022.

B. Ma, N. Xiao, and J. Zhang. A single-loop regularized Newton method for nonconvex–strongly-concave minimax optimization. *arXiv preprint arXiv:2606.07003*, 2026.

P. Mahdavinia, Y. Deng, H. Li, and M. Mahdavi. Tight analysis of extra-gradient and optimistic gradient methods for nonconvex minimax problems. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31213–31225, 2022.

N. Marumo, T. Okuno, and A. Takeda. Accelerated-gradient-based generalized Levenberg–Marquardt method with oracle complexity bound and local quadratic convergence. *Mathematical Programming*, 213(1):771–822, 2025.

Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, volume 32, pp. 14905–14916, 2019.

R. T. Rockafellar and R. J. Wets. *Variational analysis*. Springer, 1998.

S. Shafieezadeh-Abadeh, P. M. Mohajerin Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1576–1584, 2015.

J. Shi and L. Luo. Nonsmooth nonconvex-concave minimax optimization: Convergence criteria and algorithms. *arXiv preprint arXiv:2604.21371*, 2026.

A. Sinha, H. Namkoong, R. Volpi, and J. Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

J. Wang and Z. Xu. Gradient norm regularization second-order algorithms for solving nonconvex–strongly concave minimax problems. *Journal of Scientific Computing*, 105(2):41, 2025.

Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151(1):249–281, 2015.

J. Zhang, P. Xiao, R. Sun, and Z. Luo. A single-loop smoothed gradient descent-ascent algorithm for nonconvex-concave min-max problems. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7377–7389, 2020.

T. Zheng, L. Zhu, A. M.-C. So, J. Blanchet, and J. Li. Universal gradient descent ascent method for nonconvex–nonconcave minimax optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 54075–54110, 2023.

T. Zheng, A. M.-C. So, and J. Li. Doubly smoothed optimistic gradients: A universal approach for smooth minimax problems. *arXiv preprint arXiv:2506.07397*, 2025.

Q. Zhu, Y. Liu, F. Shang, H. Liu, and Z. Lin. Finding second-order stationary points for generalized-smooth nonconvex minimax optimization via gradient-based algorithm, 2025. URL `https://openreview.net/forum?id=feZcpZbbqL`. OpenReview preprint.

A Appendix

A.1 Supporting Preliminary Results

A.1.1 Gradient and Hessian Estimators

Guan and So (2026a) constructed unbiased estimators for the gradient and Hessian of the smoothed function and established bounds on their estimation errors.

Lemma A.1. *(Guan & So, 2026a, Proposition 3.2) Let $\sigma > 0$ be arbitrary and $h : \mathbb{R}^n \to \mathbb{R}$ be L-smooth. Then,*

$$g_b(\boldsymbol{x}) := \frac{1}{b}\sum_{j=1}^{b} \frac{\nabla h(\boldsymbol{x}+\sigma \boldsymbol{u}_j) + \nabla h(\boldsymbol{x}-\sigma \boldsymbol{u}_j)}{2}$$

is an unbiased approximation of $\nabla h_\sigma(\boldsymbol{x})$, where $b = 1, 2, \cdots$ and $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_b \sim \mathrm{Unif}(\mathbb{B}^{\mathrm{n}})$ independently. Moreover, the approximation satisfies

$$\mathbb{P}\{\|g_b(\boldsymbol{x}) - \nabla h_\sigma(\boldsymbol{x})\| \geq t\} \leq 2\exp\left(-\frac{bt^2}{2L^2\sigma^2 + 4L\sigma t/3}\right) \quad \textit{for all} \quad t > 0.$$

Lemma A.2. *(Guan & So, 2026a, Proposition 3.4) Let $\sigma > 0$ be arbitrary and $h : \mathbb{R}^n \to \mathbb{R}$ be L-smooth. Then,*

$$H_b(\boldsymbol{x}) := \frac{1}{b}\sum_{j=1}^{b} \mathrm{Sym}\left(\frac{n}{2\sigma}(\nabla h(\boldsymbol{x}+\sigma \boldsymbol{s}_j) - \nabla h(\boldsymbol{x}-\sigma \boldsymbol{s}_j))\boldsymbol{s}_j^\top\right)$$

is an unbiased approximation of $\nabla^2 h_\sigma(\boldsymbol{x})$, where $b = 1, 2, \cdots$ and $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_b \sim \mathrm{Unif}(\mathbb{S}^{\mathrm{n}})$ independently. Moreover, the approximation satisfies that

$$\mathbb{P}\{\|H_b(\boldsymbol{x}) - \nabla^2 h_\sigma(\boldsymbol{x})\| \geq t\} \leq 2n\exp\left(-\frac{bt^2/2}{(n+1)^2L^2 + (n+1)Lt/3}\right) \quad \textit{for all} \quad t > 0.$$

A.1.2 Other Basic Results

In this subsection, we summarize some basic results.

