Finding Second-Order Stationary Points in Nonconvex-Strongly-Concave Minimax Optimization

Luo Luo
School of Data Science
Fudan University
luoluo@fudan.edu.cn

Yujun Li
Noah's Ark Lab
Huawei Technologies Co., Ltd.
liyujun9@huawei.com

Cheng Chen*
School of Physical and Mathematical Sciences
Nanyang Technological University
cheng.chen@ntu.edu.sg

Abstract

We study the smooth minimax optimization problem $\min_{\mathbf{x}} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$, where f is ℓ-smooth, strongly-concave in $\mathbf{y}$ but possibly nonconvex in $\mathbf{x}$. Most of existing works focus on finding the first-order stationary points of the function $f(\mathbf{x}, \mathbf{y})$ or its primal function $P(\mathbf{x}) \triangleq \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$, but few of them focus on achieving second-order stationary points. In this paper, we propose a novel approach for minimax optimization, called Minimax Cubic Newton (MCN), which could find an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-second-order stationary point of $P(\mathbf{x})$ with calling $\mathcal{O}(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5})$ times of second-order oracles and $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$ times of first-order oracles, where κ is the condition number and ρ is the Lipschitz continuous constant for the Hessian of $f(\mathbf{x}, \mathbf{y})$. In addition, we propose an inexact variant of MCN for high-dimensional problems to avoid calling expensive second-order oracles. Instead, our method solves the cubic sub-problem inexactly via gradient descent and matrix Chebyshev expansion. This strategy still obtains the desired approximate second-order stationary point with high probability but only requires $\tilde{\mathcal{O}}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right)$ Hessian-vector oracle calls and $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$ first-order oracle calls. To the best of our knowledge, this is the first work that considers the non-asymptotic convergence behavior of finding second-order stationary points for minimax problems without the convex-concave assumptions.

1 Introduction

We consider minimax optimization of the form

$$\min_{\mathbf{x}\in\mathbb{R}^{d_x}} \max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $f(\mathbf{x}, \mathbf{y})$ is ℓ-smooth, μ-strongly-concave in $\mathbf{y}$, but possibly nonconvex in $\mathbf{x}$. Problem (1) can also be written as

$$\min_{\mathbf{x}\in\mathbb{R}^{d_x}} \left\{ P(\mathbf{x}) \triangleq \max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x}, \mathbf{y}) \right\}. \tag{2}$$

This framework covers a wide range of applications in machine learning such as regularized GAN [30], reinforcement learning [29], domain adaptation [9] and adversarial training [33].

*The corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Most recent works focus on finding an ε-first-order stationary point (FSP) of $P(\mathbf{x})$. Lin et al. [20] showed that the vanilla gradient descent ascent (GDA) method could obtain an ε-FSP with $\mathcal{O}((\kappa^2\ell + \kappa\ell^2)\varepsilon^{-2})$ first-order oracle calls. This complexity can be reduced to $\tilde{\mathcal{O}}\left(\sqrt{\kappa}\ell\varepsilon^{-2}\right)$ by proximal iteration algorithms [21], which matches the gradient oracle lower bound for finding ε-FSP of $P(\mathbf{x})$ [13, 42]. The theory of first-order optimization for problem (1) has also been studied in stochastic settings [12, 15, 20, 22, 39, 40]. However, the approximate FSP obtained by these algorithms cannot guarantee the local optimality since the primal function $P(\mathbf{x})$ could be nonconvex.

In this paper, we focus on finding a second-order stationary point (SSP) of $P(\mathbf{x})$ to capture the local optimal properties [8, 24]. Inspired by the success of second-order optimization in nonconvex minimization [1, 5, 6, 14, 18, 26, 36, 43], we propose a novel method, called Minimax Cubic Newton (MCN), which runs cubic Newton update on $\mathbf{x}$ and maximizes the objective on $\mathbf{y}$ alternatively. This iteration scheme avoids getting stuck at an unexpected FSP. Specifically, we show MCN will converge to an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-SSP of $P(\mathbf{x})$ with $\mathcal{O}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)$ number of iterations, where κ is the condition number and ρ is the Lipschitz continuous constant of $\nabla^2 f(\mathbf{x}, \mathbf{y})$. For high-dimensional problems, we also propose an efficient algorithm, called Inexact Minimax Cubic Newton (IMCN), which avoids the expensive second-order oracle calls. IMCN approximates the second-order information by matrix Chebyshev polynomial and solves the cubic regularized sub-problem inexactly. It only requires $\tilde{\mathcal{O}}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right)$ Hessian-vector oracle calls and $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$ first-order oracle calls to find an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-SSP. Under mild strict saddle condition [3, 10, 11, 34, 35], the approximate SSP of $P(\mathbf{x})$ implies an approximate local minimax point of $f(\mathbf{x}, \mathbf{y})$ defined by Jin et al. [17], which successfully characterizes the local optimality for problem (1). To the best of our knowledge, this is the first work that considers non-asymptotic convergence behavior of finding SSP for minimax problems without convex-concave assumptions. We also conduct experiments on both synthetic function and the real application to validate our theoretical analysis. The empirical results show that the proposed algorithms significantly outperform the GDA method.

In a concurrent work, Chen et al. [7] also studied Problem (1) and proposed Cubic-GDA which is similar to our MCN algorithm. MCN has advantage on complexity of first-order oracles by a factor of $\sqrt{\kappa}$ since Cubic-GDA adopts GD to update $\mathbf{y}$ while MCN uses AGD instead. Chen et al. [7] mentioned that the cubic sub-problem can be efficiently solved by gradient-based algorithms, but they did not provide theoretical analysis for this inexact variant, which is more practical in high dimensional case. As a comparison, we provide the complexity of both Hessian-vector oracles and first-order oracles of our inexact algorithm IMCN.

2 Preliminaries

This section first presents the notations and assumptions for our settings. Then we introduce the background of local optimality for minimax optimization and some basic algorithms.

2.1 Notations and Assumptions

For a twice differentiable function $f(\mathbf{x}, \mathbf{y})$, its partial gradients with respect to $\mathbf{x}$ and $\mathbf{y}$ are denoted as $\nabla_x f(\mathbf{x}, \mathbf{y})$ and $\nabla_y f(\mathbf{x}, \mathbf{y})$ respectively. Its Hessian matrix at point $(\mathbf{x}, \mathbf{y})$ can be partitioned as $\nabla^2 f(\mathbf{x}, \mathbf{y}) = [\nabla^2_{xx} f(\mathbf{x}, \mathbf{y}), f(\mathbf{x}, \mathbf{y}); \nabla^2_{yx} f(\mathbf{x}, \mathbf{y}), \nabla^2_{yy} f(\mathbf{x}, \mathbf{y})]$, where $\nabla^2_{xx} f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_x \times d_x}$, $\nabla^2_{xy} f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_x \times d_y}$, $\nabla^2_{yx} f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_y \times d_x}$ and $\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_y \times d_y}$. We also denote $\mathbf{H}(\mathbf{x}, \mathbf{y}) = \nabla^2_{xx} f(\mathbf{x}, \mathbf{y}) - \nabla^2_{xy} f(\mathbf{x}, \mathbf{y})(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}))^{-1} \nabla^2_{yx} f(\mathbf{x}, \mathbf{y})$ if $\nabla^2_{yy} f(\mathbf{x}, \mathbf{y})$ is invertible.

Given a symmetric matrix $\mathbf{A}$, we denote $\lambda_{\min}(\mathbf{A})$ as the smallest eigenvalue of $\mathbf{A}$. We use $\|\cdot\|_2$ to denote the spectral norm of matrices and Euclidean norm of vectors. We also denote the closed Euclidean ball with radius r and center $\mathbf{x}^*$ as $\mathcal{B}(\mathbf{x}^*, r) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}^*\|_2 \leq r\}$. Additionally, we use notation $\tilde{\mathcal{O}}(\cdot)$ to hide logarithmic terms in the complexity.

We suppose the objective function $f(\mathbf{x}, \mathbf{y})$ of Problem (1) satisfies the following assumptions.

Assumption 1. *The function $f(\mathbf{x}, \mathbf{y})$ has ℓ-Lipschitz gradients, i.e., there exists a constant $\ell > 0$ such that $\|\nabla f(\mathbf{x}, \mathbf{y}) - \nabla f(\mathbf{x}', \mathbf{y}')\|_2^2 \leq \ell^2\left(\|\mathbf{x} - \mathbf{x}'\|_2^2 + \|\mathbf{y} - \mathbf{y}'\|_2^2\right)$ for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{d_x}$ and $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^{d_y}$.*

Assumption 2. *The function $f(\mathbf{x},\mathbf{y})$ has ρ-Lipschitz Hessian, i.e., there exists a constant $\rho > 0$ such that $\left\|\nabla^2 f(\mathbf{x},\mathbf{y}) - \nabla^2 f(\mathbf{x}',\mathbf{y}')\right\|_2^2 \leq \rho^2\left(\|\mathbf{x}-\mathbf{x}'\|_2^2 + \|\mathbf{y}-\mathbf{y}'\|_2^2\right)$ for any $\mathbf{x},\mathbf{x}' \in \mathbb{R}^{d_x}$ and $\mathbf{y},\mathbf{y}' \in \mathbb{R}^{d_y}$.*

Assumption 3. *The function $f(\mathbf{x},\mathbf{y})$ is μ-strongly-concave in $\mathbf{y}$, i.e., there exists a constant $\mu > 0$ such that $f(\mathbf{x},\mathbf{y}) \leq f(\mathbf{x},\mathbf{y}') + \nabla_y f(\mathbf{x},\mathbf{y})^\top(\mathbf{y}-\mathbf{y}') - \frac{\mu}{2}\|\mathbf{y}-\mathbf{y}'\|_2^2$ for any $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y},\mathbf{y}' \in \mathbb{R}^{d_y}$.*

Assumption 4. *The function $P(\mathbf{x}) \triangleq \max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x},\mathbf{y})$ satisfies $P^* \triangleq \inf_{\mathbf{x}\in\mathbb{R}^{d_x}} P(\mathbf{x}) > -\infty$.*

Definition 1. *Under Assumption 1 and 3, we define the condition number of $f(\mathbf{x},\mathbf{y})$ as $\kappa \triangleq \ell/\mu$.*

The assumptions of Lipschitz continuous gradient and strongly-concavity on f indicate that the primal function $P(\mathbf{x}) \triangleq \max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x},\mathbf{y})$ is well-defined and has Lipschitz continuous gradients as shown in Lemma 1.

Lemma 1 ([20, Lemma 4.3])**.** *Suppose the objective function f satisfies Assumptions 1 and 3, then the primal function $P(\mathbf{x}) \triangleq \max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x},\mathbf{y})$ has $(\kappa+1)\ell$-Lipschitz continuous gradients. Additionally, the function $\mathbf{y}^*(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathbb{R}^{d_y}} f(\mathbf{x},\mathbf{y})$ is well-defined and κ-Lipschitz. We also have $\nabla P(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y}^*(\mathbf{x}))$.*

Now we give the definitions of ε-FSP and (ε,δ)-SSP as follows.

Definition 2. *Suppose the function $f(\mathbf{x},\mathbf{y})$ satisfies Assumption 1 and 3, then we call $\mathbf{x}$ an ε-FSP of $P(\mathbf{x})$ if $\|\nabla P(\mathbf{x})\|_2 \leq \varepsilon$.*

Definition 3. *Suppose the function $f(\mathbf{x},\mathbf{y})$ satisfies Assumption 1, 2 and 3, then we call $\mathbf{x}$ is an (ε,δ)-SSP of $P(\mathbf{x})$ if $\|\nabla P(\mathbf{x})\|_2 \leq \varepsilon$ and $\nabla^2 P(\mathbf{x}) \succeq -\delta\mathbf{I}$.*

The following two lemmas provide the closed form of $\nabla^2 P(\mathbf{x})$ and its Lipschitz continuity.

Lemma 2 ([32])**.** *Suppose the function $f(\mathbf{x},\mathbf{y})$ satisfies Assumption 1, 2 and 3. We use the definition of $\mathbf{y}^*(\mathbf{x})$ in Lemma 1, then it holds that $\nabla^2 P(\mathbf{x}) = \mathbf{H}(\mathbf{x},\mathbf{y}^*(\mathbf{x}))$.*

Lemma 3. *Under assumptions of Lemma 2, we have $\left\|\nabla^2 P(\mathbf{x}) - \nabla^2 P(\mathbf{x}')\right\|_2 \leq 4\sqrt{2}\kappa^3\rho\|\mathbf{x}-\mathbf{x}'\|_2$ for any $\mathbf{x}$ and $\mathbf{x}'$ in $\mathbb{R}^{d_x}$.*

2.2 Local Optimality of Minimax Optimization

The Nash equilibrium is widely used in the study of convex-concave minimax optimization [38, 41], but it is intractable in general when the objective function $f(\mathbf{x},\mathbf{y})$ is nonconvex in $\mathbf{x}$ or nonconcave in $\mathbf{y}$. For the general minimax problem, we introduce the local minimax point [17], which characterizes the optimality in two-player sequential games where players can only change their strategies locally.

Definition 4 ([17])**.** *Given a differentiable function $f(\mathbf{x},\mathbf{y}) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ that is strongly-concave in $\mathbf{y}$, a point $(\mathbf{x}^*,\mathbf{y}^*) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ is called a local minimax point of f, if there exists $\delta_0 > 0$ and a function h satisfying $h(\delta) \to 0$ as $\delta \to 0$, such that for any $\delta \in (0,\delta_0]$, $\mathbf{x} \in \mathcal{B}(\mathbf{x}^*, h(\delta))$ and $\mathbf{y} \in \mathbb{R}^{d_y}$, we have*

$$f(\mathbf{x}^*,\mathbf{y}) \leq f(\mathbf{x}^*,\mathbf{y}^*) \leq \max_{\mathbf{y}'\in\mathbb{R}^{d_y}} f(\mathbf{x},\mathbf{y}'). \tag{3}$$

Remark 1. *The definition of local minimax point for general nonconvex-nonconcave [17] only requires (3) holds for any $\mathbf{y}$ and $\mathbf{y}'$ in a neighbour of $\mathbf{y}^*$, while the constraint on $\mathbf{y}$ and $\mathbf{y}'$ is unnecessary in our setting since we assume that $f(\mathbf{x},\mathbf{y})$ is strongly-concave in $\mathbf{y}$.*

The local minimax point enjoys the following property.

Lemma 4 ([17, Proposition 19])**.** *Suppose $f(\mathbf{x},\mathbf{y})$ is twice differentiable, then any point $(\mathbf{x}^*,\mathbf{y}^*)$ satisfying $\nabla f(\mathbf{x}^*,\mathbf{y}^*) = \mathbf{0}$, $\nabla^2_{yy} f(\mathbf{x}^*,\mathbf{y}^*) \prec \mathbf{0}$ and $\mathbf{H}(\mathbf{x}^*,\mathbf{y}^*) \succ 0$ is a local minimax point of $f(\mathbf{x},\mathbf{y})$.*

Based on Lemma 4, we introduce the strict-minimax condition on f which is an extension of strict-saddle condition for nonconvex minimization [3, 10, 11, 16, 34, 35].

Definition 5. *Under Assumption 1, 2 and 3, we say $f(\mathbf{x},\mathbf{y})$ is (α,β,γ)-strict-minimax for some $\alpha>0$, $\beta>0$ and $\gamma>0$ if every $(\hat{\mathbf{x}},\hat{\mathbf{y}})\in\mathbb{R}^{d_x}\times\mathbb{R}^{d_y}$ satisfies at least one of the following three conditions: (a) $\|\nabla f(\hat{\mathbf{x}},\hat{\mathbf{y}})\|_2>\alpha$; (b) $\lambda_{\min}(\mathbf{H}(\hat{\mathbf{x}},\hat{\mathbf{y}}))<-\beta$; (c) There exists a local minimax point $(\mathbf{x}^*,\mathbf{y}^*)$ such that $\|\hat{\mathbf{x}}-\mathbf{x}^*\|_2^2+\|\hat{\mathbf{y}}-\mathbf{y}^*\|_2^2\le\gamma^2$.*

Note that if f is (α,β,γ)-strict-minimax and there exists a point $\hat{\mathbf{x}}$ which is an (ε,δ)-SSP of $P(\mathbf{x})$ with sufficient small ε and δ, then we can find $\hat{\mathbf{y}}\approx\arg\max_{\mathbf{y}} f(\hat{\mathbf{x}},\mathbf{y})$ via running a first-order algorithm to minimize $-f(\hat{\mathbf{x}},\cdot)$ and obtain $(\hat{\mathbf{x}},\hat{\mathbf{y}})$ which is in a neighborhood of $(\mathbf{x}^*,\mathbf{y}^*)$. In other words, under the strict-minimax condition, we can reduce the task of finding approximate local minimax point of $f(\mathbf{x},\mathbf{y})$ to finding an approximate SSP of $P(\mathbf{x})$. We will provide the formal statement in Section 3.2.

2.3 Accelerated Gradient Descent

Nesterov's accelerated gradient descent (AGD) is the optimal first-order algorithm for convex optimization [25, 27], which is widely used in minimax optimization algorithms [21, 37]. We describe the details of AGD for smooth and strongly-convex functions in Algorithm 1, which has the following convergence rate.

Algorithm 1 $\mathrm{AGD}(h,\mathbf{y}_0,K,\eta,\theta)$

$\tilde{\mathbf{y}}_0=\mathbf{y}_0$
for $k=0,\dots,K-1$ **do**
  $\mathbf{y}_{k+1}=\tilde{\mathbf{y}}_k-\eta\nabla h(\tilde{\mathbf{y}}_k)$
  $\tilde{\mathbf{y}}_{k+1}=\mathbf{y}_{t+1}+\theta(\mathbf{y}_{k+1}-\mathbf{y}_k)$
end for
Output: $\mathbf{y}_K$

Lemma 5 ([37, Lemma 2])**.** *Running Algorithm 1 on a ℓ_h-smooth and μ_h-strongly-convex objective function $h(\cdot)$ with parameters $\eta=1/\ell_h$ and $\theta=\frac{\sqrt{\kappa_h}-1}{\sqrt{\kappa_h}+1}$ produces the output $\mathbf{y}_K$ satisfying $\|\mathbf{y}_K-\mathbf{y}^*\|_2^2\le(\kappa_h+1)\big(1-\frac{1}{\sqrt{\kappa_h}}\big)^K\|\mathbf{y}_0-\mathbf{y}^*\|_2^2$, where $\mathbf{y}^*=\arg\min_y h(\mathbf{y})$ and $\kappa_h=\ell_h/\mu_h$.*

2.4 Cubic Regularized Newton

Cubic regularized Newton (CRN) is a classic algorithm for nonconvex minimization [5, 6, 25, 26, 36]. It solves the nonconvex minimization problem $\min_{\mathbf{x}} g(\mathbf{x})$ via the following update rules

$$\mathbf{s}_t=\arg\min_{\mathbf{s}\in\mathbb{R}^d}\nabla g(\mathbf{x}_t)^\top\mathbf{s}+\frac{1}{2}\mathbf{s}^\top\nabla^2 g(\mathbf{x}_t)\mathbf{s}+\frac{\rho_g}{6}\|\mathbf{s}\|_2^3,\qquad \mathbf{x}_{t+1}=\mathbf{x}_t+\mathbf{s}_t.$$

The CRN method could find an $(\varepsilon,\sqrt{\rho_g\varepsilon})$-SSP of $g(\mathbf{x})$ with $\mathcal{O}(\sqrt{\rho_g}\varepsilon^{-1.5})$ number of iterations, where ρ_g is the Lipschitz continuous constant of the Hessian of $g(\mathbf{x})$.

3 Minimax Cubic Newton Algorithm

In this section, we propose our minimax cubic Newton algorithm and give its convergence results.

3.1 Minimax Cubic Newton Method

We present the details of Minimax Cubic Newton (MCN) method in Algorithm 2. In each round, the MCN algorithm performs following steps:

- Run AGD as presented in Algorithm 1 to estimate $\mathbf{y}_t\approx\mathbf{y}^*(\mathbf{x}_t)=\arg\max_{\mathbf{y}} f(\mathbf{x}_t,\mathbf{y})$.
- Compute the inexact first-order and second-order information of P at $\mathbf{x}_t$ as $\nabla P(\mathbf{x}_t)\approx\mathbf{g}_t=\nabla_x f(\mathbf{x}_t,\mathbf{y}_t)$ and $\nabla^2 P(\mathbf{x}_t)\approx\mathbf{H}_t=\mathbf{H}(\mathbf{x}_t,\mathbf{y}_t)$.
- Solve the following cubic regularized problem

$$\mathbf{s}_t^*=\arg\min_{\mathbf{s}\in\mathbb{R}^{d_x}}\left(\mathbf{g}_t^\top\mathbf{s}+\frac{1}{2}\mathbf{s}^\top\mathbf{H}_t\mathbf{s}+\frac{M}{6}\|\mathbf{s}\|_2^3\right).\tag{4}$$

The expressions of $\mathbf{g}_t$ and $\mathbf{H}_t$ in the algorithm are inspired from Lemma 1 and 3. The smoothness of $\nabla P(\mathbf{x})$ and $\nabla^2 P(\mathbf{x})$ allow the total complexity of AGD iteration of the algorithm has the desired upper bound. Our theoretical analysis show the termination condition in Line 7 of Algorithm 2 can be attained in no more than $\mathcal{O}(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5})$ number of iterations. We also show that a small $\|\mathbf{s}_t^*\|_2$ will lead to a desired approximate second-order stationary point of $P(\mathbf{x})$. Hence, using AGD to optimize $f(\mathbf{x}_t+\mathbf{s}_t^*,\mathbf{y})$ with respect to $\mathbf{y}$ generates an approximate local minimax point of $f(\mathbf{x},\mathbf{y})$.

3.2 Complexity Analysis for MCN

In this section, we let $M = 4\sqrt{2}\kappa^3\rho$ for MCN. Our analysis for MCN algorithm contains three parts:

1. We follow Tripuraneni et al. [36]'s idea to show that our algorithm with sufficient accurate gradient and Hessian estimator of $P(\mathbf{x})$ requires no more than $T = \mathcal{O}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)$ rounds of iterations to achieve an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-SSP of $P(\mathbf{x})$.
2. We prove the AGD step in line 3 requires at most $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$ gradient calls in total.
3. The last part shows we can achieve an approximate local minimax point of $f(\mathbf{x}, \mathbf{y})$ from an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-second-order stationary point of $P(\mathbf{x})$ under strict-saddle condition.

Algorithm 2 Minimax Cubic-Newton (MCN)

1: **Input:** $\mathbf{x}_0 \in \mathbb{R}^{d_x}$, $\mathbf{y}_{-1} = \mathbf{0}$, T, $\{K_t\}_{t=0}^T$, ε.
2: **for** $t = 0, \cdots T-1$ **do**
3: $\quad \mathbf{y}_t = \mathrm{AGD}\left(-f(\mathbf{x}_t, \cdot), \mathbf{y}_{t-1}, K_t, \frac{1}{\ell}, \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)$
4: $\quad \mathbf{g}_t = \nabla_x f(\mathbf{x}_t, \mathbf{y}_t)$, $\mathbf{H}_t = \mathbf{H}(\mathbf{x}_t, \mathbf{y}_t)$
5: $\quad \mathbf{s}_t^* = \arg\min_{\mathbf{s}\in\mathbb{R}^{d_x}} \left(\mathbf{g}_t^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_t \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|_2^3\right)$
6: $\quad$ **if** $\|\mathbf{s}_t^*\|_2 \le \frac{1}{2}\sqrt{\varepsilon/M}$ **then** break
7: $\quad \mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{s}_t^*$
8: **end for**
9: **Output:** $\hat{\mathbf{x}} = \mathbf{x}_t + \mathbf{s}_t^*$

Cubic Newton Iteration on $P(\mathbf{x})$ The procedure of our Algorithm 2 can be regarded as applying cubic Newton method to minimize nonconvex function $P(\mathbf{x})$, but using inexact first-order and second-order information. We consider the following conditions on the inexact gradient and Hessian, which will hold if we run AGD with enough number of iterations.

Assumption 5. *Suppose the estimators $\mathbf{g}_t \in \mathbb{R}^{d_x}$ and $\mathbf{H}_t \in \mathbb{R}^{d_x\times d_x}$ satisfy conditions $\|\nabla P(\mathbf{x}_t) - \mathbf{g}_t\|_2 \le C_g\varepsilon$ and $\left\|\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t\right\|_2 \le C_H\sqrt{M\varepsilon}$ for some $C_g > 0$ and $C_H > 0$.*

The following lemma implies the analysis of MCN algorithm only needs to focus on the case of each $\|\mathbf{s}_t^*\|_2$ is large, otherwise we have already find $\mathbf{x}_{t+1}$ which a desired approximate SSP of $P(\mathbf{x})$.

Lemma 6. *Under Assumption 5 with $C_g = 1/192$ and $C_H = 1/48$, if $\mathbf{x}_{t+1}$ from Algorithm 2 is not an $\left(\varepsilon, \sqrt{M\varepsilon}\right)$-SSP of $P(\mathbf{x})$, we have $\|\mathbf{s}_t^*\|_2 \ge \frac{1}{2}\sqrt{\varepsilon/M}$.*

By Lemma 6, we can show that MCN with sufficient accurate gradient and Hessian estimators can find an $\left(\varepsilon, \sqrt{M\varepsilon}\right)$-SSP of $P(\mathbf{x})$ with $T = \mathcal{O}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)$ number of iterations as follows.

Theorem 1. *Under Assumption 1-4, we run Algorithm 2 with $T = \left\lceil 192(P(\mathbf{x}_0) - P^*)\sqrt{M}\varepsilon^{-1.5}\right\rceil + 1$; and further suppose K_t is sufficient large so that results $\mathbf{g}_t$ and $\mathbf{H}_t$ satisfy Assumption 5 with $C_g = 1/192$ and $C_H = 1/48$. Then the output $\hat{\mathbf{x}}$ is an $\left(\varepsilon, \sqrt{M\varepsilon}\right)$-SSP of $P(\mathbf{x})$.*

Total Complexity of AGD Note that MCN (Algorithm 2) applies AGD to maximize $f(\mathbf{x}_t, \cdot)$ by using $\mathbf{y}_{t-1}$ as initialization at the t-th round. With such initialization, the following theorem provides the upper bound of total number of gradient calls required by AGD and guarantees that $\mathbf{g}_t$ and $\mathbf{H}_t$ in the MCN algorithm satisfy Assumption 5.

Theorem 2. *Under Assumption 1-4, we run Algorithm 2 with $K_0 = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\|\mathbf{y}^*(\mathbf{x}_0)\|_2\right)\right\rceil$ and $K_t = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left(\tilde{\varepsilon} + \kappa\left\|\mathbf{s}_{t-1}^*\right\|_2\right)\right)\right\rceil$ for $t \ge 1$, where $\tilde{\varepsilon} = \min\left\{C_g\varepsilon/\ell, C_H\sqrt{M\varepsilon}/\rho\right\}$ and $\mathbf{s}_0 = \mathbf{0}$. Then all $\mathbf{g}_t$ and $\mathbf{H}_t$ satisfy the condition of Assumption 5. We also have*

$$\sum_{t=0}^{T} K_t \le T+1+\frac{2\sqrt{\kappa}T}{3}\left[3\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\|\mathbf{y}^*(\mathbf{x}_0)\|_2\right)+\log\left(8(\kappa+1)^{1.5}+\frac{8\kappa^3(\kappa+1)^{1.5}}{T\tilde{\varepsilon}^3}\sum_{t=1}^{T}\|\mathbf{s}_{t-1}^*\|_2^3\right)\right].$$

Combining Theorem 1 and Theorem 2, we can obtain the total number of gradient oracle calls, Hessian (inverse) oracle calls and exact cubic sub-problem solver calls as follows.

Corollary 1. *Under Assumption 1-4, running Algorithm 2 with $T = \left\lceil 192(P(\mathbf{x}_0) - P^*)\sqrt{M}\varepsilon'^{-1.5}\right\rceil + 1$, $K_0 = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}'}\|\mathbf{y}^*(\mathbf{x}_0)\|_2\right)\right\rceil$ and $K_t = \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}'}\left(\tilde{\varepsilon}' + \kappa\|\mathbf{s}_{t-1}\|_2\right)\right)\right\rceil$ for $t \ge 1$, where $\tilde{\varepsilon}' = \min\left\{C_g\varepsilon'/\ell, C_H\sqrt{M\varepsilon'}/\rho\right\}$, $\mathbf{s}_0 = \mathbf{0}$ and $\varepsilon' = 2^{-2.5}\varepsilon$, then the output $\hat{\mathbf{x}}$ is an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-SSP*

of $P(\mathbf{x})$ and the number of gradient oracle calls is at most $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$. The total number of Hessian (inverse) oracle calls and exact cubic sub-problem solver calls is at most $\mathcal{O}\left(\sqrt{\rho}\kappa^{1.5}\varepsilon^{-1.5}\right)$.

Note that MCN method needs to construct the Hessian estimator $\mathbf{H}_t = \mathbf{H}(\mathbf{x}_t, \mathbf{y}_t)$ and solves the cubic regularized sub-problem (4) in each round. Constructing $\mathbf{H}_t$ requires calling the second-order oracle at $(\mathbf{x}_t, \mathbf{y}_t)$ and taking $\mathcal{O}\left(d_y^3 + d_x d_y^2 + d_x^2 d_y\right)$ flops for matrix multiplication and inversion. Solving sub-problem (4) requires $\mathcal{O}\left(d_x^3\right)$ flops for matrix factorization or inversion [5, 6]. Hence, besides the gradient calls from AGD step, MCN requires $\mathcal{O}\left(d_x^3 + d_y^3\right)$ time complexity in each round and its space complexity is $\mathcal{O}\left(d_x^2 + d_y^2\right)$.

Algorithm 3 Inexact Minimax Cubic-Newton

Input: $\mathbf{x}_0 \in \mathbb{R}^{d_x}$, $\mathbf{y}_{-1} = \mathbf{0}$, T, $\{K_t\}_{t=0}^T$, ε
$c_k = \frac{2}{\sqrt{\ell\mu}}\left(\frac{\sqrt{\mu/\ell}-1}{\sqrt{\mu/\ell}+1}\right)^k$
for $t = 0, \cdots$ **do**
  $\mathbf{y}_t = \mathrm{AGD}\big(-f(\mathbf{x}_t, \cdot), \mathbf{y}_{t-1}, K_t, \frac{1}{\ell}, \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)$
  $\mathbf{g}_t = \nabla_x f(\mathbf{x}_t, \mathbf{y}_t)$
  $\mathbf{Z}_t = -\frac{2}{\ell-\mu}\left(\nabla_{yy}^2 f(\mathbf{x}_t, \mathbf{y}_t) + \frac{\ell+\mu}{2}\mathbf{I}\right)$
  Compute $\mathbf{H}_t$ as equation (5)
  $(\mathbf{s}_t, \Delta_t)$=Cubic-Solver$(\mathbf{g}_t, \mathbf{H}_t, \sigma, \mathcal{K}(\varepsilon, \delta'))$
  $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{s}_t$
  if $\Delta_t > -\frac{1}{128}\sqrt{\varepsilon^3/M}$ **then**
    $\hat{\mathbf{s}} =$ Final-Cubic-Solver$(\mathbf{g}_t, \mathbf{H}_t, \varepsilon)$
    $\mathbf{x}_{t+1} = \mathbf{x}_t + \hat{\mathbf{s}}$
    break
  end if
end for
Output: $\hat{\mathbf{x}} = \mathbf{x}_{t+1}$

Algorithm 4 Cubic-Solver

Input: $\mathbf{g}$, $\mathbf{H}$, σ, $\mathcal{K}(\varepsilon, \delta')$
if $\|\mathbf{g}\|_2 \geq L^2/M$ **then**
  $R_C = -\frac{\mathbf{g}^\top\mathbf{H}\mathbf{g}}{M\|\mathbf{g}\|_2^2} + \sqrt{\left(\frac{\mathbf{g}^\top\mathbf{H}\mathbf{g}}{M\|\mathbf{g}\|_2^2}\right)^2 + \frac{2\|\mathbf{g}\|_2}{M}}$
  $\hat{\mathbf{s}} = -R_C\mathbf{g}/\|\mathbf{g}\|_2$
else
  $\mathbf{s}_0 = \mathbf{0}$, $\eta = 1/(20L)$
  $\tilde{\mathbf{g}} = \mathbf{g} + \sigma\boldsymbol{\zeta}$ for $\boldsymbol{\zeta} \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$
  for $k = 0, \cdots, \mathcal{K}(\varepsilon, \delta')$ **do**
    $\mathbf{s}_{k+1} = \mathbf{s}_k - \eta\left(\tilde{\mathbf{g}} + \mathbf{H}\mathbf{s}_k + \frac{M}{2}\|\mathbf{s}_k\|_2\mathbf{s}_k\right)$
  end for
  $\hat{\mathbf{s}} = \mathbf{s}_{\mathcal{K}(\varepsilon,\delta')}$
end if
Output: $\hat{\mathbf{s}}$ and $\Delta = \mathbf{g}^\top\hat{\mathbf{s}} + \frac{1}{2}\hat{\mathbf{s}}^\top\mathbf{H}\hat{\mathbf{s}} + \frac{M}{6}\|\hat{\mathbf{s}}\|_2^3$

Approximate Local Minimax Point Under the strict-minimax condition, we can find an approximate local minimax point by performing an additional AGD procedure on the output of MCN.

Corollary 2. *Suppose $f(\mathbf{x}, \mathbf{y})$ is (α, β, γ)-strict-minimax and satisfies Assumption 1-4. Let $\hat{\mathbf{x}}$ be the output of running Algorithm 2 with the setting of Corollary 1 and $\varepsilon = \min\left\{\alpha/3,\ \beta^2/(8\kappa^3\rho)\right\}$. Let $\hat{\mathbf{y}} = \mathrm{AGD}\big(-f(\hat{\mathbf{x}}, \cdot), \mathbf{y}_t, \hat{K}, \frac{1}{\ell}, \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)$ where t corresponds to the last iteration of Algorithm 2 such that $\hat{\mathbf{x}} = \mathbf{x}_t + \mathbf{s}_t^*$ and $\hat{K} = \sqrt{\kappa}\log\left(\min\left\{\frac{\alpha}{2\ell},\ \frac{\beta}{8\kappa^2\rho}\right\}/\left(\sqrt{\kappa+1}\left(\tilde{\varepsilon} + \frac{\kappa}{2^{2.25}}\sqrt{\frac{\varepsilon}{M}}\right)\right)\right)$. Then there exists a local minimax point $(\mathbf{x}^*, \mathbf{y}^*)$ of $f(\mathbf{x}, \mathbf{y})$ such that $\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2^2 + \|\mathbf{y}^* - \hat{\mathbf{y}}\|_2^2 \leq \gamma^2$.*

4 Inexact Minimax Cubic Newton Algorithm

In this section, we proposed an efficient algorithm called inexact minimax cubic Newton (IMCN), which avoids any operation related to the second-order oracle and only requires $\tilde{\mathcal{O}}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right)$ Hessian-vector product calls and $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$ gradient calls in total to find $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-SSP of $P(\mathbf{x})$. Since the Hessian-vector products can be computed as fast as gradients [28, 31], IMCN is much more efficient than MCN in high-dimensional cases.