Lemma A.3. *(Luo et al., 2022, Lemma 5) For an ℓ-smooth and μ-strongly convex function $\varphi(y)$ with $\kappa = \ell/\mu$, we run Nesterov's accelerated gradient descent as follows:*

$$\begin{aligned}
\boldsymbol{y}_{t+1} &= \widetilde{\boldsymbol{y}}_t - \frac{1}{\ell}\nabla\varphi(\widetilde{\boldsymbol{y}}_t), \\
\widetilde{\boldsymbol{y}}_{t+1} &= \boldsymbol{y}_{t+1} + \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}(\boldsymbol{y}_{t+1} - \boldsymbol{y}_t),
\end{aligned}$$

for $t = 0, 1, 2, \cdots, T-1$. Then the output $\boldsymbol{y}_T$ satisfies that

$$\|\boldsymbol{y}_T - \boldsymbol{y}^\star\|_2^2 \leq (\kappa+1)(1 - \frac{1}{\sqrt{\kappa}})^T\|\boldsymbol{y}_0 - \boldsymbol{y}^\star\|_2^2,$$

where $\boldsymbol{y}^\star = \arg\min_{\boldsymbol{y}} \varphi(\boldsymbol{y})$. We denote the output as $\boldsymbol{y}_T = \mathrm{AGD}(\varphi, \boldsymbol{y}_0, T)$.

Lemma A.4. *(Nesterov & Polyak, 2006, Lemma 1) For a twice continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ with an L-Lipschitz continuous Hessian, we have*

$$\|\nabla h(\boldsymbol{y}) - \nabla h(\boldsymbol{x}) - \nabla^2 h(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})\| \leq \frac{L}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2, \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n,$$

and

$$\left|h(\boldsymbol{y}) - h(\boldsymbol{x}) - \langle\nabla h(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle - \frac{1}{2}\langle\nabla^2 h(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle\right| \leq \frac{L}{6}\|\boldsymbol{y}-\boldsymbol{x}\|^3, \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

B Proofs of the Conditional Guarantees

The optimality conditions of the subproblem (3.2) in Algorithm 1 are as follows.

Lemma B.1. *Let $\boldsymbol{d}_k \in \mathbb{R}^n$ be the solution of* (3.2)*. Then, it satisfies*

$$\boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{d}_k + \frac{M}{2}\|\boldsymbol{d}_k\|_2\boldsymbol{d}_k = \boldsymbol{0}, \tag{B.1}$$

$$\boldsymbol{H}(\boldsymbol{x}_k) + \frac{M}{2}\|\boldsymbol{d}_k\|\boldsymbol{I} \succeq \boldsymbol{0}, \tag{B.2}$$

$$\boldsymbol{g}(\boldsymbol{x}_k)^\top \boldsymbol{d}_k + \frac{1}{2}\boldsymbol{d}_k^\top \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{d}_k + \frac{M}{6}\|\boldsymbol{d}_k\|_2^3 \le -\frac{M}{12}\|\boldsymbol{d}_k\|_2^3. \tag{B.3}$$

For brevity, let

$$\boldsymbol{g}_k := \boldsymbol{g}(\boldsymbol{x}_k), \qquad \boldsymbol{H}_k := \boldsymbol{H}(\boldsymbol{x}_k),$$

and define

$$\boldsymbol{q}_k := \nabla P_\sigma(\boldsymbol{x}_k) - \boldsymbol{g}_k, \qquad \boldsymbol{Q}_k := \nabla^2 P_\sigma(\boldsymbol{x}_k) - \boldsymbol{H}_k.$$

B.1 Proof of Theorem 4.1

Proof. By the relationship between P and P_σ established in Theorem 2.1 and the definitions of ε_g and ε_H, it suffices to show that whenever the step $\boldsymbol{d}_k$ generated by Algorithm 1 satisfies $\|\boldsymbol{d}_k\|_2 \le r_\star$, the point $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k$ is an $(\varepsilon_g, \varepsilon_H)$-SOSP of P_σ.

Then, we have

$$\begin{aligned}
\|\nabla P_\sigma(\boldsymbol{x}_k + \boldsymbol{d}_k)\|_2 =& \left\|\nabla P_\sigma(\boldsymbol{x}_k) + \int_0^1 \nabla^2 P_\sigma(\boldsymbol{x}_k + t\boldsymbol{d}_k)\boldsymbol{d}_k dt\right\|_2 \\
=& \Bigg\|\boldsymbol{q}_k + \boldsymbol{g}_k + \int_0^1 \left(\nabla^2 P_\sigma(\boldsymbol{x}_k + t\boldsymbol{d}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\right)\boldsymbol{d}_k dt \\
&\qquad + \boldsymbol{Q}_k\boldsymbol{d}_k + \boldsymbol{H}_k\boldsymbol{d}_k\Bigg\|_2 \\
\le& C_g + C_H\|\boldsymbol{d}_k\|_2 + \frac{M}{2}\|\boldsymbol{d}_k\|_2^2 \\
&\qquad + \|\boldsymbol{d}_k\|_2 \int_0^1 \left\|\nabla^2 P_\sigma(\boldsymbol{x}_k + t\boldsymbol{d}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\right\|_2 dt \\
\le& C_g + C_H\|\boldsymbol{d}_k\|_2 + \frac{M}{2}\|\boldsymbol{d}_k\|_2^2 + \|\boldsymbol{d}_k\|_2^2 M\int_0^1 t dt \\
=& C_g + C_H\|\boldsymbol{d}_k\|_2 + M\|\boldsymbol{d}_k\|_2^2,
\end{aligned}$$

where the first inequality follows from the estimator bounds in (4.1) and the first-order optimality condition (B.1), while the second follows from the M-Lipschitz continuity of $\nabla^2 P_\sigma$. Furthermore, it implies

$$\|\boldsymbol{d}_k\|_2 \ge \frac{1}{2M}\left(\sqrt{C_H^2 + 4M\|\nabla P_\sigma(\boldsymbol{x}_{k+1})\|_2 - 4MC_g} - C_H\right). \tag{B.4}$$