Algorithm 5 Final-Cubic-Solver

Input: $\mathbf{g}$, $\mathbf{H}$, ε
$\mathbf{s}_0 = \mathbf{0}$, $\mathbf{g}_0 = \mathbf{g}$, $\eta = 1/(20L)$
for $t = 0, \ldots$ **do**
  if $\|\mathbf{g}_t\|_2 \leq \varepsilon/2$ **then break**
  $\mathbf{s}_{t+1} = \mathbf{s}_t - \eta\mathbf{g}_t$
  $\mathbf{g}_{t+1} = \mathbf{g} + \mathbf{H}\mathbf{s}_{t+1} + \frac{M}{2}\|\mathbf{s}_{t+1}\|_2\mathbf{s}_{t+1}$
end for
Output: $\mathbf{s}_t$

We present the details of IMCN in Algorithm 3. Unlike MCN which solves problem (4) exactly, IMCN uses gradient-based cubic sub-problem solver (Algorithm 4) [4] to compute $\mathbf{s}_t \approx$

$\arg\min_{\mathbf{s}\in\mathbb{R}^{d_x}} m_t(\mathbf{s}) \triangleq \mathbf{g}_t^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_t \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|_2^3$. If the condition in line 10 of Algorithm 3 holds, the point $\mathbf{x}_t + \mathbf{s}_t^*$ should be a desired approximate second-order stationary point. Due to $\mathbf{s}_t^*$ is hard to obtain, we introduce additional gradient descent steps (Algorithm 5) to approximate it by $\hat{\mathbf{s}}$ and use $\mathbf{x}_t + \hat{\mathbf{s}}$ as the final output.

The design and the convergence analysis of IMCN is more challenging than existing inexact cubic Newton algorithms for minimization problems [18, 36] since the Hessian estimator in IMCN has a more complicated structure. To address this issue, we approximate the Hessian $\nabla^2 P(\mathbf{x}_t)$ by $\mathbf{H}_t$ as follows

$$\mathbf{H}_t = \nabla_{xx}^2 f(\mathbf{x}_t, \mathbf{y}_t) + \nabla_{xy}^2 f(\mathbf{x}_t, \mathbf{y}_t)\mathbf{C}_t \nabla_{yx}^2 f(\mathbf{x}_t, \mathbf{y}_t), \tag{5}$$

where $\mathbf{C}_t = \frac{c_0}{4\ell}\mathbf{I} + \frac{1}{2\ell}\sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)$ and $\mathbf{Z}_t = \frac{4\ell}{\ell-\mu}\big(-\frac{1}{2\ell}\nabla_{yy}^2 f(\mathbf{x}_t, \mathbf{y}_t) - \frac{\ell+\mu}{4\ell}\mathbf{I}\big)$.

Here $\mathbf{T}_k(\cdot)$ is the matrix Chebyshev polynomials (shown in Section 4.1) leading to $\mathbf{C}_t \approx -\big(\nabla_{yy}^2(\mathbf{x}_t, \mathbf{y}_t)\big)^{-1}$. Note that we never construct matrix $\mathbf{H}_t$ explicitly in implementation because all operations related to $\mathbf{H}_t$ can be reduced to compute Hessian-vector products, which avoid any second-order oracle calls or matrix factorization/inversion.

Then we provide the convergence analysis for the IMCN algorithm. Throughout this section, we let $L = 2\kappa\ell$ and $M = 4\sqrt{2}\kappa^3\rho$ be the Lipschitz continuous constants of $\nabla P(x)$ and $\nabla^2 P(x)$. We suppose $\varepsilon \le L^2/M$, otherwise, the second-order condition $\nabla^2 P(\mathbf{x}) \succeq -\sqrt{M\varepsilon}\mathbf{I}$ always holds and we only need to use gradient methods [20, 21] to find first-order stationary point.

4.1 Approximating Hessian by Matrix Chebyshev Polynomials

We first show the error bound of matrix inverse approximation via matrix Chebyshev polynomials.

Lemma 7. *Suppose symmetric matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$ satisfieses $\mu'\mathbf{I} \preceq \mathbf{X} \preceq \ell'\mathbf{I}$ with $0 < \mu' \le \ell' < 1$, then we have $\big\|\mathbf{X}^{-1} - \big(\frac{c_0}{2}\mathbf{I} + \sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}')\big)\big\|_2 \le \frac{\sqrt{\ell'/\mu'}-1}{\sqrt{\ell'\mu'}}\big(1 - 2/(\sqrt{\ell'/\mu'}+1)\big)^{K'}$ where $\mathbf{Z}' = \frac{2}{\ell'-\mu'}\big(\mathbf{X} - \frac{\ell'+\mu'}{2}\mathbf{I}\big)$, $c_k = \frac{2}{\sqrt{\ell'\mu'}}\Big(\frac{\sqrt{\mu'/\ell'}-1}{\sqrt{\mu'/\ell'}+1}\Big)^k$, and $\mathbf{T}_k(\cdot)$ are matrix Chebyshev polynomials with $T_0(\mathbf{Z}') = \mathbf{I}$, $T_1(\mathbf{Z}') = \mathbf{Z}'$ and $\mathbf{T}_k(\mathbf{Z}') = 2\mathbf{Z}'\mathbf{T}_{k-1}(\mathbf{Z}') - \mathbf{T}_{k-2}(\mathbf{Z}')$ for $k \ge 2$.*

Based on Lemma 7, we can bound the approximation error of the Hessian estimator $\mathbf{H}_t$ as follows.

Lemma 8. *Using the notation of Algorithm 3, we have*

$$\big\|\nabla_{xx}^2 P(\mathbf{x}_t) - \mathbf{H}_t\big\|_2 \le 3\rho\kappa^2\sqrt{\kappa+1}\Big(1 - \frac{1}{\sqrt{\kappa}}\Big)^{K_t/2}\|\mathbf{y}_{t-1} - \mathbf{y}^*(\mathbf{x}_t)\|_2 + \kappa\ell\Big(1 - \frac{2}{\sqrt{\kappa}+1}\Big)^{K'}.$$

Lemma 8 means that taking the number of iterations for AGD with $K_t = \mathcal{O}\left(\sqrt{\kappa}\log(\kappa\rho/\varepsilon_H)\right) = \tilde{\mathcal{O}}\left(\sqrt{\kappa}\right)$[2] and the number of terms for matrix Chebyshev polynomials with $K' = \mathcal{O}\left(\sqrt{\kappa}\log(\kappa\ell/\varepsilon_H)\right) = \tilde{\mathcal{O}}\left(\sqrt{\kappa}\right)$ could achieve $\mathbf{H}_t$ with $\big\|\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t\big\|_2 \le \varepsilon_H$ for any $\varepsilon_H > 0$.

In the implementation of IMCN, all operations related to $\mathbf{H}_t$ can be viewed as computing Hessian-vector products. Actually, we can obtain $\mathbf{H}_t\mathbf{u}'$ with $\mathcal{O}(K') = \tilde{\mathcal{O}}(\sqrt{\kappa})$ Hessian-vector calls for any $\mathbf{u}' \in \mathbb{R}^{d_x}$, which avoids $\mathcal{O}\left(d_x^2 + d_y^2\right)$ space to keep Hessian matrices. The detailed implementation is presented in Appendix E.

4.2 Complexity Analysis for IMCN

The IMCN method calls a gradient-based sub-problem solver (Algorithm 4) to optimize the following cubic regularized problem [4, 36] in each iteration

$$\min_{\mathbf{s}\in\mathbb{R}^{d_x}} \tilde{m}_t(\mathbf{s}) \triangleq \mathbf{g}_t^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{H}_t \mathbf{s} + \frac{M}{6}\|\mathbf{s}\|_2^3. \tag{6}$$

[2] Rigorously speaking, the term $\log(\|\mathbf{y}_{t-1} - \mathbf{y}^*(\mathbf{x}_t)\|_2)$ also should be considered into the total complexity, which will be discussed in later sections.

It requires at most $\mathcal{K}(\varepsilon,\delta') = \mathcal{O}\big(\frac{L}{\sqrt{M}\varepsilon}\big(\log\big(\frac{\sqrt{d_x}}{\delta'}\big)+\log\big(\frac{L+\sqrt{\rho\varepsilon}}{\sqrt{M}\varepsilon}\big)\big)\big)$ number of iterations to achieve an approximate solution with enough accuracy. The detailed analysis for the complexity of Algorithm 4 is deferred to appendix D.

Then we bound the total number of iterations for Algorithm 3.

Theorem 3. *Running Algorithm 3 with $C_\sigma = 1/4$, $\delta' = \delta/T$, $T = \lceil 626(P(\mathbf{x}_0) - P^*)\sqrt{M}\varepsilon^{-1.5}\rceil$ and sufficient large K_t and K' such that Assumption 5 holds with $C_g = 1/240$ and $C_H = 1/200$. Then the condition $\Delta_t \geq -\frac{1}{128}\sqrt{\varepsilon^3/M}$ in line 11 must hold in no more than $T = \mathcal{O}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)$ iterations; and the output $\hat{\mathbf{x}}$ is an $(\varepsilon, 2\kappa^{1.5}\sqrt{\rho\varepsilon})$-SSP with probability $1-\delta$.*

We also bound the number of gradient calls from AGD procedure in Algorithm 3 as follows.

Theorem 4. *Under the setting of Theorem 3, if we run Algorithm 3 with*

$$K' = \left\lceil \frac{\sqrt{\kappa}+1}{2}\log\left(\frac{\kappa\ell}{2\min\left\{C_H\sqrt{M\varepsilon}, \varepsilon_H L\right\}}\right)\right\rceil \text{ and } K_t = \begin{cases} \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{y}^*(\mathbf{x}_0)\right\|_2\right)\right\rceil, & t=0 \\ \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left(\tilde{\varepsilon}+\kappa\left\|\mathbf{s}_{t-1}\right\|_2\right)\right)\right\rceil, & t\geq 1 \end{cases}$$

where $\tilde{\varepsilon} = \min\left\{C_g\varepsilon/\ell, \min\left\{C_H\sqrt{M\varepsilon}, \varepsilon_H L\right\}/(6\rho\kappa^2)\right\}$ and $\mathbf{s}_0 = \mathbf{0}$, then it holds that $\left\|\nabla P(\mathbf{x}_t) - \mathbf{g}_t\right\|_2 \leq C_g\varepsilon$, $\left\|\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t\right\|_2 \leq \min\left\{C_H\sqrt{M\varepsilon}, \varepsilon_H L\right\}$ and

$$\sum_{t=0}^{T} K_t \leq T+1+\frac{2\sqrt{\kappa}T}{3}\left[\frac{3}{T}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{y}^*(\mathbf{x}_0)\right\|_2\right)+\log\left(8(\kappa+1)^{1.5}+\frac{8\kappa^3(\kappa+1)^{1.5}}{T\tilde{\varepsilon}^3}\sum_{t=1}^{T}\left\|\mathbf{s}_{t-1}\right\|_2^3\right)\right].$$

Note that the value of $K' = \tilde{\mathcal{O}}(\sqrt{\kappa})$ corresponds to the number of Hessian vector calls for each iteration of cubic sub-problem solver (Algorithm 4 and 5). Combining Theorem 3, Theorem 4 and the value of $\mathcal{K}(\varepsilon,\delta')$, we obtain the main result for Algorithm 3 as follows.

Corollary 3. *Under Assumption 1-4, if we run Algorithm 3 with $C_\sigma = 1/4$, $\delta' = \delta/T$,*

$$K' = \left\lceil \frac{\sqrt{\kappa}+1}{2}\log\left(\frac{\kappa\ell}{2\min\left\{C_H\sqrt{M\varepsilon}, \frac{3L}{25}\right\}}\right)\right\rceil \text{ and } K_t = \begin{cases} \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{y}^*(\mathbf{x}_0)\right\|_2\right)\right\rceil, & t=0 \\ \left\lceil 2\sqrt{\kappa}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left(\tilde{\varepsilon}+\kappa\left\|\mathbf{s}_{t-1}\right\|_2\right)\right)\right\rceil, & t\geq 1 \end{cases}$$

where $T = \lceil 626(P(\mathbf{x}_0) - P^*)\sqrt{M}\varepsilon'^{-1.5}\rceil$, $C_g = 1/240$, $C_H = 1/200$, $\mathbf{s}_0 = \mathbf{0}$, $\tilde{\varepsilon}' = \varepsilon/4$ and $\tilde{\varepsilon} = \min\left\{C_g\varepsilon/(4\ell), \min\left\{C_H\sqrt{M\varepsilon}, \frac{3L}{25}\right\}/(12\rho\kappa^2)\right\}$, then the output $\hat{\mathbf{x}}$ is an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-second-order stationary point of $P(\mathbf{x})$ with probability $1-\delta$ and the number of gradient calls is at most $\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$. The number of Hessian-vector product calls is at most $\mathcal{O}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right)$.

Following the analysis of Corollary 2, we can also achieve an approximate local minimax point by running AGD based on the $\hat{\mathbf{x}}$ obtained from Algorithm 3.

5 Experiments

In this section, we conduct empirical studies for our methods against the classical GDA algorithm [20] on both synthetic problem and real-world application.

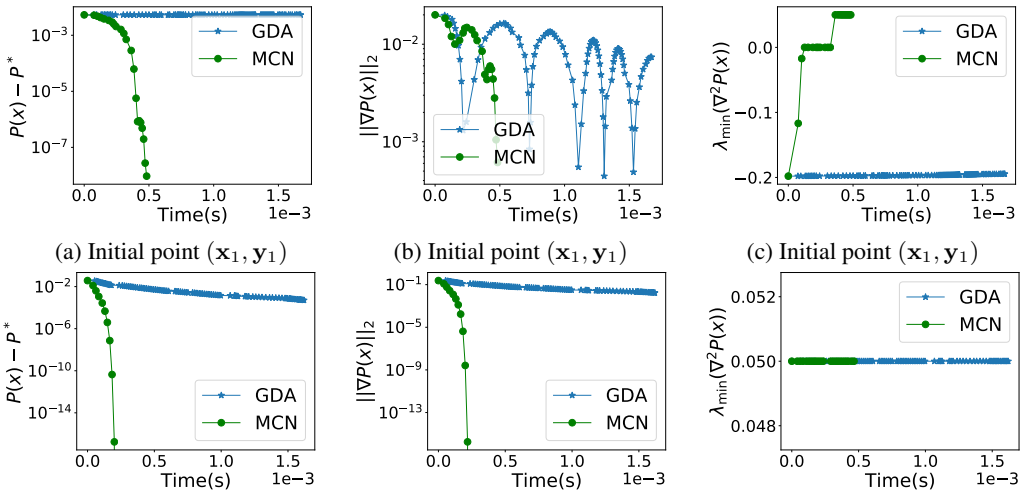
5.1 Synthetic Minimax Problem

We construct the following nonconvex-strongly-concave minimax problem:

$$\min_{\mathbf{x}\in\mathbb{R}^3}\max_{\mathbf{y}\in\mathbb{R}^2} f(\mathbf{x},\mathbf{y}) = w(x_3) - \frac{y_1^2}{40} + x_1y_1 - \frac{5y_2^2}{2} + x_2y_2 \tag{7}$$

where $\mathbf{x} = [x_1, x_2, x_3]^\top$, $\mathbf{y} = [y_1, y_2]^\top$ and $w(\cdot)$ is the W-shaped scalar function [36] whose exact form is shown in Appendix G.1. It is easy to verify that the problem has an strict saddle point at $(\mathbf{x}_0, \mathbf{y}_0) = ([0,0,0]^\top, [0,0]^\top)$.

We conduct experiments on problem (7) with two different initial points $(\mathbf{x}_1, \mathbf{y}_1) = \left([10^{-3}, 10^{-3}, 10^{-3}]^\top, [0,0]^\top\right)$ and $(\mathbf{x}_2, \mathbf{y}_2) = \left([0,0,1]^\top, [0,0]^\top\right)$. Notice that problem (7) has



(a) Initial point $(\mathbf{x}_1, \mathbf{y}_1)$ (b) Initial point $(\mathbf{x}_1, \mathbf{y}_1)$ (c) Initial point $(\mathbf{x}_1, \mathbf{y}_1)$

(d) Initial point $(\mathbf{x}_2, \mathbf{y}_2)$ (e) Initial point $(\mathbf{x}_2, \mathbf{y}_2)$ (f) Initial point $(\mathbf{x}_2, \mathbf{y}_2)$

Figure 1: We present comparisons of error, gradient norms and Hessian minimum eigenvalues on the synthetic problem. Figure (a), (b) and (c) shows the results with initial point $(\mathbf{x}_1, \mathbf{y}_1)$. Figure (d), (e) and (f) shows the results with initial point $(\mathbf{x}_2, \mathbf{y}_2)$.

an strict saddle point at $(\mathbf{x}_0, \mathbf{y}_0) = ([0,0,0]^\top, [0,0]^\top)$. The initial point $(\mathbf{x}_1, \mathbf{y}_1)$ is close to $(\mathbf{x}_0, \mathbf{y}_0)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ is far from $(\mathbf{x}_0, \mathbf{y}_0)$. We compare the proposed algorithm MCN with GDA. The learning rate of GDA and AGD step in MCN is selected from $\{c \cdot 10^{-i} : c \in \{1, 5\}, i \in \{1, 2, 3\}\}$. For MCN method, we choose $M = 10$. We compare the running time against $P(\mathbf{x}) - P^*$, $\|\nabla P(\mathbf{x})\|_2$ and $\lambda_{\min}(\nabla^2 P(\mathbf{x}))$ for two algorithms and plot the results in Figure 1. From the curves corresponding to initial point $(\mathbf{x}_2, \mathbf{y}_2)$, we observe that both MCN and GDA converge to the minimum when the initial point is far from the strict saddle point, but MCN converges much faster than GDA. When the initial point is close to the strict saddle point, Figure 4(b) shows that the GDA algorithm gets stuck at the strict saddle point since its Hessian minimum eigenvalue are always negative. However, our MCN algorithm can reach the points which have positive Hessian minimum eigenvalues.