By the M-Lipschitz continuity of $\nabla^2 P_\sigma$ and Weyl's inequality,

$$\lambda_{\min}\left(\nabla^2 P_\sigma(\boldsymbol{x}_{k+1})\right) \ge \lambda_{\min}\left(\nabla^2 P_\sigma(\boldsymbol{x}_k)\right) - M\|\boldsymbol{d}_k\|_2. \tag{B.5}$$

In the same way, using $\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \le C_H$, we have

$$\lambda_{\min}\left(\nabla^2 P_\sigma(\boldsymbol{x}_k)\right) \ge \lambda_{\min}(\boldsymbol{H}_k) - C_H. \tag{B.6}$$

Combining (B.5) and (B.6), we get

$$\begin{aligned}
\lambda_{\min}(\nabla^2 P_\sigma(\boldsymbol{x}_{k+1})) &\ge \lambda_{\min}(\boldsymbol{H}(\boldsymbol{x}_k)) - M\|\boldsymbol{d}_k\|_2 - C_H \\
&\ge -\frac{M}{2}\|\boldsymbol{d}_k\|_2 - M\|\boldsymbol{d}_k\|_2 - C_H \\
&= -\frac{3}{2}M\|\boldsymbol{d}_k\|_2 - C_H,
\end{aligned}$$

which implies that

$$\|\boldsymbol{d}_k\|_2 \geq -\frac{2}{3M}C_H - \frac{2}{3M}\lambda_{\min}(\nabla^2 P_\sigma(\boldsymbol{x}_{k+1})). \tag{B.7}$$

We prove the claim by contraposition. Suppose that $\boldsymbol{x}_{k+1}$ is not an $(\varepsilon_g, \varepsilon_H)$-SOSP of P_σ. Then at least one of the following holds:

$$\|\nabla P_\sigma(\boldsymbol{x}_{k+1})\|_2 > \varepsilon_g, \qquad \lambda_{\min}\big(\nabla^2 P_\sigma(\boldsymbol{x}_{k+1})\big) < -\varepsilon_H.$$

We distinguish two cases:

(1) If $\|\nabla P_\sigma(\boldsymbol{x}_{k+1})\|_2 > \varepsilon_g$, (B.4) implies that

$$\|\boldsymbol{d}_k\|_2 > \frac{1}{2M}\left(\sqrt{C_H^2 + 4M\varepsilon_g - 4MC_g} - C_H\right) \geq \sqrt{\frac{r_\star^2}{48^2} + \frac{17r_\star^2}{16} - \frac{r_\star^2}{48}} - \frac{r_\star}{48} = r_\star.$$

Here, the second inequality uses the definition of C_g and C_H, and the result $\varepsilon_g \geq 17Mr_\star^2/16$, which follows from the definition of $r_\star$.

(2) If $\lambda_{\min}(\nabla^2 P_\sigma(\boldsymbol{x}_{k+1})) < -\varepsilon_H$, (B.7) implies that

$$\|\boldsymbol{d}_k\|_2 > -\frac{2}{3M}C_H + \frac{2\varepsilon_H}{3M} \geq r_\star,$$

where the second inequality used the definition of C_H, and the result $\varepsilon_H \geq 37Mr_\star/24$, which follows from the definition of $r_\star$.

In either case, $\|\boldsymbol{d}_k\|_2 > r_\star$, proving the claim by contraposition. □

B.2 Proof of Theorem 4.2

Proof. We first establish the decrease in P_σ at each nonterminal iteration. By Lemma A.4,

$$\begin{aligned}
P_\sigma(\boldsymbol{x}_{k+1}) - P_\sigma(\boldsymbol{x}_k) &\leq \nabla P_\sigma(\boldsymbol{x}_k)^\top \boldsymbol{d}_k + \frac{1}{2}\boldsymbol{d}_k^\top \nabla^2 P_\sigma(\boldsymbol{x}_k)\boldsymbol{d}_k + \frac{M}{6}\|\boldsymbol{d}_k\|_2^3 \\
&= \boldsymbol{g}_k^\top \boldsymbol{d}_k + \frac{1}{2}\boldsymbol{d}_k^\top \boldsymbol{H}_k \boldsymbol{d}_k + \frac{M}{6}\|\boldsymbol{d}_k\|_2^3 + \boldsymbol{q}_k^\top \boldsymbol{d}_k + \frac{1}{2}\boldsymbol{d}_k^\top \boldsymbol{Q}_k \boldsymbol{d}_k \\
&\leq -\frac{M}{12}\|\boldsymbol{d}_k\|_2^3 + \|\boldsymbol{q}_k\|_2 \cdot \|\boldsymbol{d}_k\|_2 + \frac{1}{2}\|\boldsymbol{Q}_k\|_2 \cdot \|\boldsymbol{d}_k\|_2^2 \\
&\leq -\frac{M}{12}\|\boldsymbol{d}_k\|_2^3 + \frac{Mr_\star^2}{48}\|\boldsymbol{d}_k\|_2 + \frac{Mr_\star}{48}\|\boldsymbol{d}_k\|_2^2.
\end{aligned}$$

Here, the second inequality follows from (B.3) and the Cauchy–Schwarz inequality, while the last inequality follows from the estimator bounds in (4.1).