5.2 Domain Adaptation

The Domain-Adversarial Neural Network (DANN) [9] is a classic method to domain adaptation. Suppose the source domain dataset is $\mathcal{S} = \{(\mathbf{a}_i^S, b_i^S)\}_{i=1}^{N_\mathcal{S}}$ where $\mathbf{a}_i^\mathcal{S}$ is the feature vector of the i-th sample and $b_i^\mathcal{S}$ is the corresponding label. The target domain dataset $\mathcal{T} = \{\mathbf{a}_i^\mathcal{T}\}_{i=1}^{N_\mathcal{T}}$ only contains features. Then DANN aims to solve the following nonconvex-strongly-concave minimax problem

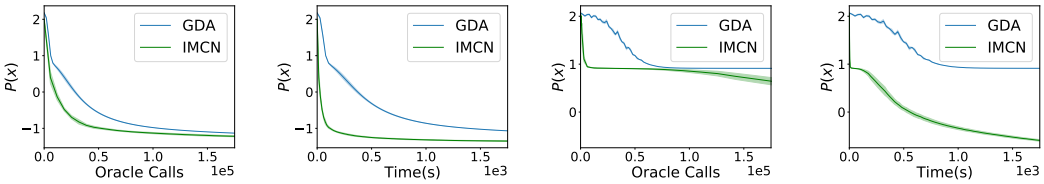
$$\min_{[\mathbf{x}_1;\mathbf{x}_2] \in \mathbb{R}^{d_x}} \max_{\mathbf{y} \in \mathbb{R}^{d_y}} L_1(\mathbf{x}_1, \mathbf{x}_2) - \alpha \cdot L_2(\mathbf{x}_1, \mathbf{y})$$

where $L_1(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{N_\mathcal{S}} \sum_{i=1}^{N_\mathcal{S}} l(\mathbf{x}_2; \Phi(\mathbf{x}_1; \mathbf{a}_i^\mathcal{S}), b_i^\mathcal{S})$ is the loss of supervised learning and

$$L_2(\mathbf{x}_1, \mathbf{y}) = \frac{1}{N_\mathcal{S}} \sum_{i=1}^{N_\mathcal{S}} D_\mathcal{S}(h(\mathbf{y}; \Phi(\mathbf{x}_1; \mathbf{a}_i^\mathcal{S}))) - \frac{1}{N_\mathcal{T}} \sum_{i=1}^{N_\mathcal{T}} D_\mathcal{T}(h(\mathbf{y}; \Phi(\mathbf{x}_1; \mathbf{a}_i^\mathcal{T}))) + \lambda \|\mathbf{y}\|^2$$

is the domain classification loss. Here Φ is a single-layer neural network of size $(28 \times 28) \times 200$ with parameter $\mathbf{x}_1$ and l is a two-layer neural network of size $200 \times 20 \times 10$ with parameter $\mathbf{x}_2$, followed by a cross entropy loss.We choose the sigmoid function as the activation function for them. With the commonly used logistic loss for L_2, we let $h(\mathbf{y}; \mathbf{z}) = 1/(1 + \exp(-\mathbf{y}^\top \mathbf{z}))$, $D_S(z) = 1 - \log(z)$ and $D_T(z) = \log(1 - z)$.

Since the dimension of the minimax problem is quite large, we implement the IMCN algorithm instead of the MCN algorithm for efficiency. We compare IMCN with GDA on the domain adaptation problem between two different datasets: MNIST [19] and MNIST-m [9]. Since we do not know the close form of $P(\mathbf{x})$, we estimate the value of $P(\mathbf{x}) = \max_\mathbf{y} f(\mathbf{x}, \mathbf{y})$ by AGD procedure. More details about our experimental setup can be found in Appendix G.2.



(a) MNIST to MNIST-m (b) MNIST to MNIST-m (c) MNIST-m to MNIST (d) MNIST-m to MNIST

Figure 2: We present results on DANN model. Figure (a) and (b) show the results of domain adaptation from MNIST to MNIST-m. Figure (c) and (d) show the results of domain adaptation from MNIST-m to MNIST.

We compare IMCN and GDA algorithms via running time and oracle calls and show the results in Figure 2. We run the experiments for five times with different random initialization and report the average results. The oracle calls of GDA only contains gradient while the oracle calls of IMCN contains both gradient and Hessian-vector product. Based on Figure 2, we observe that IMCN significantly outperforms GDA in both time and oracle comparison. Notice that IMCN requires to call much more gradient/Hessian-vector oracles on y than the gradient oracles on $\mathbf{x}$. These results verify our convergence analysis and show the advantage of proposed algorithm.

6 Conclusion and Future Work

In this paper, we study second-order optimization methods for nonconvex-strongly-concave minimax problems. We have proposed a novel algorithm so-called minimax cubic Newton (MCN) which could find an $(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon})$-SSP of the primal function with $\mathcal{O}(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5})$ second-order oracle calls and $\tilde{\mathcal{O}}(\kappa^{2}\sqrt{\rho}\varepsilon^{-1.5})$ gradient oracle calls. We also provide an efficient algorithm for high dimensional problem, which avoids accessing second-order oracle and contains $\tilde{\mathcal{O}}(\kappa^{1.5}\ell\varepsilon^{-2})$ Hessian-vector oracle calls and $\tilde{\mathcal{O}}(\kappa^{2}\sqrt{\rho}\varepsilon^{-1.5})$ gradient oracle calls. To best of our knowledge, this paper first achieves non-asymptotic convergence result for finding SSP of minimax problem without convex-concave assumption. It is interesting to investigate how to find SSPs by pure first-order methods, which are more commonly used in deep learning than second-order methods.

Acknowledgements

Luo Luo is supported by National Natural Science Foundation of China (No. 62206058) and Shanghai Sailing Program (22YF1402900). Cheng Chen is supported by Singapore Ministry of Education (AcRF) Tier 1 grant RG75/21.

References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *STOC*, 2017.

[2] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

[3] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016.

[4] Yair Carmon and John Duchi. Gradient descent finds the cubic-regularized nonconvex newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.

[5] Coralia Cartis, Nicholas I.M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[6] Coralia Cartis, Nicholas I.M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011.

[7] Ziyi Chen, Qunwei Li, and Yi Zhou. Escaping saddle points in nonconvex minimax optimization via cubic-regularized gradient descent-ascent. *arXiv preprint arXiv:2110.07098*, 2021.

[8] Tanner Fiez, Lillian Ratliff, Eric Mazumdar, Evan Faulkner, and Adhyyan Narang. Global convergence to local minmax equilibrium in classes of nonconvex zero-sum games. In *NeurIPS*, 2021.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[10] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*, 2015.

[11] Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *NIPS*, 2016.

[12] Zhishuai Guo, Zhuoning Yuan, Yan Yan, and Tianbao Yang. Fast objective & duality gap convergence for nonconvex-strongly-concave min-max problems. *arXiv preprint arXiv:2006.06889*, 2020.

[13] Yuze Han, Guangzeng Xie, and Zhihua Zhang. Lower complexity bounds of finite-sum optimization problems: The results and construction. *arXiv preprint arXiv:2103.08280*, 2021.

[14] Filip Hanzely, Nikita Doikov, Yurii Nesterov, and Peter Richtarik. Stochastic subspace cubic newton method. In *ICML*, 2020.

[15] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020.

[16] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR, 2017.

[17] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *ICML*, 2020.

[18] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *ICML*, 2017.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Tianyi Lin, Chi Jin, and Michael I. Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *ICML*, 2020.

[21] Tianyi Lin, Chi Jin, and Michael I. Jordan. Near-optimal algorithms for minimax optimization. In *COLT*, 2020.

[22] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. In *NeurIPS*, 2020.

[23] Luo Luo, Guangzeng Xie, Tong Zhang, and Zhihua Zhang. Near optimal stochastic algorithms for finite-sum unbalanced convex-concave minimax optimization. *arXiv preprint arXiv:2106.01761*, 2021.

[24] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[25] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[26] Yurii Nesterov and Boris T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[27] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[28] Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural computation*, 6(1): 147–160, 1994.

[29] Shuang Qiu, Zhuoran Yang, Xiaohan Wei, Jieping Ye, and Zhaoran Wang. Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear TD learning. *arXiv preprint arXiv:2008.10103*, 2020.

[30] Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D. Lee. On the convergence and robustness of training GANs with regularized optimal transport. *arXiv preprint arXiv:1802.08249*, 2018.

[31] Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723, 2002.

[32] Alexander Shapiro. Second-order derivatives of extremal-value functions and optimality conditions for semi-infinite programs. *Mathematics of Operations Research*, 10(2):207–219, 1985.

[33] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

[34] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2016.

[35] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, 2018.

[36] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I. Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *NeurIPS*, 2018.

[37] Yuanhao Wang and Jian Li. Improved algorithms for convex-concave minimax optimization. In *NeurIPS*, 2020.

[38] Yuanhao Wang, Guodong Zhang, and Jimmy Ba. On solving minimax optimization locally: A follow-the-ridge approach. In *ICLR*, 2019.

[39] Wenhan Xian, Feihu Huang, Yanfu Zhang, and Heng Huang. A faster decentralized algorithm for nonconvex minimax problems. *NeurIPS*, 2021.

[40] Tengyu Xu, Zhe Wang, Yingbin Liang, and H Vincent Poor. Gradient free minimax optimization: Variance reduction and faster convergence. *arXiv preprint arXiv:2006.09361*, 2020.

[41] Guojun Zhang, Kaiwen Wu, Pascal Poupart, and Yaoliang Yu. Newton-type methods for minimax optimization. *arXiv preprint arXiv:2006.14592*, 2020.

[42] Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. *arXiv preprint arXiv:2103.15888*, 2021.

[43] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularization methods. *Journal of Machine Learning Research*, 20(134):1–47, 2019.

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]
   (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]

The appendix is organized as follows. We first present some basic lemmas used in our proofs in Section A. Then we provide the proofs for Section 2, Section 3 and Section 4 in Section B, Section C and Section D, respectively. We compare the complexity of MCN and IMCN with first-order methods in Section F. We introduce the implementation details of IMCN in Section E and provide the details of experiments in Section G.

A Basic Lemmas

In this section, we provide some basic lemmas.

Lemma 9. *Under assumptions of Lemma 2, then* $\nabla_{xy}^2 f(\mathbf{x},\cdot)\left(\nabla_{yy}^2 f(\mathbf{x},\cdot)\right)^{-1}\nabla_{yx}^2 f(\mathbf{x},\cdot)$ *is* $3\kappa^2\rho$*-Lipschitz continuous for fixed* $\mathbf{x}$*.*

Proof. For fixed $x \in \mathbb{R}^{d_x}$ and any $y, y' \in \mathbb{R}^{d_y}$, we have

$$\begin{aligned}
&\left\|\left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y})\right)^{-1} - \left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y}')\right)^{-1}\right\|_2 \\
=&\left\|\left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y})\right)^{-1}\left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y}) - \nabla_{yy}^2 f(\mathbf{x},\mathbf{y}')\right)\left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y}')\right)^{-1}\right\|_2 \\
\leq&\frac{1}{\mu^2}\left\|\nabla_{yy}^2 f(\mathbf{x},\mathbf{y}) - \nabla_{yy}^2 f(\mathbf{x},\mathbf{y}')\right\|_2 \leq \frac{\rho}{\mu^2},
\end{aligned}$$

and

$$\left\|\left(\nabla_{yy}^2 f(\mathbf{x},\mathbf{y})\right)^{-1}\right\|_2 \leq \frac{1}{\mu}.$$

Similar to the analysis of Lemma 3, we apply Lemma 12 on $\left(\nabla_{yy}^2 f(\mathbf{x},\cdot)\right)^{-1}$ and $\nabla_{yx}^2 f(\mathbf{x},\cdot)$ which means $\left(\nabla_{yy}^2 f(\mathbf{x},\cdot)\right)^{-1}\nabla_{yx}^2 f(\mathbf{x},\cdot)$ is $(\kappa+1)\rho/\mu$-Lipschitz continuous and its norm is bounded by κ. Then again we apply Lemma 12 on $\nabla_{xy}^2 f(\mathbf{x},\cdot)$ and $\left(\nabla_{yy}^2 f(\mathbf{x},\cdot)\right)^{-1}\nabla_{yx}^2 f(\mathbf{x},\cdot)$, which means the Lipschitz continuous coefficient of $\nabla_{xy}^2 f(\mathbf{x},\cdot)\left(\nabla_{yy}^2 f(\mathbf{x},\cdot)\right)^{-1}\nabla_{yx}^2 f(\mathbf{x},\cdot)$ is $(\kappa^2+2\kappa)\rho \leq 3\kappa^2\rho$. □

Then we introduce two classical lemmas for cubic Newton methods.

Lemma 10 ([26])**.** *Under assumptions of Lemma 3, we have*

$$\left\|\nabla P(\mathbf{x}') - \nabla P(\mathbf{x}) - \nabla^2 P(\mathbf{x})(\mathbf{x}'-\mathbf{x})\right\|_2 \leq \frac{M}{2}\left\|\mathbf{x}-\mathbf{x}'\right\|_2^2 \tag{8}$$

and

$$\left|P(\mathbf{x}') - P(\mathbf{x}) - \nabla P(\mathbf{x})^\top(\mathbf{x}'-\mathbf{x}) - \frac{1}{2}(\mathbf{x}'-\mathbf{x})^\top\nabla^2 P(\mathbf{x})(\mathbf{x}'-\mathbf{x})\right| \leq \frac{M}{6}\left\|\mathbf{x}-\mathbf{x}'\right\|_2^3 \tag{9}$$

Lemma 11 ([26])**.** *For any* $M' > 0$*, the solution* $\mathbf{s}^*$ *of the following cubic regularized quadratic problem*

$$\mathbf{s}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^{d_x}}\left(\mathbf{g}^\top\mathbf{s} + \frac{1}{2}\mathbf{s}^\top\mathbf{H}\mathbf{s} + \frac{M'}{6}\left\|\mathbf{s}\right\|_2^3\right)$$

satisfies

$$\mathbf{g} + \mathbf{H}\mathbf{s}^* + \frac{M'}{2}\left\|\mathbf{s}^*\right\|_2\mathbf{s}^* = \mathbf{0} \tag{10}$$

$$\mathbf{H} + \frac{M'}{2}\left\|\mathbf{s}^*\right\|_2\mathbf{I} \succeq \mathbf{0} \tag{11}$$

$$\mathbf{g}^\top\mathbf{s}^* + \frac{1}{2}(\mathbf{s}^*)^\top\mathbf{H}\mathbf{s}^* + \frac{M'}{6}\left\|\mathbf{s}^*\right\|_2^3 \leq -\frac{M'}{12}\left\|\mathbf{s}^*\right\|_2^3 \tag{12}$$

B Proofs for Section 2

In this section, we provide the proofs for lemmas in Section 2.

B.1 The Proof of Lemma 2

Proof. The implicit function theorem means $\mathbf{y}^*(\cdot)$ is differentiable. The optimality of $\mathbf{y}^*$ means

$$\nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) = \mathbf{0}.$$

Taking total derivative on above equation achieves

$$\nabla^2_{yx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \nabla \mathbf{y}^*(\mathbf{x}) = \mathbf{0}. \tag{13}$$

Taking total derivative on the result of Lemma 1, we have

$$\nabla^2 P(\mathbf{x}) = \nabla^2_{xx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \nabla^2_{xy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \nabla \mathbf{y}^*(\mathbf{x}). \tag{14}$$

The strongly concavity on $\mathbf{y}$ implies $\nabla^2_{yy} f(\mathbf{x}, \mathbf{y})$ is non-singular. Hence, we connect (13) and (14) to obtain the desired result. □

B.2 The Proof of Lemma 3

We first introduce the following lemma.

Lemma 12. *Suppose operators $\mathcal{H}_1 : \mathcal{D} \to \mathcal{F}$ and $\mathcal{H}_2 : \mathcal{D} \to \mathcal{F}$ are α_1-Lipschitz continuous and α_2-Lipschitz continuous with respect to norms $\|\cdot\|_{\mathcal{D}}$ and $\|\cdot\|_{\mathcal{F}}$ defined on $\mathcal{D}$ and $\mathcal{F}$, that is, for any z and z' in $\mathcal{D}$, we have*

$$\|\mathcal{H}_1(\mathbf{Z}) - \mathcal{H}_1(\mathbf{Z}')\|_{\mathcal{F}} \le \alpha_1 \|\mathbf{Z} - \mathbf{Z}'\|_{\mathcal{D}} \quad \text{and} \quad \|\mathcal{H}_2(\mathbf{Z}) - \mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} \le \alpha_2 \|\mathbf{Z} - \mathbf{Z}'\|_{\mathcal{D}}.$$

We further suppose that there exist some $\beta_1, \beta_2 \ge 0$ such that $\|\mathcal{H}_1(\mathbf{Z})\|_{\mathcal{F}} \le \beta_1$ and $\|\mathcal{H}_2(\mathbf{Z})\|_{\mathcal{F}} \le \beta_2$ for any z in $\mathcal{D}$; and norm $\|\cdot\|_{\mathcal{F}}$ is submultiplicative on $\mathcal{F}$, that is, for any $\mathbf{Z}$ and $\mathbf{Z}'$ in $\mathcal{F}$, we have $\|\mathbf{Z}\mathbf{Z}'\|_{\mathcal{F}} \le \|\mathbf{Z}\|_{\mathcal{F}} \|\mathbf{Z}'\|_{\mathcal{F}}$ if $\mathbf{Z}\mathbf{Z}'$ also in $\mathcal{F}$. Then, the operator $\mathcal{H}_1(\cdot)\mathcal{H}_2(\cdot)$ is $(\alpha_1\beta_2 + \alpha_2\beta_1)$-Lipschitz continuous, that is, for any $\mathbf{Z}, \mathbf{Z}' \in \mathcal{D}$, we have

$$\|\mathcal{H}_1(\mathbf{Z})\mathcal{H}_2(\mathbf{Z}) - \mathcal{H}_1(\mathbf{Z}')\mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} \le (\alpha_1\beta_2 + \alpha_2\beta_1) \|\mathbf{Z} - \mathbf{Z}'\|_{\mathcal{D}}.$$

Proof. For any $\mathbf{Z}, \mathbf{Z}' \in \mathcal{D}$, we have

$$\begin{aligned}
&\|\mathcal{H}_1(\mathbf{Z})\mathcal{H}_2(\mathbf{Z}) - \mathcal{H}_1(\mathbf{Z}')\mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} \\
\le& \|\mathcal{H}_1(\mathbf{Z})\mathcal{H}_2(\mathbf{Z}) - \mathcal{H}_1(\mathbf{Z})\mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} + \|\mathcal{H}_1(\mathbf{Z})\mathcal{H}_2(\mathbf{Z}') - \mathcal{H}_1(\mathbf{Z}')\mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} \\
\le& \|\mathcal{H}_1(\mathbf{Z})\|_{\mathcal{F}} \|\mathcal{H}_2(\mathbf{Z}) - \mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} + \|\mathcal{H}_2(\mathbf{Z}')\|_{\mathcal{F}} \|\mathcal{H}_1(\mathbf{Z}) - \mathcal{H}_1(\mathbf{Z}')\|_{\mathcal{F}} \\
\le& (\alpha_1\beta_2 + \alpha_2\beta_1) \|\mathbf{Z} - \mathbf{Z}'\|_{\mathcal{D}}.
\end{aligned}$$

□

Then we prove Lemma 3 as follows.