For every nonterminal iteration, the stopping criterion gives $\|\boldsymbol{d}_k\|_2 > r_\star$. Hence,

$$r_\star^2\|\boldsymbol{d}_k\|_2 + r_\star\|\boldsymbol{d}_k\|_2^2 \leq 2\|\boldsymbol{d}_k\|_2^3,$$

and therefore

$$P_\sigma(\boldsymbol{x}_{k+1}) - P_\sigma(\boldsymbol{x}_k) \leq -\frac{M}{24}\|\boldsymbol{d}_k\|_2^3 < -\frac{M}{24}r_\star^3. \tag{B.8}$$

We now show that Algorithm 1 must terminate within $K_\star$ iterations. Suppose, to the contrary, that it does not terminate during the first $K_\star$ iterations. Then $\|\boldsymbol{d}_k\|_2 > r_\star$ for $k = 0, \ldots, K_\star - 1$. Summing (B.8) yields

$$P_\sigma(\boldsymbol{x}_{K_\star}) < P_\sigma(\boldsymbol{x}_0) - \frac{MK_\star}{24}r_\star^3.$$

By the definition of $K_\star$,

$$\frac{MK_\star}{24}r_\star^3 \geq P_\sigma(\boldsymbol{x}_0) - \min_{\boldsymbol{x}\in\mathbb{R}^n} P_\sigma(\boldsymbol{x}),$$

and hence

$$P_\sigma(\boldsymbol{x}_{K_\star}) < \min_{\boldsymbol{x}\in\mathbb{R}^n} P_\sigma(\boldsymbol{x}),$$

which is impossible. Therefore, Algorithm 1 terminates after at most $K_\star$ iterations.

At termination, the corresponding step satisfies $\|\boldsymbol{d}_k\|_2 \leq r_\star$. By Theorem 4.1, the output $\widehat{\boldsymbol{x}}$ is therefore an $(\varepsilon_1, \varepsilon_2, \delta)$-Goldstein approximate SOSP of P. □

C Proofs of the Inner Accuracy Lemmas

C.1 Proof of Lemma 4.1

Proof. We proceed by induction. For $k = 0$, $\boldsymbol{y}_{-1} = \mathbf{0}$, so Lemma A.3 gives

$$\begin{aligned}\|\boldsymbol{y}_0 - \boldsymbol{y}^\star(\boldsymbol{x}_0)\|_2 &\le \sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_0/2}\|\boldsymbol{y}^\star(\boldsymbol{x}_0)\|_2\\ &\le \sqrt{\kappa+1}\exp\left(-\frac{T_0}{2\sqrt{\kappa}}\right)\|\boldsymbol{y}^\star(\boldsymbol{x}_0)\|_2\\ &\le \varepsilon_{\boldsymbol{y}},\end{aligned}$$

where the second inequality follows from $1 - t \le e^{-t}$ and the last one follows from the definition of T_0.

Suppose that, for some $k \ge 1$,

$$\|\boldsymbol{y}_{k-1} - \boldsymbol{y}^\star(\boldsymbol{x}_{k-1})\|_2 \le \varepsilon_{\boldsymbol{y}}.$$

Then, by Lemma A.3, the triangle inequality, and the κ-Lipschitz continuity of $\boldsymbol{y}^\star$,

$$\begin{aligned}\|\boldsymbol{y}_k - \boldsymbol{y}^\star(\boldsymbol{x}_k)\|_2 &\le \sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_k/2}\|\boldsymbol{y}_{k-1} - \boldsymbol{y}^\star(\boldsymbol{x}_k)\|_2\\ &\le \sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_k/2}(\|\boldsymbol{y}_{k-1} - \boldsymbol{y}^\star(\boldsymbol{x}_{k-1})\|_2 + \|\boldsymbol{y}^\star(\boldsymbol{x}_{k-1}) - \boldsymbol{y}^\star(\boldsymbol{x}_k)\|_2)\\ &\le \sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_k/2}(\varepsilon_{\boldsymbol{y}} + \kappa\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\|_2)\\ &= \sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_k/2}(\varepsilon_{\boldsymbol{y}} + \kappa\|\boldsymbol{d}_{k-1}\|_2)\\ &\le \sqrt{\kappa+1}\exp\left(-\frac{T_k}{2\sqrt{\kappa}}\right)(\varepsilon_{\boldsymbol{y}} + \kappa\|\boldsymbol{d}_{k-1}\|_2) \le \varepsilon_{\boldsymbol{y}},\end{aligned}$$

where the last inequality follows from the definition of T_k. Hence, the result follows by induction. □

C.2 Proof of Lemma 4.2

Proof. According to Lemma A.3, we have

$$\|\widetilde{\boldsymbol{y}}_{k_j}^+ - \boldsymbol{y}^\star(\boldsymbol{x}_k + \sigma\boldsymbol{u}_j)\|_2 \le \sqrt{(\kappa+1)}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{T_g/2}\|\boldsymbol{y}_k - \boldsymbol{y}^\star(\boldsymbol{x}_k + \sigma\boldsymbol{u}_j)\|_2.$$

Furthermore,

$$\|\boldsymbol{y}_k - \boldsymbol{y}^\star(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)\|_2 \le \|\boldsymbol{y}_k - \boldsymbol{y}^\star(\boldsymbol{x}_k)\|_2 + \|\boldsymbol{y}^\star(\boldsymbol{x}_k) - \boldsymbol{y}^\star(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)\|_2 \le \varepsilon_{\boldsymbol{y}} + \kappa\sigma\|\boldsymbol{u}_j\|_2 \le \varepsilon_{\boldsymbol{y}} + \kappa\sigma,$$

where the first inequality uses the triangle inequality, and the second inequality is based on Lemma 4.1 and the Lipschitz continuity of $\boldsymbol{y}^\star$. The last inequality is due to $\|\boldsymbol{u}_j\| \le 1$. Moreover, using $1 - \boldsymbol{x} \le e^{-\boldsymbol{x}}$, we have

$$\|\widetilde{\boldsymbol{y}}_{k_j}^+ - \boldsymbol{y}^\star(\boldsymbol{x}_k + \sigma\boldsymbol{u}_j)\|_2 \le \sqrt{\kappa+1}\exp\left(-\frac{T_g}{2\sqrt{\kappa}}\right)(\varepsilon_{\boldsymbol{y}} + \kappa\sigma) \le \frac{\widetilde{\varepsilon}_{\boldsymbol{y}}}{\ell}.$$

Finally, in the same way, we can prove the second part of Lemma 4.2. □

C.3 Proof of Lemma 4.3

The proof follows the same argument as that of Lemma 4.2, with $\widetilde{\varepsilon}_{\boldsymbol{y}}/\ell$ replaced by $\sigma\widehat{\varepsilon}_{\boldsymbol{y}}/(n\ell)$.

D Estimator Accuracy and Sample Sizes

Building on the above accuracy guarantees for the inner AGD iterations, we next establish error bounds for the gradient and Hessian estimators with respect to the exact quantities ∇P_σ and $\nabla^2 P_\sigma$.

D.1 Proof of Lemma 4.4

Proof. Define

$$\widehat{\boldsymbol{g}}(\boldsymbol{x}_k) := \frac{1}{b_g}\sum_{j=1}^{b_g}\frac{\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)+\nabla P(\boldsymbol{x}_k-\sigma\boldsymbol{u}_j)}{2}.$$

By Lemma A.1, $\widehat{\boldsymbol{g}}(\boldsymbol{x}_k)$ is an unbiased estimator of $\nabla P_\sigma(\boldsymbol{x}_k)$.

For brevity, write $\boldsymbol{g}(\boldsymbol{x}_k) = \frac{1}{b_g}\sum_{j=1}^{b_g}\boldsymbol{g}_j(\boldsymbol{x}_k)$ and $\widehat{\boldsymbol{g}}(\boldsymbol{x}_k) = \frac{1}{b_g}\sum_{j=1}^{b_g}\widehat{\boldsymbol{g}}_j(\boldsymbol{x}_k)$. By the triangle inequality,

$$\begin{aligned}\|\boldsymbol{g}_j(\boldsymbol{x}_k)-\widehat{\boldsymbol{g}}_j(\boldsymbol{x}_k)\|_2 \le \frac{1}{2}\Big(&\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j,\widetilde{\boldsymbol{y}}^+_{k_j})-\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)\|_2\\ &+\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k-\sigma\boldsymbol{u}_j,\widetilde{\boldsymbol{y}}^-_{k_j})-\nabla P(\boldsymbol{x}_k-\sigma\boldsymbol{u}_j)\|_2\Big).\end{aligned} \tag{D.1}$$

Since $\nabla P(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{y}^\star(\boldsymbol{x}))$, Assumption 3.1 and Lemma 4.2 imply

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j,\widetilde{\boldsymbol{y}}^+_{k_j})-\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)\|_2 \le \ell\|\widetilde{\boldsymbol{y}}^+_{k_j}-\boldsymbol{y}^\star(\boldsymbol{x}_k+\sigma\boldsymbol{u}_j)\|_2 \le \widetilde{\varepsilon}_{\boldsymbol{y}}.$$

Similarly,

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k-\sigma\boldsymbol{u}_j,\widetilde{\boldsymbol{y}}^-_{k_j})-\nabla P(\boldsymbol{x}_k-\sigma\boldsymbol{u}_j)\|_2 \le \widetilde{\varepsilon}_{\boldsymbol{y}}.$$

Therefore,

$$\|\boldsymbol{g}_j(\boldsymbol{x}_k)-\widehat{\boldsymbol{g}}_j(\boldsymbol{x}_k)\|_2 \le \widetilde{\varepsilon}_{\boldsymbol{y}},$$

and hence

$$\|\boldsymbol{g}(\boldsymbol{x}_k)-\widehat{\boldsymbol{g}}(\boldsymbol{x}_k)\|_2 \le \frac{1}{b_g}\sum_{j=1}^{b_g}\|\boldsymbol{g}_j(\boldsymbol{x}_k)-\widehat{\boldsymbol{g}}_j(\boldsymbol{x}_k)\|_2 \le \widetilde{\varepsilon}_{\boldsymbol{y}}.$$