Proof. Using Lemma 2 and triangle inequality, we have

$$\begin{aligned}
&\left\|\nabla^2 P(\mathbf{x}) - \nabla^2 P(\mathbf{x}')\right\|_2 \\
\le& \left\|\nabla^2_{xx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{xx} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))\right\|_2 + \\
&\left\|\nabla^2_{xy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})))^{-1} \nabla^2_{yx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \right. \\
&\quad \left. - \nabla^2_{xy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))(\nabla^2_{yy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')))^{-1} \nabla^2_{yx} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))\right\|.
\end{aligned} \tag{15}$$

Assumption 2 means $\nabla^2_{xx} f(\cdot, \cdot)$ is ρ-Lipschitz continuous, which implies for any x and x', we have

$$\begin{aligned}
&\left\|\nabla^2_{xx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{xx} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))\right\|_2^2 \\
\le& \rho^2 \left(\|\mathbf{x} - \mathbf{x}'\|_2^2 + \|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x}')\|_2^2\right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \rho^2 \left(\|\mathbf{x}-\mathbf{x}'\|_2^2 + \kappa^2 \|\mathbf{x}-\mathbf{x}'\|_2^2 \right) \\
&\leq 2\kappa^2\rho^2 \|\mathbf{x}-\mathbf{x}'\|_2^2 ,
\end{aligned}
$$

where the second inequality use Lemma 1 that $\mathbf{y}^*(\cdot)$ is κ-Lipschitz continuous. In other words, the partial Hessian $\nabla^2_{xx} f(\cdot, \mathbf{y}^*(\cdot))$ is $\sqrt{2}\kappa\rho$-Lipschitz continuous. Similarly, the partial Hessians $\nabla^2_{xy} f(\cdot, \mathbf{y}^*(\cdot))$, $\nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot))$ and $\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))$ are also $\sqrt{2}\kappa\rho$-Lipschitz continuous.

Then we verify the Lipschitz continuity of $\left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1}$. For any x and x', we have

$$
\begin{aligned}
&\left\| \left(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right)^{-1} - \left(\nabla^2_{yy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))\right)^{-1} \right\|_2 \\
=&\left\| \left(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right)^{-1} \left(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{yy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))\right) \left(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}'))\right)^{-1} \right\|_2 \\
\leq&\frac{1}{\mu^2} \left\| \nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{yy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) \right\|_2 \leq \frac{\sqrt{2}\kappa\rho}{\mu^2} \|\mathbf{x}-\mathbf{x}'\|_2 ,
\end{aligned}
$$

where we use the strongly-concavity of f which implies $(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})))^{-1} \preceq \mu^{-1}\mathbf{I}$, then $\left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1}$ is $(\sqrt{2}\kappa\rho/\mu^2)$-Lipschitz continuous.

We use Lemma 12 with

$$
\begin{aligned}
&\mathcal{H}_1(\cdot) = \left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1}, \quad \mathcal{H}_2(\cdot) = \nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot)), \\
&\alpha_1 = \frac{\sqrt{2}\kappa\rho}{\mu^2}, \quad \beta_1 = \frac{1}{\mu}, \quad \alpha_2 = \sqrt{2}\kappa\rho, \quad \beta_2 = \ell,
\end{aligned}
$$

which means $\left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1} \nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot))$ is $(2\sqrt{2}\kappa^2\rho/\mu)$-Lipschitz continuous, since

$$
\alpha_1\beta_2 + \alpha_2\beta_1 = \frac{\sqrt{2}\kappa^2\rho}{\mu} + \frac{\sqrt{2}\kappa\rho}{\mu} \leq \frac{2\sqrt{2}\kappa^2\rho}{\mu}.
$$

We also have $\left\| \left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1} \nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot)) \right\|_2 \leq \kappa$. We use Lemma 12 again with

$$
\begin{aligned}
&\mathcal{H}_1(\cdot) = \nabla^2_{xy} f(\cdot, \mathbf{y}^*(\cdot)), \quad \mathcal{H}_2(\cdot) = \left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1} \nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot)) \\
&\alpha_1 = \sqrt{2}\kappa\rho, \quad \beta_1 = \ell, \quad \alpha_2 = \frac{2\sqrt{2}\kappa^2\rho}{\mu}, \quad \beta_2 = \kappa,
\end{aligned}
$$

then we obtain

$$
\alpha_1\beta_2 + \alpha_2\beta_1 = \sqrt{2}\kappa^2\rho + 2\sqrt{2}\kappa^3\rho \leq 3\sqrt{2}\kappa^3\rho
$$

and $\nabla^2_{xy} f(\cdot, \mathbf{y}^*(\cdot)) \left(\nabla^2_{yy} f(\cdot, \mathbf{y}^*(\cdot))\right)^{-1} \nabla^2_{yx} f(\cdot, \mathbf{y}^*(\cdot))$ is $3\sqrt{2}\kappa^3\rho$-Lipschitz continuous. Finally, we have

$$
\begin{aligned}
&\left\| \nabla^2 P(\mathbf{x}) - \nabla^2 P(\mathbf{x}') \right\|_2 \\
\leq& \left\| \nabla^2_{xx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{xx} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) \right\|_2 \\
&+ \left\| \nabla^2_{xy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})))^{-1} \nabla^2_{yx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \right. \\
&\qquad \left. - \nabla^2_{xy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))(\nabla^2_{yy} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')))^{-1} \nabla^2_{yx} f(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) \right\| \\
\leq& \sqrt{2}\kappa\rho \|\mathbf{x}-\mathbf{x}'\|_2 + 3\sqrt{2}\kappa^3\rho \|\mathbf{x}-\mathbf{x}'\|_2 \\
=& 4\sqrt{2}\kappa^3\rho \|\mathbf{x}-\mathbf{x}'\|_2 .
\end{aligned}
$$

□

C Proofs for Section 3

In this section, we provide the proofs for theorems and lemmas in Section 3.

C.1 Proof of Lemma 6

Proof. Lemma 11 means

$$
\begin{aligned}
&\|\nabla P(\mathbf{x}_{t+1})\|_2 \\
=&\left\|\nabla P(\mathbf{x}_{t+1}) - \mathbf{g}_t - \mathbf{H}_t\mathbf{s}_t - \frac{M}{2}\|\mathbf{s}_t\|_2 \mathbf{s}_t\right\|_2 \\
\leq&\left\|\nabla P(\mathbf{x}_{t+1}) - \nabla P(\mathbf{x}_t) - \nabla^2 P(\mathbf{x}_t)\mathbf{s}_t\right\|_2 + \|\nabla P(\mathbf{x}_t) - \mathbf{g}_t\|_2 + \left\|\nabla^2 P(\mathbf{x}_t)\mathbf{s}_t - \mathbf{H}_t\mathbf{s}_t\right\|_2 + \frac{M}{2}\|\mathbf{s}_t\|_2^2 \\
\leq&\frac{M}{2}\|\mathbf{s}_t\|_2^2 + C_g\varepsilon + C_H\sqrt{M\varepsilon}\|\mathbf{s}_t\|_2 + \frac{M}{2}\|\mathbf{s}_t\|_2^2 \\
=&M\|\mathbf{s}_t\|_2^2 + C_g\varepsilon + C_H\sqrt{M\varepsilon}\|\mathbf{s}_t\|_2 \\
\leq&M\|\mathbf{s}_t\|_2^2 + C_g\varepsilon + \frac{C_H\left(\varepsilon + M\|\mathbf{s}_t\|_2^2\right)}{2} \\
=&\frac{(1+C_H)M}{2}\|\mathbf{s}_t\|_2^2 + \left(C_g + \frac{C_H}{2}\right)\varepsilon
\end{aligned}
$$

which implies

$$
\|\mathbf{s}_t\|_2^2 \geq \frac{2}{(1+C_H)M}\left(\|\nabla P(\mathbf{x}_{t+1})\|_2 - \left(C_g + \frac{C_H}{2}\right)\varepsilon\right).
$$

We also have

$$
\begin{aligned}
&\nabla^2 P(\mathbf{x}_{t+1}) \\
\succeq&\mathbf{H}_t - \left\|\mathbf{H}_t - \nabla^2 P(\mathbf{x}_{t+1})\right\|_2 \mathbf{I} \\
\succeq&-\frac{M}{2}\|\mathbf{s}_t\|_2 \mathbf{I} - \left\|\mathbf{H}_t - \nabla^2 P(\mathbf{x}_{t+1})\right\|_2 \mathbf{I} \\
\succeq&-\frac{M}{2}\|\mathbf{s}_t\|_2 \mathbf{I} - \left\|\mathbf{H}_t - \nabla^2 P(\mathbf{x}_t)\right\|_2 \mathbf{I} - \left\|\nabla^2 P(\mathbf{x}_t) - \nabla^2 P(\mathbf{x}_{t+1})\right\|_2 \mathbf{I} \\
\succeq&-\frac{M}{2}\|\mathbf{s}_t\|_2 \mathbf{I} - C_H\sqrt{M\varepsilon}\mathbf{I} - M\|\mathbf{s}_t\|_2 \mathbf{I} \\
\succeq&-\frac{M+2M}{2}\|\mathbf{s}_t\|_2 \mathbf{I} - C_H\sqrt{M\varepsilon}\mathbf{I}
\end{aligned}
$$

which implies

$$
\|\mathbf{s}_t\|_2 \geq -\frac{2}{M+2M}\lambda_{\min}\left(\nabla^2 P(\mathbf{x}_{t+1})\right) - \frac{2C_H\sqrt{M}}{M+2M}\sqrt{\varepsilon}.
$$

If $\mathbf{x}_{t+1}$ is not an $(\varepsilon, \sqrt{M\varepsilon})$-second-order stationary point, then

- If $\|\nabla P(\mathbf{x}_{t+1})\|_2 \geq \varepsilon$, we have

$$
\|\mathbf{s}_t\|_2 \geq \sqrt{\frac{2}{(1+C_H)M}\left(1 - C_g - \frac{C_H}{2}\right)\varepsilon} > \frac{1}{2}\sqrt{\frac{\varepsilon}{M}}.
$$

- If $-\lambda_{\min}\left(\nabla^2 P(\mathbf{x}_{t+1})\right) \geq \sqrt{M\varepsilon}$, we have

$$
\|\mathbf{s}_t\|_2 \geq \frac{2}{3M}\sqrt{M\varepsilon} - \frac{2C_H\sqrt{M\varepsilon}}{3M} > \frac{1}{2}\sqrt{\frac{\varepsilon}{M}}.
$$

□

C.2 Proof of Theorem 1

Proof. Lemma 6 means we only needs to show that T is no more than $\lceil 192(P(\mathbf{x}_0)-P^*)\sqrt{M}\varepsilon^{-1.5}\rceil$ by assuming $\|\mathbf{s}_t^*\|_2 \geq \frac{1}{2}\sqrt{\varepsilon/M}$ for all $t=1,\ldots,T-1$. We have

$$\begin{aligned}
&P(\mathbf{x}_{t+1})-P(\mathbf{x}_t)\\
\leq&\nabla P(\mathbf{x}_t)^\top \mathbf{s}_t^* + \frac{1}{2}(\mathbf{s}_t^*)^\top \nabla^2 P(\mathbf{x}_t)\mathbf{s}_t^* + \frac{M}{6}\|\mathbf{s}_t^*\|_2^3\\
=&\mathbf{g}_t^\top \mathbf{s}_t^* + \frac{1}{2}(\mathbf{s}_t^*)^\top \mathbf{H}_t\mathbf{s}_t^* + \frac{M}{6}\|\mathbf{s}_t^*\|_2^3 + (\nabla P(\mathbf{x}_t)-\mathbf{g}_t)^\top \mathbf{s}_t^* + \frac{1}{2}(\mathbf{s}_t^*)^\top(\nabla^2 P(\mathbf{x}_t)-\mathbf{H}_t)\mathbf{s}_t^*\\
\leq&-\frac{M}{12}\|\mathbf{s}_t^*\|_2^3 + C_g\varepsilon\|\mathbf{s}_t^*\|_2 + \frac{C_H\sqrt{M\varepsilon}}{2}\|\mathbf{s}_t^*\|_2^2\\
\leq&-\frac{M}{12}\|\mathbf{s}_t^*\|_2^3 + 4C_gM\|\mathbf{s}_t^*\|_2^3 + C_HM\|\mathbf{s}_t^*\|_2^3
\end{aligned} \tag{16}$$

where the first inequality comes from (9) of Lemma 10; the second inequality comes from (12) of Lemma 11 and Assumption 5; the last step is due to the assumption $\|\mathbf{s}_t^*\|_2 \geq \frac{1}{2}\sqrt{\varepsilon/M}$. The result of (16) can be written as

$$\begin{aligned}
&\frac{1}{8}\left(\frac{1}{12}-4C_g-C_H\right)\sqrt{\frac{\varepsilon^3}{M}}\\
\leq&\left(\frac{1}{12}-4C_g-C_H\right)M\|\mathbf{s}_t^*\|_2^3\\
\leq&P(\mathbf{x}_t)-P(\mathbf{x}_{t+1}).
\end{aligned} \tag{17}$$

Summing over inequality (17) with $t=0,\ldots,T-1$, we have

$$\frac{T}{192}\sqrt{\frac{\varepsilon^3}{M}} \leq P(\mathbf{x}_0)-P(\mathbf{x}_t) \leq P(\mathbf{x}_0)-P^*$$

which implies $T \leq 192(P(\mathbf{x}_0)-P^*)\sqrt{M}\varepsilon^{-3/2}$. Hence, Lemma 6 means the algorithm could find an $(\varepsilon,\sqrt{M\varepsilon})$-second-order stationary point of $P(\mathbf{x})$ with at most $\lceil 192(P(\mathbf{x}_0)-P^*)\sqrt{M}\varepsilon^{-1.5}\rceil+1$ iterations. □

C.3 Proof of Theorem 2

Proof. We first use induction to show that

$$\|\mathbf{y}_t-\mathbf{y}^*(\mathbf{x}_t)\|_2 \leq \tilde{\varepsilon} \tag{18}$$

holds for any $t \geq 0$. For $t=0$, Lemma 5 directly implies $\|\mathbf{y}_0-\mathbf{y}^*(\mathbf{x}_0)\|_2 \leq \tilde{\varepsilon}$. Suppose it holds that $\|\mathbf{y}_{t-1}-\mathbf{y}^*(x_{t-1})\|_2 \leq \tilde{\varepsilon}$ for any $t=t'-1$, then we have

$$\begin{aligned}
&\|\mathbf{y}_{t'}-\mathbf{y}^*(x_{t'})\|_2\\
\leq&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\|\mathbf{y}_{t'-1}-\mathbf{y}^*(\mathbf{x}_{t'})\|_2\\
\leq&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\|\mathbf{y}_{t'-1}-\mathbf{y}^*(\mathbf{x}_{t'-1})\|_2+\|\mathbf{y}^*(\mathbf{x}_{t'-1})-\mathbf{y}^*(\mathbf{x}_{t'})\|_2\right)\\
\leq&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\tilde{\varepsilon}+\kappa\|\mathbf{x}_{t'-1}-\mathbf{x}_{t'}\|_2\right)\\
=&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\tilde{\varepsilon}+\kappa\left\|\mathbf{s}_{t'-1}^*\right\|_2\right) \leq \tilde{\varepsilon},
\end{aligned}$$

where the first inequality is based on Lemma 5; the second one use triangle inequality; the third one is based on induction hypothesis and the last step use the definition of K_t and $\tilde{\varepsilon}$.

Combining inequality (18) with Lemma 1, Assumption 1 and Assumption 2, we obtain

$$\|\mathbf{g}_t-\nabla P(\mathbf{x}_t)\|_2$$

$$
\begin{aligned}
&= \left\|\nabla_x f(\mathbf{x}_t, \mathbf{y}_t) - \nabla_x f(\mathbf{x}_t, \mathbf{y}^*(\mathbf{x}_t))\right\|_2 \\
&\leq \ell \left\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\right\|_2 \leq C_g \varepsilon
\end{aligned}
$$

and

$$
\begin{aligned}
&\left\|\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t\right\|_2 \\
&= \left\|\nabla^2 f(\mathbf{x}_t, \mathbf{y}_t^*(\mathbf{x}_t)) - \nabla^2 f(\mathbf{x}_t, \mathbf{y}_t)\right\|_2 \\
&\leq \rho \left\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\right\|_2 \leq C_H \sqrt{M\varepsilon}.
\end{aligned}
$$

The total gradient calls from AGD in Algorithm 2 satisties

$$
\begin{aligned}
&\sum_{t=0}^{T} K_t \\
\leq& 2\sqrt{\kappa}\left[K_0 + \sum_{t=1}^{T} \log\left(\sqrt{\kappa+1} + \frac{\kappa\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{s}_{t-1}^*\right\|_2\right)\right] + T + 1 \\
=& \frac{2\sqrt{\kappa}}{3}\left[3K_0 + \sum_{t=1}^{T} \log\left(\sqrt{\kappa+1} + \frac{\kappa\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{s}_{t-1}^*\right\|_2\right)^3\right] + T + 1 \\
\leq& \frac{2\sqrt{\kappa}}{3}\left[3K_0 + \sum_{t=1}^{T} \log\left(8(\kappa+1)^{1.5} + \frac{8\kappa^3(\kappa+1)^{1.5}}{\tilde{\varepsilon}^3}\left\|\mathbf{s}_{t-1}^*\right\|_2^3\right)\right] + T + 1 \\
=& \frac{2\sqrt{\kappa}}{3}\left[3K_0 + \log\left(\prod_{t=1}^{T}\left(8(\kappa+1)^{1.5} + \frac{8\kappa^3(\kappa+1)^{1.5}}{\tilde{\varepsilon}^3}\left\|\mathbf{s}_{t-1}^*\right\|_2^3\right)\right)\right] + T + 1 \\
\leq& \frac{2\sqrt{\kappa}}{3}\left[3K_0 + \log\left(\frac{1}{T}\sum_{t=1}^{T}\left(8(\kappa+1)^{1.5} + \frac{8\kappa^3(\kappa+1)^{1.5}}{\tilde{\varepsilon}^3}\left\|\mathbf{s}_{t-1}^*\right\|_2^3\right)\right)^T\right] + T + 1 \\
=& \frac{2\sqrt{\kappa}T}{3}\left[\frac{3}{T}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{y}^*(\mathbf{x}_0)\right\|_2\right) + \log\left(8(\kappa+1)^{1.5} + \frac{8\kappa^3(\kappa+1)^{1.5}}{T\tilde{\varepsilon}^3}\sum_{t=1}^{T}\left\|\mathbf{s}_{t-1}^*\right\|_2^3\right)\right] + T + 1
\end{aligned}
$$

where the first inequality is based on the fact $(a+b)^3 \leq 8(a^3+b^3)$ for $a, b \geq 0$; the second inequality is based on AM–GM inequality. □

C.4 Proof of Corollary 1

Proof. The output is a desired second-order-stationary point can be proved by directly combining Theorem 1 and Theorem 2. Here we introduce ε' to eliminate the constant term $4\sqrt{2}$ in M. Connecting the upper bound of $\sum_{t=1}^{T} K_t$ in Theorem 2 and inequality (17) in the proof of Theorem 1, we have

$$
\begin{aligned}
\sum_{t=1}^{T} K_t \leq& \frac{2\sqrt{\kappa}T}{3}\log\left(\frac{3}{T}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\left\|\mathbf{y}^*(\mathbf{x}_0)\right\|_2\right) + 8(\kappa+1)^{1.5} + \frac{192\kappa^3(\kappa+1)^{1.5}}{TM\tilde{\varepsilon}'^3}\left(P(\mathbf{x}_0) - P^*\right)\right) \\
=& \tilde{\mathcal{O}}\left(\sqrt{\kappa M}\varepsilon^{-1.5}\right) = \tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right).
\end{aligned}
$$

The claim follows from the fact that we call gradient oracle for $\mathcal{O}(\sum_{t=1}^{T} K_t)$ times and perform Hessian (inverse) and exact cubic sub-problem solver calls for $\mathcal{O}(T)$ times. □

C.5 Proof of Corollary 2

Proof. Following the the proof of Theorem 2, Lemma 5 means

$$
\begin{aligned}
&\|\hat{\mathbf{y}} - \mathbf{y}^*(\hat{\mathbf{x}})\|_2 \\
\leq &\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{\hat{K}/2} \|\mathbf{y}_t - \mathbf{y}^*(\hat{\mathbf{x}})\|_2 \\
\leq &\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{\hat{K}/2} \left(\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\|_2 + \|\mathbf{y}^*(\mathbf{x}_t) - \mathbf{y}^*(\hat{\mathbf{x}})\|_2\right) \\
\leq &\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{\hat{K}/2} (\tilde{\varepsilon} + \kappa \|\mathbf{s}_t^*\|_2) \\
\leq &\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{\hat{K}/2} \left(\tilde{\varepsilon} + \frac{\kappa}{2^{2.25}}\sqrt{\frac{\varepsilon}{M}}\right) \\
\leq &\min\left\{\frac{\alpha}{2\ell}, \frac{\beta}{8\kappa^2\rho}\right\}
\end{aligned}
\tag{19}
$$

where we use $\|\mathbf{s}_t^*\|_2 \leq \frac{1}{2}\sqrt{\varepsilon/M}$ which is based on Lemma 6. Corollary 1 means $\hat{\mathbf{x}}$ is an $\left(\varepsilon, \kappa^{1.5}\sqrt{\rho\varepsilon}\right)$-second-order stationary point of $P(\mathbf{x})$. Then using smoothness of f and inequality (19), we have

$$
\begin{aligned}
\|\nabla_x f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\|_2 &\leq \|\nabla_x f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\|_2 + \|\nabla_x f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}})) - \nabla_x f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\|_2 \\
&\leq \|\nabla P(\hat{\mathbf{x}})\|_2 + \ell\|\mathbf{y}^*(\hat{\mathbf{x}}) - \hat{\mathbf{y}}\|_2 = \frac{5\alpha}{6}, \\
\|\nabla_y f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\|_2 &= \|\nabla_y f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla_y f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\|_2 \leq \ell\|\hat{\mathbf{y}} - \mathbf{y}^*(\hat{\mathbf{x}})\|_2 \leq \frac{\alpha}{2},
\end{aligned}
$$

which means $\|\nabla f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\|_2 \leq \alpha$. We also have

$$
\begin{aligned}
&\nabla_{xx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla_{xy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\left(\nabla_{yy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\right)^{-1}\nabla_{yx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \\
\succeq &\nabla_{xx}^2 P(\hat{\mathbf{x}}) - \left\|\nabla_{xx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla_{xy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\left(\nabla_{yy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\right)^{-1}\nabla_{yx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla^2 P(\hat{\mathbf{x}})\right\|_2 \mathbf{I} \\
\succeq &\nabla_{xx}^2 P(\hat{\mathbf{x}}) - \left\|\nabla_{xx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla_{xx}^2 f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\right\|_2 \mathbf{I} \\
&- \left\|\nabla_{xy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\left(\nabla_{yy}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}})\right)^{-1}\nabla_{yx}^2 f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \nabla_{xy}^2 f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\left(\nabla_{yy}^2 f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\right)^{-1}\nabla_{yx}^2 f(\hat{\mathbf{x}}, \mathbf{y}^*(\hat{\mathbf{x}}))\right\|_2 \mathbf{I} \\
\succeq &-\frac{\beta}{2}\mathbf{I} - \kappa\rho\|\hat{\mathbf{y}} - \mathbf{y}^*(\hat{\mathbf{x}})\|_2 \mathbf{I} - 3\kappa^2\rho\|\hat{\mathbf{y}} - \mathbf{y}^*(\hat{\mathbf{x}})\|_2 \mathbf{I} \\
\succeq &-\beta\mathbf{I},
\end{aligned}
$$

where the third inequality depends Lemma 9 that $\nabla_{xy}^2 f(\mathbf{x}, \cdot)\left(\nabla_{yy}^2 f(\mathbf{x}, \cdot)\right)^{-1}\nabla_{yx}^2$ is $3\kappa^2\rho$-Lipschitz continuous.Hence, Definition 5 means there exists a local minimax point $(\mathbf{x}^*, \mathbf{y}^*)$ such that $\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 + \|\hat{\mathbf{y}} - \mathbf{y}^*\|_2^2 \leq \gamma^2$. □

D Proof of Section 4

In this section, we provide the proofs for Theorems and Lemmas in Section 4.

D.1 Proof of Lemma 7

Proof. Since $\mathbf{X}$ is symmetric positive definite and $\mathbf{0} \prec \mu'\mathbf{I} \preceq \mathbf{X} \preceq \ell'\mathbf{I} \prec \mathbf{I}$, we have Axelsson [2, Section 8.6.1]

$$
\mathbf{X}^{-1} = \frac{c_0}{2}\mathbf{I} + \sum_{k=1}^{\infty} c_k \mathbf{T}_k(\mathbf{Z}') \tag{20}
$$

Chebyshev polynomials defined on scalar domain $-1 \leq \lambda \leq 1$ can be written as $T_k(\lambda) = \cos\left(k\cos^{-1}(\lambda)\right)$. Let the eigenvalue decomposition of $\mathbf{Z}'$ is $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, then we have $\|\mathbf{T}_k(\mathbf{Z}')\|_2 =$

$\left\|\mathbf{U}\mathbf{T}_k(\mathbf{\Lambda})\mathbf{U}^\top\right\|_2 \leq 1$. Combining definition of c_k and (20), we have

$$\begin{aligned}
&\left\|\mathbf{X}^{-1} - \left(\frac{c_0}{2}\mathbf{I} + \sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}')\right)\right\|_2 \\
=&\left\|\sum_{k=K'+1}^{\infty} c_k \mathbf{T}_k(\mathbf{Z}')\right\|_2 \leq \frac{2}{\sqrt{\ell'\mu'}} \sum_{k=K'+1}^{\infty} \left|\frac{\sqrt{\mu'/\ell'}-1}{\sqrt{\mu'/\ell'}+1}\right|^k \\
=&\frac{\sqrt{\ell'/\mu'}-1}{\sqrt{\ell'\mu'}}\left(1 - \frac{2}{\sqrt{\ell'/\mu'}+1}\right)^{K'}.
\end{aligned}$$

□

D.2 Proof of Lemma 8

Proof. Recall that $-\ell\mathbf{I} \preceq \nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t) \preceq -\mu\mathbf{I}$. We estimate the inverse of Hessian of $\mathbf{y}$ as

$$\left(-\frac{1}{2\ell}\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1} \approx \frac{c_0}{2}\mathbf{I} + \sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t).$$

Lemma 7 means

$$\left\|\left(-\frac{1}{2\ell}\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1} - \left(\frac{c_0}{2}I + \sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)\right)\right\|_2 \leq 2(\kappa - \sqrt{\kappa})\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^{K'}.$$

Hence, we have

$$\begin{aligned}
&\left\|\nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(-\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1}\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t) - \nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(\frac{c_0}{4\ell}I + \frac{1}{2\ell}\sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)\right)\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right\|_2 \\
\leq&\left\|\nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\right\|_2 \left\|\left(-\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1} - \left(\frac{c_0}{4\ell}I + \frac{1}{2\ell}\sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)\right)\right\|_2 \left\|\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right\|_2 \\
\leq&\ell(\kappa - \sqrt{\kappa})\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^{K'} \leq \kappa\ell\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^{K'}
\end{aligned} \tag{21}$$

Consider that $\nabla^2_{xx} P(\mathbf{x}) = \nabla^2_{xx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla^2_{xy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\left(\nabla^2_{yy} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right)^{-1}\nabla^2_{yx} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ and we obtain $\mathbf{y}_t \approx \mathbf{y}^*(\mathbf{x}_t)$ by AGD. Then we can bound the approximation error of $\mathbf{H}_t$ as follows

$$\begin{aligned}
&\left\|\nabla^2_{xx} P(\mathbf{x}_t) - \left(\nabla^2_{xx} f(\mathbf{x}_t, \mathbf{y}_t) + \nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(\frac{c_0}{4\ell}I + \frac{1}{2\ell}\sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)\right)\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right)\right\|_2 \\
\leq&\left\|\nabla^2_{xx} P(\mathbf{x}_t) - \left(\nabla^2_{xx} f(\mathbf{x}_t, \mathbf{y}_t) - \nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1}\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right)\right\|_2 \\
&+\left\|\nabla^2_{xx} f(\mathbf{x}_t, \mathbf{y}_t) - \nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(\nabla^2_{yy} f(\mathbf{x}_t, \mathbf{y}_t)\right)^{-1}\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right. \\
&\quad\left.-\left(\nabla^2_{xx} f(\mathbf{x}_t, \mathbf{y}_t) + \nabla^2_{xy} f(\mathbf{x}_t, \mathbf{y}_t)\left(\frac{c_0}{4\ell}\mathbf{I} + \frac{1}{2\ell}\sum_{k=1}^{K'} c_k \mathbf{T}_k(\mathbf{Z}_t)\right)\nabla^2_{yx} f(\mathbf{x}_t, \mathbf{y}_t)\right)\right\|_2 \\
\leq&3\kappa^2\rho\left\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\right\|_2 + \kappa\ell\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^{K'} \\
\leq&3\rho\kappa^2\sqrt{\kappa+1}\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{K_t/2}\left\|\mathbf{y}_{t-1} - \mathbf{y}^*(\mathbf{x}_t)\right\|_2 + \kappa\ell\left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^{K'}
\end{aligned}$$

where the second inequality is according to $\nabla^2_{xy} f(\mathbf{x}, \cdot)\left(\nabla^2_{yy} f(\mathbf{x}, \cdot)\right)^{-1}\nabla^2_{yx} f(\mathbf{x}, \cdot)$ is $3\kappa^2\rho$-Lipschitz continuous (Lemma 9) and the result of (21); the last step is based on Lemma 5. □

D.3 Proof of Theorem 3

We first introduce some lemmas.

Lemma 13 (Tripuraneni et al. [36, Lemma 7 and Lemma 11])**.** *Suppose problem (6) holds that* $\|\mathbf{g}\|_2 \le L^2/M$ *and* $\|\mathbf{H}\|_2 \le (1+\varepsilon_H)L$ *for some* $\varepsilon_H \le 3/25$*. Running Algorithm 4 with*

$$\eta = \frac{1}{20L}, \quad \sigma = \frac{C_\sigma M^2 \sqrt{\varepsilon^3/M^3}}{4608(4L+\sqrt{M\varepsilon})}$$

and

$$\begin{aligned}\mathcal{K}(\varepsilon,\delta') =& \frac{19200L}{C_\sigma\sqrt{M\varepsilon}}\left[6\log\left(3+\frac{9\sqrt{d_x}}{\delta'}\right)+18\log\left(\frac{6L}{\sqrt{M\varepsilon}}\right)+14\log\left(\frac{48(L+C_H\sqrt{M\varepsilon})}{C_\sigma\sqrt{M\varepsilon}}+\frac{24}{C_\sigma}\right)\right]\\ =&\tilde{\mathcal{O}}\left(\frac{L}{\sqrt{M\varepsilon}}\right)\end{aligned}$$

where $C_\sigma > 0$*. Then with probability* $1-\delta'$*, the output satisfies*

$$\|\hat{\mathbf{s}}\|_2 \le \|\mathbf{s}^*\|_2 + \frac{\sqrt{C_\sigma}}{24}\sqrt{\frac{\varepsilon}{M}}.$$

If we further suppose $\|\mathbf{s}^*\|_2 \ge \frac{1}{2}\sqrt{\varepsilon/M}$*, it holds that*

$$\tilde{m}(\hat{\mathbf{s}}) \le \tilde{m}(\mathbf{s}^*) + \frac{C_\sigma M}{12}\|\mathbf{s}^*\|_2^3 \ \text{ and } \ \Delta = \tilde{m}(\hat{\mathbf{s}}) \le -\frac{1-C_\sigma}{96}\sqrt{\frac{\varepsilon^3}{M}}.$$

Proof. Most of the results in this lemma can be obtain by Lemma 7 and Lemma 11 of Tripuraneni et al. [36] directly. Here, we should the derivation of detailed expression for $\mathcal{K}(\varepsilon,\delta')$. We follow the proof of Tripuraneni et al. [36]'s Lemma 11 to show it. Using notations in this paper, we have $\frac{L'^3}{3\|\mathbf{s}_t^*\|_2^3} \le \bar{\sigma} \le 1$, $\eta = 1/20L$, $L' = \frac{1}{2}\sqrt{\varepsilon/M}$, $\varepsilon' = \frac{C_\sigma M}{24}\|\mathbf{s}_t^*\|_2^3$, $\|\mathbf{s}^*\|_2 \ge \sqrt{\varepsilon/M} = L'$ and $\gamma = -\lambda_{\min}(\mathbf{H})$. The notation $\mathcal{K}(\varepsilon,\delta')$ corresponds to $\mathcal{T}(\varepsilon)$ of Tripuraneni et al. [36]. The optimal condition of cubic sub-problem means $\|\mathbf{s}_t^*\|_2 \le 3L/M$. We have

$$\min\left\{\frac{1}{M\|\mathbf{s}_t^*\|_2-\gamma}, \frac{10\|\mathbf{s}_t^*\|_2^2}{\varepsilon'}\right\} = \min\left\{\frac{1}{M\|\mathbf{s}_t^*\|_2-\gamma}, \frac{240}{C_\sigma M\|\mathbf{s}_t^*\|_2}\right\},$$

then

$$\begin{aligned}&\mathcal{K}(\varepsilon,\delta')\\ =&\frac{1+\bar{\sigma}}{\eta}\min\left\{\frac{1}{M\|\mathbf{s}_t^*\|_2-\gamma}, \frac{10\|\mathbf{s}_t^*\|_2^2}{\varepsilon'}\right\}\left[6\log\left(1+\mathbb{I}_{\{\gamma>0\}}\frac{3\sqrt{d}}{\bar{\sigma}\delta'}\right)+14\log\left(\frac{(\beta+M\|\mathbf{s}_t^*\|_2)\|\mathbf{s}_t^*\|_2^2}{\varepsilon'}\right)\right]\\ \le&\frac{2}{1/20L}\cdot\frac{240}{C_\sigma M\|\mathbf{s}_t^*\|_2}\left[6\log\left(1+\frac{3\sqrt{d}}{\bar{\sigma}\delta'}\right)+14\log\left(\frac{(\beta+M\|\mathbf{s}_t^*\|_2)\|\mathbf{s}_t^*\|_2^2}{\varepsilon'}\right)\right]\\ =&\frac{9600L}{C_\sigma M\|\mathbf{s}_t^*\|_2}\left[6\log\left(3\bar{\sigma}+\frac{9\sqrt{d}}{\delta'}\right)+6\log\left(\frac{1}{3\bar{\sigma}}\right)+14\log\left(\frac{24\beta\|\mathbf{s}_t^*\|_2^2}{C_\sigma M\|\mathbf{s}_t^*\|_2^3}+\frac{24M\|\mathbf{s}_t^*\|_2^3}{C_\sigma M\|\mathbf{s}_t^*\|_2^3}\right)\right]\\ \le&\frac{9600L}{C_\sigma M\frac{1}{2}\sqrt{\varepsilon/M}}\left[6\log\left(3+\frac{9\sqrt{d}}{\delta'}\right)+18\log\left(\frac{\|\mathbf{s}_t^*\|_2}{L'}\right)+14\log\left(\frac{24\beta}{C_\sigma M\frac{1}{2}\sqrt{\varepsilon/M}}+\frac{24}{C_\sigma}\right)\right]\\ \le&\frac{19200L}{C_\sigma\sqrt{M\varepsilon}}\left[6\log\left(3+\frac{9\sqrt{d}}{\delta'}\right)+18\log\left(\frac{6L}{\sqrt{M\varepsilon}}\right)+14\log\left(\frac{48(L+C_H\sqrt{\rho\varepsilon})}{C_\sigma\sqrt{M\varepsilon}}+\frac{24}{C_\sigma}\right)\right]\\ =&\tilde{\mathcal{O}}\left(\frac{L}{\sqrt{M\varepsilon}}\right).\end{aligned}$$

□

The following lemma corresponds to the case of calling Algorithm 4 when $\|\mathbf{g}\|_2 \geq L^2/M$. It extends Lemma 9 of Tripuraneni et al. [36], leading to the result includes term $\|\mathbf{s}_t\|_2$ additionally, which is useful to bound the number of gradient calls from AGD in further analysis.