It follows that

$$\|\boldsymbol{g}(\boldsymbol{x}_k)-\nabla P_\sigma(\boldsymbol{x}_k)\|_2 \le \widetilde{\varepsilon}_{\boldsymbol{y}} + \|\widehat{\boldsymbol{g}}(\boldsymbol{x}_k)-\nabla P_\sigma(\boldsymbol{x}_k)\|_2.$$

Thus, for any $t > \widetilde{\varepsilon}_{\boldsymbol{y}}$,

$$\mathbb{P}\left\{\|\boldsymbol{g}(\boldsymbol{x}_k)-\nabla P_\sigma(\boldsymbol{x}_k)\|_2 \ge t\right\} \le \mathbb{P}\left\{\|\widehat{\boldsymbol{g}}(\boldsymbol{x}_k)-\nabla P_\sigma(\boldsymbol{x}_k)\|_2 \ge t-\widetilde{\varepsilon}_{\boldsymbol{y}}\right\}.$$

Since P is $(\kappa+1)\ell$-smooth, applying Lemma A.1 with $L=(\kappa+1)\ell$ and t replaced by $t-\widetilde{\varepsilon}_{\boldsymbol{y}}$ yields the desired result. □

D.2 Proof of Lemma 4.5

Proof. Define

$$\widehat{\boldsymbol{H}}(\boldsymbol{x}_k) := \frac{1}{b_H}\sum_{j=1}^{b_H}\mathrm{Sym}\left(\frac{n}{2\sigma}\left(\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j)-\nabla P(\boldsymbol{x}_k-\sigma\boldsymbol{v}_j)\right)\boldsymbol{v}_j^\top\right).$$

By Proposition 3.4 of (Guan & So, 2026a), $\widehat{\boldsymbol{H}}(\boldsymbol{x}_k)$ is an unbiased estimator of $\nabla^2 P_\sigma(\boldsymbol{x}_k)$. For brevity, write $\boldsymbol{H}(\boldsymbol{x}_k) = {b_H}^{-1}\sum_{j=1}^{b_H}\boldsymbol{H}_j(\boldsymbol{x}_k)$ and $\widehat{\boldsymbol{H}}(\boldsymbol{x}_k) = {b_H}^{-1}\sum_{j=1}^{b_H}\widehat{\boldsymbol{H}}_j(\boldsymbol{x}_k)$. Using $\|\mathrm{Sym}(\boldsymbol{A})\|_2 \le \|\boldsymbol{A}\|_2$ and $\|\boldsymbol{v}_j\|_2 = 1$, we have

$$\begin{aligned}\|\boldsymbol{H}_j(\boldsymbol{x}_k)-\widehat{\boldsymbol{H}}_j(\boldsymbol{x}_k)\|_2 \le \frac{n}{2\sigma}\big(&\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j,\widehat{\boldsymbol{y}}^+_{k_j})-\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j)\|_2\\ &+\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k-\sigma\boldsymbol{v}_j,\widehat{\boldsymbol{y}}^-_{k_j})-\nabla P(\boldsymbol{x}_k-\sigma\boldsymbol{v}_j)\|_2\big).\end{aligned}$$

Since $\nabla P(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{y}^\star(\boldsymbol{x}))$, Assumption 3.1 and Lemma 4.3 imply

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j,\widehat{\boldsymbol{y}}^+_{k_j})-\nabla P(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j)\|_2 \le \ell\|\widehat{\boldsymbol{y}}^+_{k_j}-\boldsymbol{y}^\star(\boldsymbol{x}_k+\sigma\boldsymbol{v}_j)\|_2 \le \frac{\sigma\widehat{\varepsilon}_{\boldsymbol{y}}}{n}.$$

Similarly,

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k - \sigma \boldsymbol{v}_j, \widehat{\boldsymbol{y}}_{k_j}^-) - \nabla P(\boldsymbol{x}_k - \sigma \boldsymbol{v}_j)\|_2 \le \frac{\sigma \widehat{\varepsilon}_{\boldsymbol{y}}}{n}.$$

Hence,

$$\|\boldsymbol{H}_j(\boldsymbol{x}_k) - \widehat{\boldsymbol{H}}_j(\boldsymbol{x}_k)\|_2 \le \widehat{\varepsilon}_{\boldsymbol{y}},$$

and therefore

$$\|\boldsymbol{H}(\boldsymbol{x}_k) - \widehat{\boldsymbol{H}}(\boldsymbol{x}_k)\|_2 \le \frac{1}{b_H} \sum_{j=1}^{b_H} \|\boldsymbol{H}_j(\boldsymbol{x}_k) - \widehat{\boldsymbol{H}}_j(\boldsymbol{x}_k)\|_2 \le \widehat{\varepsilon}_{\boldsymbol{y}}.$$

It follows that

$$\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \le \widehat{\varepsilon}_{\boldsymbol{y}} + \|\widehat{\boldsymbol{H}}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2.$$