Lemma 14. *Using the notation of Algorithm 3, we suppose $\|\mathbf{g}_t\|_2 \geq L^2/M$ and Assumption 5 holds with $C_H \leq 1/40$ and $C_g \leq 7/600$ at t-th iteration. Let $\mathbf{s}_t = -R_C\mathbf{g}_t/\|\mathbf{g}_t\|_2$, where*

$$R_C = -\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{M\|\mathbf{g}_t\|_2^2} + \sqrt{\left(\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{M\|\mathbf{g}_t\|_2^2}\right)^2 + \frac{2\|\mathbf{g}_t\|_2}{M}}.$$

Then we have

$$\Delta_t = \tilde{m}_t(\mathbf{s}_t) \leq -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} \quad \textit{and} \quad \frac{M}{24}\|\mathbf{s}_t\|_2^3 \leq -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} + P(\mathbf{x}_t) - P(\mathbf{x}_t + \mathbf{s}_t).$$

Proof. The assumption on $\|\mathbf{g}_t\|_2$ means

$$\begin{aligned} R_C =& -\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{M\|\mathbf{g}_t\|_2^2} + \sqrt{\left(\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{M\|\mathbf{g}_t\|_2^2}\right)^2 + \frac{2\|\mathbf{g}_t\|_2}{M}} \\ \geq& \frac{1}{M}\left(-\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{\|\mathbf{g}_t\|_2^2} + \sqrt{\left(\frac{\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t}{\|\mathbf{g}_t\|_2^2}\right)^2 + 2L^2}\right) \end{aligned}$$

Note that the function $-x + \sqrt{x^2+2}$ is decreasing on $\mathbb{R}$ and we have $\left\|\nabla^2 P(\mathbf{x}_t)\right\|_2 \leq L$. Combining the condition $\left\|\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t\right\|_2 \leq C_H\sqrt{M\varepsilon}$ and $\varepsilon \leq L^2/M$, we obtain $\left\|(\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t)\mathbf{g}_t\right\|_2 \leq C_H L\|\mathbf{g}_t\|_2$ which implies $\mathbf{g}_t^\top \mathbf{H}_t \mathbf{g}_t \leq (1+C_H)L\|\mathbf{g}_t\|_2^2$. Hence, for all $C_H \leq 1/40$, we have

$$\begin{aligned} R_C \geq& \frac{L}{M}\left(-(1+C_H) + \sqrt{(1+C_H)^2+2}\right) \\ \geq& \frac{L}{M}\left(-\left(1+\frac{1}{40}\right) + \sqrt{\left(1+\frac{1}{40}\right)^2+2}\right) \geq \frac{7L}{10M}. \end{aligned} \tag{22}$$

Observe that we have $R_C = \arg\min_{\eta\in\mathbb{R}^+} \tilde{m}_t(-\eta\mathbf{g}_t/\|\mathbf{g}_t\|_2)$, which means the derivative of function $q_t(\eta) = \tilde{m}_t(-\eta\cdot\mathbf{g}_t/\|\mathbf{g}_t\|_2)$ is 0 at $\eta = R_C$. Hence, we have

$$\tilde{m}_t(s_t) = \tilde{m}_t\left(-\frac{R_C\mathbf{g}_t}{\|\mathbf{g}_t\|_2}\right) = -\frac{R_C\|\mathbf{g}_t\|_2}{2} - \frac{MR_C^3}{12} \leq -\frac{7L^3}{20M^2} - \frac{MR_C^3}{12} \leq -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} - \frac{MR_C^3}{12}. \tag{23}$$

Using inequality (9) of Lemma 10, we have

$$\begin{aligned} & P(\mathbf{x}_t+\mathbf{s}_t) - P(\mathbf{x}_t) \\ \leq& \nabla P(\mathbf{x}_t)^\top \mathbf{s}_t + \frac{1}{2}(\mathbf{s}_t)^\top \nabla^2 P(\mathbf{x}_t)\mathbf{s}_t + \frac{M}{6}\|\mathbf{s}_t\|_2^3 \\ =& \mathbf{g}_t^\top \mathbf{s}_t + \frac{1}{2}(\mathbf{s}_t)^\top \mathbf{H}_t\mathbf{s}_t + \frac{M}{6}\|\mathbf{s}_t\|_2^3 + (\nabla P(\mathbf{x}_t) - \mathbf{g}_t)^\top \mathbf{s}_t + \frac{1}{2}(\mathbf{s}_t)^\top(\nabla^2 P(\mathbf{x}_t) - \mathbf{H}_t)\mathbf{s}_t \\ \leq& \tilde{m}_t(\mathbf{s}_t) + C_g\varepsilon\|\mathbf{s}_t\|_2 + \frac{C_H\sqrt{M\varepsilon}}{2}\|\mathbf{s}_t\|_2^2 \\ =& -\frac{R_C\|\mathbf{g}_t\|_2}{2} - \frac{MR_C^3}{12} + C_g\varepsilon\|\mathbf{s}_t\|_2 + \frac{C_H\sqrt{M\varepsilon}}{2}\|\mathbf{s}_t\|_2^2 \\ =& -\frac{R_C\|\mathbf{g}_t\|_2}{2} - \frac{MR_C^3}{24} - \left(\frac{MR_C^3}{24} - C_g\varepsilon\|\mathbf{s}_t\|_2 - \frac{C_H\sqrt{M\varepsilon}}{2}\|\mathbf{s}_t\|_2^2\right) \\ \leq& -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} - \frac{MR_C^3}{24} - \frac{L^2R_C}{24M}\left(\left(\frac{MR_C}{L}\right)^2 - 24C_g - 12C_H\left(\frac{MR_C}{L}\right)\right) \end{aligned} \tag{24}$$

Consider that function $q(x) = x^2 - 24C_g - 12C_H x$ is minimized at $7/10$ on $\{x : x \geq 7/10\}$ when $C_H \leq 1/40$. Since inequality (22) means $MR_C/L \geq 7/10$, we have

$$-\left(\left(\frac{MR_C}{L}\right)^2 - 24C_g - 12C_H\left(\frac{MR_C}{L}\right)\right) \leq -\left(\frac{7}{10}\right)^2 + 24C_g + 12C_H\left(\frac{7}{10}\right) \leq 0$$

when $C_H \leq 1/40$ and $C_g \leq 7/600$. Combining all above results, we have

$$\frac{M}{24}\|\mathbf{s}_t\|_2^3 \leq -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} + P(\mathbf{x}_t) - P(\mathbf{x}_t + \mathbf{s}_t)$$

□

Since we approximate $\nabla P(\mathbf{x}_t)$ and $\nabla^2 P(\mathbf{x}_t)$ by $\mathbf{g}_t$ and $\mathbf{H}_t$ respectively, the procedure of Algorithm 3 can be viewed as solving nonconvex optimization problem $\min_{\mathbf{x}\in\mathbb{R}^{d_x}} P(\mathbf{x})$ by inexact first-order and second-order information. Hence, the following lemma holds for our algorithm.

Lemma 15 (Tripuraneni et al. [36, Claim 2])**.** *Suppose that Assumption 5 holds with $C_g \leq 1/240$ and $C_H \leq 1/200$; and we have $\|\mathbf{g}_t\|_2 \leq L^2/M$ and $\tilde{m}_t(\mathbf{s}_t) \leq -\frac{1-C_\sigma}{96}\sqrt{\varepsilon^3/M}$ with $C_\sigma \leq 1/4$. We have*

$$P(\mathbf{x}_t + \mathbf{s}_t) - P(\mathbf{x}_t) \leq -\frac{M}{32}\|\mathbf{s}_t^*\|_2^3 - \frac{1}{625}\sqrt{\frac{\varepsilon^3}{M}}.$$

Proof. We directly use the proof of Claim 2 of Tripuraneni et al. [36]. The notation of c_1, c_2, c_3 and c_4 of Tripuraneni et al. [36]'s paper corresponds to C_g, C_H, C_σ and $\sqrt{C_\sigma}/24$ here. If $\|\mathbf{s}_t^*\|_2 \geq \frac{1}{2}\sqrt{\varepsilon/M}$, we have

$$\begin{aligned}
&P(\mathbf{x}_{t+1}) - P(\mathbf{x}_t)\\
\leq& -\left(\frac{1 - C_\sigma - 48(C_g + C_H\sqrt{C_\sigma}/24) - 12C_H C_\sigma/576}{12}\right) M\|\mathbf{s}_t\|_2^* + \left(C_g + \frac{C_H\sqrt{C_\sigma}}{48}\right)\frac{C_\sigma}{24}\sqrt{\frac{\varepsilon^3}{M}}\\
\leq& -\left(\frac{1 - 1/4 - 48(1/240 + 1/9600) - 3/576/200}{12}\right) M\|\mathbf{s}_t\|_2^* + \left(\frac{1}{240} + \frac{1/2}{9600}\right)\frac{1}{96}\sqrt{\frac{\varepsilon^3}{M}}\\
\leq& -\frac{9M}{20}\|\mathbf{s}_t^*\|_2^3 + \frac{1}{10000}\sqrt{\frac{\varepsilon^3}{M}} \leq -\frac{2M}{5}\|\mathbf{s}_t^*\|_2^3 - \frac{1}{150}\sqrt{\frac{\varepsilon^3}{M}}.
\end{aligned}$$

If $\|\mathbf{s}_t^*\|_2 \leq \frac{1}{2}\sqrt{\varepsilon/M}$, we have

$$\begin{aligned}
&P(\mathbf{x}_{t+1}) - P(\mathbf{x}_t)\\
\leq& -\frac{1 - C_\sigma}{96}\sqrt{\frac{\varepsilon^3}{M}} + \frac{(C_g + C_H\sqrt{C_\sigma}/24)\varepsilon}{2}\sqrt{\frac{\varepsilon}{M}} + \frac{C_H C_\sigma/576\sqrt{M\varepsilon^3}}{8M} + \left(C_g + \frac{C_H\sqrt{C_\sigma}/24}{2}\right)\frac{\sqrt{C_\sigma}}{24}\sqrt{\frac{\varepsilon^3}{M}}\\
\leq& -\frac{1}{256}\sqrt{\frac{\varepsilon^3}{M}} - \frac{1}{256}\sqrt{\frac{\varepsilon^3}{M}} + \frac{1/240 + 1/9600}{2}\sqrt{\frac{\varepsilon^3}{M}} + \frac{1/800}{4608}\sqrt{\frac{\varepsilon^3}{M}} + \left(1/240 + \frac{1/9600}{2}\right)\frac{1}{48}\sqrt{\frac{\varepsilon^3}{M}}\\
\leq& -\frac{1}{256}\sqrt{\frac{\varepsilon^3}{M}} - \frac{1}{256}\sqrt{\frac{\varepsilon^3}{M}} + \frac{23}{10000}\sqrt{\frac{\varepsilon^3}{M}} \leq -\frac{M}{32}\|\mathbf{s}_t^*\|_2^3 - \frac{1}{625}\sqrt{\frac{\varepsilon^3}{M}}.
\end{aligned}$$

We finish the proof by combing above results. □

We first bound the norm of $\mathbf{s}_t$.

Lemma 16. *Under the setting of Theorem 3, if it satisfies $\Delta_t \leq -\frac{1}{128}\sqrt{\varepsilon^3/M}$, we have*

$$\frac{M}{256}\|\mathbf{s}_t\|_2^3 \leq P(\mathbf{x}_t) - P(\mathbf{x}_t + \mathbf{s}_t) - \frac{1}{626}\sqrt{\frac{\varepsilon^3}{M}}.$$

with probability $1 - \delta'$.

Proof. In the case of $\|\mathbf{g}_t\|_2 \geq L^2/M$, Lemma 14 implies

$$\frac{M}{24}\|\mathbf{s}_t\|_2^3 \leq -\frac{7}{20}\sqrt{\frac{\varepsilon^3}{M}} + P(\mathbf{x}_t) - P(\mathbf{x}_{t+1}). \tag{25}$$

In the case of $\|\mathbf{g}_t\|_2 \leq L^2/M$, using the condition $\Delta_t \leq -\frac{1}{128}\sqrt{\varepsilon^3/M}$ and Lemma 15, we have

$$P(\mathbf{x}_t + \mathbf{s}_t) - P(\mathbf{x}_t) \leq -\frac{M}{32}\|\mathbf{s}_t^*\|_2^3 - \frac{1}{625}\sqrt{\frac{\varepsilon^3}{M}}. \tag{26}$$

On the other hand, Lemma 13 implies with probability $1-\delta'$, we have

$$\|\mathbf{s}_t\|_2 \leq \|\mathbf{s}_t^*\|_2 + \frac{\sqrt{C_\sigma}}{24}\sqrt{\frac{\varepsilon}{M}} \quad \Longrightarrow \quad M\|\mathbf{s}_t\|_2^3 \leq 8M\|\mathbf{s}_t^*\|_2^3 + \frac{8C_\sigma^{1.5}M}{24^3}\sqrt{\frac{\varepsilon^3}{M^3}},$$

which means

$$-\frac{M}{32}\|\mathbf{s}_t^*\|_2^3 \leq -\frac{M}{256}\|\mathbf{s}_t\|_2^3 + \frac{1}{3538944}\sqrt{\frac{\varepsilon^3}{M}} \tag{27}$$

Connecting inequalities (26) and (27), we have

$$P(\mathbf{x}_t + \mathbf{s}_t) - P(\mathbf{x}_t) \leq -\frac{M}{256}\|\mathbf{s}_t\|_2^3 - \frac{1}{626}\sqrt{\frac{\varepsilon^3}{M}}. \tag{28}$$

Combining the results of (25) and (28), we prove this lemma. □

Then we give the proof of Theorem 3.

Proof. To bound the number of iterations, we are only interested in the iteration with $\Delta_t \leq -\frac{1}{128}\sqrt{\varepsilon^3/M}$, otherwise the condition in line 7 holds and the algorithm will break the loop. In such case, Lemma 16 mean

$$\frac{M}{256}\|\mathbf{s}_t\|_2^3 \leq P(\mathbf{x}_t) - P(\mathbf{x}_t + \mathbf{s}_t) - \frac{1}{626}\sqrt{\frac{\varepsilon^3}{M}}.$$

Suppose the total number of iteration is T'. Summing over $t = 0, \ldots, T'$, we have

$$T' \leq 626(P(\mathbf{x}_0) - P(\mathbf{x}_T))\sqrt{M}\varepsilon^{-1.5} \leq 626(P(\mathbf{x}_0) - P^*)\sqrt{M}\varepsilon^{-1.5} = \mathcal{O}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right).$$

Lemma 6 says if $\mathbf{x}_t + \mathbf{s}_t^*$ is not an $\left(\varepsilon, \sqrt{M\varepsilon}\right)$-second-order stationary point of $P(\mathbf{x})$, then $\|\mathbf{s}_t^*\|_2 \geq \frac{1}{2}\sqrt{\varepsilon/M}$. Combining Lemma 13 and Lemma 14, we have

$$\Delta_t = \tilde{m}_t(\mathbf{s}_t) \leq \max\left\{-\frac{7}{20}, -\frac{1-C_\sigma}{96}\right\}\sqrt{\frac{\varepsilon^3}{M}} \leq -\frac{1}{128}\sqrt{\frac{\varepsilon^3}{M}}.$$

Hence, if condition in line 7 holds, we conclude that $\mathbf{x}_t + \mathbf{s}_t^*$ is an $\left(\varepsilon, \sqrt{M\varepsilon}\right)$-second-order stationary point and Lemma 8 of Tripuraneni et al. [36] means the output $\hat{\mathbf{x}} = \mathbf{x}_t + \hat{\mathbf{s}}$ is a $(\varepsilon, 2\kappa^{1.5}\sqrt{\rho\varepsilon})$-second-order stationary point of $P(\mathbf{x})$.