Thus, for any $t > \widehat{\varepsilon}_{\boldsymbol{y}}$,

$$\mathbb{P}\left\{\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \ge t\right\} \le \mathbb{P}\left\{\|\widehat{\boldsymbol{H}}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 \ge t - \widehat{\varepsilon}_{\boldsymbol{y}}\right\}.$$

Since P is $(\kappa+1)\ell$-smooth, Proposition 3.4 of (Guan & So, 2026a), applied with $L = (\kappa+1)\ell$ and t replaced by $t - \widehat{\varepsilon}_{\boldsymbol{y}}$, yields the desired result. □

D.3 Proof of Lemma 4.6

Proof. According to Lemmas 4.4 and 4.5 and the definitions of $K_\star$, $\widetilde{\varepsilon}_{\boldsymbol{y}}$ and $\widehat{\varepsilon}_{\boldsymbol{y}}$, we have

$$\begin{aligned}\mathbb{P}\left\{\|\boldsymbol{g}(\boldsymbol{x}_k) - \nabla P_\sigma(\boldsymbol{x}_k)\|_2 > \frac{Mr_\star^2}{48}\right\} &\le 2\exp\left(-\frac{b_g M^2 r_\star^4/2916}{2(\kappa+1)^2\ell^2\sigma^2 + 4(\kappa+1)\ell\sigma M r_\star^2/162}\right) \\ &\le \frac{\eta}{2K_\star},\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}\left\{\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 > \frac{Mr_\star}{24}\right\} &\le 2n\exp\left(-\frac{b_H M^2 r_\star^2/1458}{C_{n\kappa\ell}^2 + C_{n\kappa\ell} M r_\star/81}\right) \\ &\le \frac{\eta}{2K_\star},\end{aligned}$$

where the two inequalities use the definitions of b_g and b_H, respectively. Then, the complementary event occurs with probability at most

$$\begin{aligned}\mathbb{P}\left\{\mathcal{E}_{K_\star}^c\right\} &\le \sum_{k=0}^{K_\star-1} \mathbb{P}\left\{\|\boldsymbol{g}(\boldsymbol{x}_k) - \nabla P_\sigma(\boldsymbol{x}_k)\|_2 > \frac{Mr_\star^2}{48}\right\} \\ &\quad + \sum_{k=0}^{K_\star-1} \mathbb{P}\left\{\|\boldsymbol{H}(\boldsymbol{x}_k) - \nabla^2 P_\sigma(\boldsymbol{x}_k)\|_2 > \frac{Mr_\star}{24}\right\} \\ &\le \eta.\end{aligned}$$

Consequently, we have $\mathbb{P}\{\mathcal{E}_{K_\star}\} \ge 1 - \eta$. □

E Iteration and Oracle Complexity

Proof. When $\varepsilon_1, \varepsilon_2, \delta \asymp \varepsilon$, we have

$$\sigma = \min\left\{\delta, \frac{\varepsilon_1}{3(\kappa+1)\ell}\right\} = \Theta\left(\frac{\varepsilon}{\kappa\ell}\right),$$

and hence

$$M = \frac{c(\kappa+1)\ell\sqrt{n}}{\sigma} = \mathcal{O}\left(\frac{\kappa^2\ell^2\sqrt{n}}{\varepsilon}\right).$$

Moreover, from the definition of $r_\star$,

$$r_\star = \min\left\{\sqrt{\frac{16\varepsilon_g}{17M}}, \frac{24\varepsilon_H}{37M}\right\} = \Theta\left(\frac{\varepsilon^2}{\kappa^2\ell^2\sqrt{n}}\right),$$

for sufficiently small ε. Substituting these estimates into the definition of $K_\star$ yields

$$K_\star = \left\lceil \frac{24}{Mr_\star^3}\left(P_\sigma(\boldsymbol{x}_0) - \min_{\boldsymbol{x}\in\mathbb{R}^n} P_\sigma(\boldsymbol{x})\right)\right\rceil = \widetilde{\mathcal{O}}\left(\kappa^4\ell^4 n\varepsilon^{-5}\right).$$

The total first-order oracle complexity is obtained by combining the number of outer iterations, the sample sizes used in the gradient and Hessian estimators, and the corresponding numbers of inner AGD iterations. Using the parameter choices above and suppressing logarithmic factors, we have

$$b_g = \widetilde{\mathcal{O}}\left(\kappa^4\ell^4 n\varepsilon^{-4}\right), \qquad b_H = \widetilde{\mathcal{O}}\left(\kappa^2\ell^2 n^2\varepsilon^{-2}\right).$$

Moreover, Lemmas 4.1, 4.2, and 4.3 imply that

$$T_k,\ T_g,\ T_H = \widetilde{\mathcal{O}}(\sqrt{\kappa}).$$

Therefore, the total number of first-order oracle calls satisfies

$$T = K_\star\left(T_k + 2b_gT_g + 2b_HT_H\right) = \widetilde{\mathcal{O}}\left(\kappa^{8.5}\ell^8 n^2\varepsilon^{-9} + \kappa^{6.5}\ell^6 n^3\varepsilon^{-7}\right).$$

□

F Experimental Details and Additional Plots

All the code is written in MATLAB R2023b, and all numerical experiments are conducted on a PC with an Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz and 16GB of RAM.

F.1 Definition of the Synthetic Function

The scalar function ω used in Section 5 is defined by

$$\omega(\boldsymbol{x}) = \begin{cases} H - \dfrac{\gamma}{2}x^2 + \dfrac{\gamma}{12\xi}|\boldsymbol{x}|^3, & |\boldsymbol{x}| \le \xi, \\ H - \dfrac{3}{4}\gamma\xi|\boldsymbol{x}| + \dfrac{1}{3}\gamma\xi^2, & \xi < |\boldsymbol{x}| \le 2\xi, \\ a_3(|\boldsymbol{x}| - 2\xi)^3 + a_2(|\boldsymbol{x}| - 2\xi)^2 - \dfrac{3}{4}\gamma\xi(|\boldsymbol{x}| - 2\xi) + H - \dfrac{7}{6}\gamma\xi^2, & 2\xi < |\boldsymbol{x}| \le 3\xi, \\ -\dfrac{(4\xi - |\boldsymbol{x}|)^2}{\alpha^2}, & 3\xi < |\boldsymbol{x}| \le 4\xi, \\ 0, & |\boldsymbol{x}| > 4\xi, \end{cases}$$

where

$$a_3 = \frac{2H}{\xi^3} + \frac{4}{\alpha^2\xi} - \frac{37\gamma}{12\xi}, \qquad a_2 = 5\gamma - \frac{5}{\alpha^2} - \frac{3H}{\xi^2}.$$

We set $H = 10^{-4}, \gamma = 0.1, \xi = 0.2$ and $\alpha = 20$ in our numerical experiments. Moreover, ∇f is globally 8-Lipschitz continuous and f is 0.05-strongly concave with respect to $\boldsymbol{y}$ and thus $\kappa = \ell/\mu = 160$. Thus, we know that $\boldsymbol{y}^\star(\boldsymbol{x}) = [20\boldsymbol{x}_1, 0.2\boldsymbol{x}_2]^\top$ and

$$P(\boldsymbol{x}) = \omega(\boldsymbol{x}_3) + 10\boldsymbol{x}_1^2 + 0.1\boldsymbol{x}_2^2.$$

The nonconvexity of the problem mainly arises from $\omega(\boldsymbol{x}_3)$. For the above parameter setting, we plot $\omega(\boldsymbol{x})$ and its gradient function in Figure 4. Obviously, ω has two strict local minimizers

$$\boldsymbol{x}_{\min}^{\mathrm{loc}} = \pm\frac{112 - \sqrt{411}}{220} \approx \pm 0.41694.$$

The corresponding local minimum value of P is $P^\star = \omega(\boldsymbol{x}_{\min}^{\mathrm{loc}})$. Moreover, $\boldsymbol{x}_3 = 0$ is a strict local maximizer with $\omega(0) = H$, whereas $|\boldsymbol{x}_3| \ge 4\xi$ corresponds to nonstrict local maximizers due to the flat outer region. Hence,

$$\left([0,0,0]^\top, [0,0]^\top\right),\ \left([0,0,-0.8]^\top, [0,0]^\top\right) \text{ and } \left([0,0,0.8]^\top, [0,0]^\top\right)$$

are stationary points of f, while the local minimax solutions are $([0,0,\boldsymbol{x}_{\min}^{\mathrm{loc}}]^\top, [0,0]^\top)$.

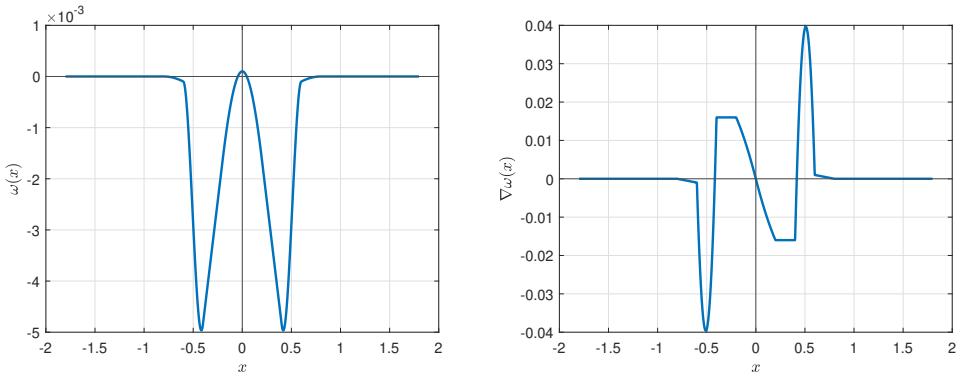

Figure 4: The function $\omega(\boldsymbol{x})$ and the gradient function $\nabla\omega(\boldsymbol{x})$.

F.2 Algorithm Parameters

In the experiments, we choose the step sizes of GDA according to the two-time-scale GDA in (Lin et al., 2020) and set $\eta_x = 2 \times 10^{-5}$ and $\eta_y = 0.2$. For SMCN, we set $M = 10$, $\sigma = 0.1$, and $b_g = b_H = 100$. AGD is terminated after a fixed number of iterations. Specifically, we use $T = 300$ iterations for approximately solving the maximization problem at the current outer iterate, and set $T_g = T_H = 50$ for the maximization problems arising in the gradient and Hessian approximations.