Note that Lemma 13 means each iteration of our algorithm succeed with probability $1-\delta'$. Let $\mathcal{F}_t$ be the event that t-th iteration fail. The union bound implies

$$\mathbb{P}\left(\bigcup_{t=1}^{T}\mathcal{F}_t\right) \leq \sum_{t=1}^{T}\mathbb{P}\left(\mathcal{F}_t\right) = T\delta' = \delta.$$

Hence, the probability of success is at least $1-\delta$. □

D.4 Proof of Theorem 4

Proof. We first use induction to show that

$$\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\|_2 \le \tilde{\varepsilon} \tag{29}$$

holds for any $t \ge 0$. For $t = 0$, Lemma 5 directly implies $\|\mathbf{y}_0 - \mathbf{y}^*(\mathbf{x}_0)\|_2 \le \tilde{\varepsilon}$. Suppose it holds that $\|\mathbf{y}_{t-1} - \mathbf{y}^*(x_{t-1})\|_2 \le \tilde{\varepsilon}$ for any $t = t' - 1$, then we have

$$\begin{aligned}
&\|\mathbf{y}_{t'} - \mathbf{y}^*(x_{t'})\|_2 \\
\le&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\|\mathbf{y}_{t'-1} - \mathbf{y}^*(\mathbf{x}_{t'})\|_2 \\
\le&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\|\mathbf{y}_{t'-1} - \mathbf{y}^*(\mathbf{x}_{t'-1})\|_2 + \|\mathbf{y}^*(\mathbf{x}_{t'-1}) - \mathbf{y}^*(\mathbf{x}_{t'})\|_2\right) \\
\le&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\tilde{\varepsilon} + \kappa\|\mathbf{x}_{t'-1} - \mathbf{x}_{t'}\|_2\right) \\
=&\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_{t'}/2}\left(\tilde{\varepsilon} + \kappa\|\mathbf{s}_{t'-1}\|_2\right) \le \tilde{\varepsilon},
\end{aligned} \tag{30}$$

where the first inequality is based on Lemma 5; the second one use triangle inequality; the third one is based on induction hypothesis and the last step use the definition of K_t and $\tilde{\varepsilon}$.

Combining inequality (29) with Lemma 1 and Assumption 1, we obtain

$$\|\mathbf{g}_t - \nabla P(\mathbf{x}_t)\|_2 = \|\nabla_x f(\mathbf{x}_t, \mathbf{y}_t) - \nabla_x f(\mathbf{x}_t, \mathbf{y}^*(\mathbf{x}_t))\|_2 \le \ell\|\mathbf{y}_t - \mathbf{y}^*(\mathbf{x}_t)\|_2 \le C_g\varepsilon.$$

Combining inequality (29) with Lemma 8 and Assumption 2, we have

$$\begin{aligned}
&\left\|\nabla^2_{xx}P(\mathbf{x}_t) - \mathbf{H}_t\right\|_2 \\
\le&3\rho\kappa^2\sqrt{\kappa+1}\left(1-\frac{1}{\sqrt{\kappa}}\right)^{K_t/2}\|\mathbf{y}_{t-1} - \mathbf{y}^*(\mathbf{x}_t)\|_2 + \kappa\ell\left(1-\frac{2}{\sqrt{\kappa}+1}\right)^{K'} \\
\le&3\rho\kappa^2\tilde{\varepsilon} + \frac{C_H\sqrt{M\varepsilon}}{2} \le C_H\sqrt{M\varepsilon}.
\end{aligned}$$

We obtain the upper bound of $\sum_{t=0}^{T} K_t$ by following the analysis of Theorem 2. □

D.5 Proof of Corollary 3

Proof. The output is a desired second-order-stationary point can be proved by directly combining Theorem 3 and Theorem 4. Connecting the upper bounds of $\sum_{t=1}^{T} K_t$ and Theorem 4 , we conclude the total number of gradient calls is at most

$$\begin{aligned}
&\sum_{t=1}^{T} K_t \\
\le&\frac{2\sqrt{\kappa}T}{3}\log\left(\frac{3}{T}\log\left(\frac{\sqrt{\kappa+1}}{\tilde{\varepsilon}}\|\mathbf{y}^*(\mathbf{x}_0)\|_2\right) + 8(\kappa+1)^{1.5} + \frac{2048\kappa^3(\kappa+1)^{1.5}}{626TM\tilde{\varepsilon}'^3}\left(P(\mathbf{x}_0) - P^*\right)\right) \\
=&\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right).
\end{aligned}$$

The total number of Hessian-vector calls from Algorithm 4 is at most

$$\begin{aligned}
&T\cdot\mathcal{K}(\varepsilon,\delta')\cdot K' \\
\le&\tilde{\mathcal{O}}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)\cdot\tilde{\mathcal{O}}\left(\frac{L}{\sqrt{M\varepsilon}}\right)\cdot\tilde{\mathcal{O}}\left(\sqrt{\kappa}\right) \\
=&\tilde{\mathcal{O}}\left(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5}\right)\cdot\tilde{\mathcal{O}}\left(\frac{\kappa\ell}{\kappa^{1.5}\sqrt{\rho\varepsilon}}\right)\cdot\tilde{\mathcal{O}}\left(\sqrt{\kappa}\right)
\end{aligned}$$

$$=\tilde{\mathcal{O}}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right).$$

Using Lemma 8 of Tripuraneni et al. [36], we know the total number of Hessian-vector calls from Algorithm 5 is at most

$$\tilde{\mathcal{O}}\left(\sqrt{\kappa}\right)\cdot\mathcal{O}\left(\frac{L^2}{M\varepsilon}\right)=\tilde{\mathcal{O}}\left(\frac{\ell^2}{\sqrt{\kappa}\rho\varepsilon}\right),$$

which is not the leading term in total complexity for small ε. □

E Implementation of IMCN

In the implementation of IMCN (the cubic sub-problem solver), all of steps related to $\mathbf{H}_t$ can be view as computing Hessian-vector product of the form

$$\mathbf{H}_t\mathbf{u}'=\left(\nabla^2_{xx}f(\mathbf{x}_t,\mathbf{y}_t)+\nabla^2_{xy}f(\mathbf{x}_t,\mathbf{y}_t)\left(\frac{c_0}{4\ell}\mathbf{I}+\frac{1}{2\ell}\sum_{k=1}^{K'}c_k\mathbf{T}_k(\mathbf{Z}_t)\right)\nabla^2_{yx}f(\mathbf{x}_t,\mathbf{y}_t)\right)\mathbf{u}' \quad (31)$$

where $\mathbf{u}'\in\mathbb{R}^{d_x}$ is given. Recall that Chebyshev polynomial satisfies $\mathbf{T}_k(\mathbf{Z}_t)=2\mathbf{Z}_t\mathbf{T}_{k-1}(\mathbf{Z}_t)-\mathbf{T}_{k-2}(\mathbf{Z}_t)$, which allow us computing (31) without constructing any Hessian matrix explicitly. Concretely, For fixed t, we define

$$\mathbf{u}=\nabla^2_{yx}f(\mathbf{x}_t,\mathbf{y}_t)\mathbf{u}',\quad \mathbf{B}_k=\frac{c_0}{4\ell}\mathbf{I}+\frac{1}{2\ell}\sum_{k=1}^{K}c_k\mathbf{T}_k(\mathbf{Z}_t),\quad \mathbf{v}_k=\mathbf{T}_k(\mathbf{Z}_t)\mathbf{u}\quad\text{and}\quad \mathbf{w}_k=\mathbf{B}_k\mathbf{u}.$$

Since we have $\mathbf{Z}_t=\frac{4\ell}{\ell-\mu}\left(-\frac{1}{2\ell}\nabla^2_{yy}f(\mathbf{x}_t,\mathbf{y}_t)-\frac{\ell+\mu}{4\ell}\mathbf{I}\right)$, it holds that

$$\begin{aligned}\mathbf{w}_k=&\left(\mathbf{B}_{k-1}+c_k\mathbf{T}_k(\mathbf{Z}_t)\right)\mathbf{u}\\=&\mathbf{w}_{k-1}+c_k\left(2\mathbf{Z}_t\mathbf{T}_{k-1}(\mathbf{Z}_t)-\mathbf{T}_{k-2}(\mathbf{Z}_t)\right)\mathbf{u}\\=&\mathbf{w}_{k-1}+c_k\left(\frac{8\ell}{\ell-\mu}\left(-\frac{1}{2\ell}\nabla^2_{yy}f(\mathbf{x}_t,\mathbf{y}_t)-\frac{\ell+\mu}{4\ell}\mathbf{I}\right)\mathbf{v}_{k-1}-\mathbf{v}_{k-2}\right)\\=&\mathbf{w}_{k-1}-\frac{4c_k}{\ell-\mu}\nabla^2_{yy}f(\mathbf{x}_t,\mathbf{y}_t)\mathbf{v}_{k-1}-\frac{2c_k(\ell+\mu)}{\ell-\mu}\mathbf{v}_{k-1}-c_k\mathbf{v}_{k-2}\end{aligned} \quad (32)$$

and

$$\begin{aligned}\mathbf{v}_k=&\mathbf{T}_k(\mathbf{Z}_t)\mathbf{u}\\=&2\mathbf{Z}_t\mathbf{T}_{k-1}(\mathbf{Z}_t)\mathbf{u}-\mathbf{T}_{k-2}(\mathbf{Z}_t)\mathbf{u}\\=&\frac{8\ell}{\ell-\mu}\left(-\frac{1}{2\ell}\nabla^2_{yy}f(\mathbf{x}_t,\mathbf{y}_t)-\frac{\ell+\mu}{4\ell}\mathbf{I}\right)\mathbf{v}_{k-1}-\mathbf{v}_{k-2}\\=&-\frac{4}{\ell-\mu}\nabla^2_{yy}f(\mathbf{x}_t,\mathbf{y}_t)\mathbf{v}_{k-1}-\frac{2(\ell+\mu)}{\ell-\mu}\mathbf{v}_{k-1}-\mathbf{v}_{k-2}\end{aligned} \quad (33)$$

Hence, based on update rules (32) and (33), we can obtain $\mathbf{H}_t\mathbf{u}'$ with $\mathcal{O}(K')=\tilde{\mathcal{O}}(\sqrt{\kappa})$ Hessian-vector calls. Additionally, this strategy avoids $\mathcal{O}\left(d_x^2+d_y^2\right)$ space to keep Hessian matrices.

F Complexity Comparison

Table 1 summarizes the theoretical results of proposed algorithms and existing methods. Note that the original analysis of GDA [20] and PPA [21] are based on the variable $\mathbf{y}$ lies in a convex and compact constraint set. In fact, both of these analysis can be extended to unconstrained case easily. We give a brief sketch for the modification as follows.

GDA Consider the proof in Section C.3 of Lin et al. [20]. We only needs to keep the term δ_0 and do not relax it into the diameter of the constraint set. Then it achieves $\mathcal{O}\left((\kappa^2\ell+\kappa\ell^2)\varepsilon^{-2}\right)$ gradient call upper bound for unconstrained case.

Table 1: We present the comparison of the algorithms for nonconvex-strongly-concave minimax optimization. Note that only the proposed MCN and IMCN guarantee to find the approximate second-order stationary point.

Algorithm	GDA [20]	PPA [21]	MCN (Corollary 1)	IMCN (Corollary 3)
$\|\nabla P(\mathbf{x})\|_2 \leq \varepsilon$	✓	✓	✓	✓
$\nabla^2 P(\mathbf{x}) \succeq -\kappa^{1.5}\sqrt{\rho\varepsilon}\,\mathbf{I}$	✗	✗	✓	✓
first-order oracle	$\mathcal{O}\left((\kappa^2\ell + \kappa\ell^2)\varepsilon^{-2}\right)$	$\tilde{\mathcal{O}}\left(\sqrt{\kappa}\ell\varepsilon^{-2}\right)$	$\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$	$\tilde{\mathcal{O}}\left(\kappa^2\sqrt{\rho}\varepsilon^{-1.5}\right)$
second-order oracle	–	–	$\mathcal{O}(\kappa^{1.5}\sqrt{\rho}\varepsilon^{-1.5})$	–
Hessian-vector oracle	–	–	–	$\tilde{\mathcal{O}}\left(\kappa^{1.5}\ell\varepsilon^{-2}\right)$

PPA Recall the proof in Section D.3 of Lin et al. [21]. Let $\delta = \mathcal{O}\left(\varepsilon^2/(\kappa^2\ell)\right)$. To achieve the corresponding result for unconstrained case in Table 1, we only needs to find $\mathbf{x}_{t+1}$ such that

$$\max_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{x}_{t+1},\mathbf{y}) + \ell\left\|\mathbf{x}_{t+1}-\mathbf{x}_t\right\|_2^2 \leq \min_{\mathbf{x}\in\mathbb{R}^x}\left\{\max_{\mathbf{y}\in\mathbb{R}^y} f(\mathbf{x},\mathbf{y}) + \ell\left\|\mathbf{x}-\mathbf{x}_t\right\|_2^2\right\} + \delta$$

in $\tilde{\mathcal{O}}(\sqrt{\kappa}\log(1/\varepsilon))$ gradient calls, which can be obtained by AL-SVRE with $n = 1$ [23]. More specifically, we require using AL-SVRE with $n = 1$ to solve the following minimax problem

$$\min_{\mathbf{x}\in\mathbb{R}^x}\max_{\mathbf{y}\in\mathbb{R}^y} f_t(\mathbf{x},\mathbf{y}) \triangleq f(\mathbf{x},\mathbf{y}) + \ell\left\|\mathbf{x}-\mathbf{x}_t\right\|_2^2. \tag{34}$$

Since we have supposed $f(\mathbf{x},\mathbf{y})$ is ℓ-smooth and μ-strongly-concave in $\mathbf{y}$, the function $f_t(\mathbf{x},\mathbf{y})$ is 3ℓ-smooth, ℓ-strongly-convex in $\mathbf{x}$ and μ-strongly-concave in $\mathbf{y}$. Then the condition numbers of $f_t(\mathbf{x},\mathbf{y})$ are $\kappa_x = 3\ell/\ell = 3$ and $\kappa_y = 3\ell/\mu = 3\kappa$ respectively. Note that the proof of Corollary 1 of [23] does not depend on that $\mathbf{x}$ or $\mathbf{y}$ is constrained in a bounded set. Hence, applying it with $n = 1$ means we need at most $\tilde{\mathcal{O}}\left(\sqrt{\kappa}\log(1/\varepsilon)\right)$ gradient calls to solve problem (34) with desired accuracy.

G Experimental Details

In this section, we present the details of both synthetic minimax problem and the setting of domain adaptation experiments.

G.1 Details of Synthetic Minimax Problem

Our synthetic experiment are based on the following nonconvex-strongly-concave minimax problem

$$\min_{\mathbf{x}\in\mathbb{R}^3}\max_{\mathbf{y}\in\mathbb{R}^2} f(\mathbf{x},\mathbf{y}) = w(x_3) - \frac{y_1^2}{40} + x_1y_1 - \frac{5y_2^2}{2} + x_2y_2$$

where $\mathbf{x} = [x_1, x_2, x_3]^\top$, $\mathbf{y} = [y_1, y_2]^\top$ and

$$w(x) = \begin{cases} \sqrt{\epsilon}\left(x + (L+1)\sqrt{\epsilon}\right)^2 - \frac{1}{3}\left(x + (L+1)\sqrt{\epsilon}\right)^3 - \frac{1}{3}(3L+1)\epsilon^{3/2}, & x \leq -L\sqrt{\epsilon}; \\ \epsilon x + \frac{\epsilon^{3/2}}{3}, & -L\sqrt{\epsilon} < x \leq -\sqrt{\epsilon}; \\ -\sqrt{\epsilon}x^2 - \frac{x^3}{3}, & -\sqrt{\epsilon} < x \leq 0; \\ -\sqrt{\epsilon}x^2 + \frac{x^3}{3}, & 0 < x \leq \sqrt{\epsilon}; \\ -\epsilon x + \frac{\epsilon^{3/2}}{3}, & \sqrt{\epsilon} < x \leq L\sqrt{\epsilon}; \\ \sqrt{\epsilon}\left(x - (L+1)\sqrt{\epsilon}\right)^2 + \frac{1}{3}\left(x - (L+1)\sqrt{\epsilon}\right)^3 - \frac{1}{3}(3L+1)\epsilon^{3/2}, & L\sqrt{\epsilon} \leq x. \end{cases}$$

The nonconvex function $w(x)$ is designed by Tripuraneni et al. [36] and we set $\epsilon = 0.01$ and $L = 5$ in our experiment. We can verify that

$$\mathbf{y}^*(\mathbf{x}) = \begin{bmatrix} 20x_1 \\ \frac{x_2}{5} \end{bmatrix} \quad \text{and} \quad P(x) = w(x_3) + 10x_1^2 + \frac{1}{10}x_2^2.$$

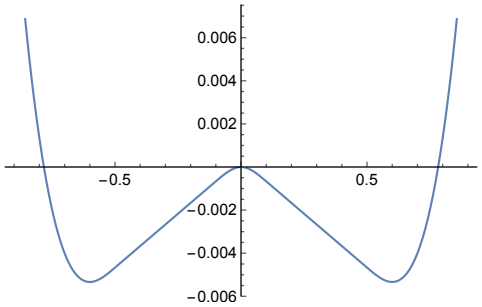


Figure 3: Tripuraneni et al. [36] designed the W-shaped function and provided this figure to visualize it.

G.2 Domain Adaptation

The experiments are conducted on a workstation with Intel Xeon 2.6GHz CPU, 256GB memory and one Nvidia Tesla V100 GPU. We implement the algorithms by using Pytorch 1.10.1 and Python 3.8.8. We choose $\alpha = 1$ and $\lambda = 0.2$ for the model.

The learning rate of GDA and AGD steps in IMCN is selected from $\{c \cdot 10^{-i} : c \in \{1, 5\}, i \in \{1, 2, 3, 4, 5\}\}$. In the implementation of the GDA, the learning rate of $\mathbf{x}$ and $\mathbf{y}$ are chosen separately. For IMCN algorithm, the parameter M is selected from $\{c \cdot 10^{-i} : c \in \{2, 5\}, i \in \{0, 1, 2\}\}$. We set the threshold of using Cauchy point (i.e., Line 2 of Algorithm 4) as 0.01. The number of AGD iterations and Cubic-Solver iterations are selected from $\{50, 100, 150, 200\}$. The learning rate of cubic solver is selected from $\{0.2, 0.02, 0.002\}$.