Finding Local Minimax Points via (Stochastic) Cubic-Regularized GDA: Global Convergence and Complexity

Ziyi Chen\textsuperscript{1}, Qunwei Li\textsuperscript{2}, and Yi Zhou\textsuperscript{1}

\textsuperscript{1}Department of Electrical & Computer Engineering, University of Utah, \{u1276972, yi.zhou\}@syr.edu

\textsuperscript{2}Ant Group, qunwei.qw@antgroup.com

July 12, 2022

Abstract

Gradient descent-ascent (GDA) is a widely used algorithm for minimax optimization. However, GDA has been proved to converge to stationary points for nonconvex minimax optimization, which are suboptimal compared with local minimax points. In this work, we develop GDA-type algorithms that globally converge to local minimax points in nonconvex-strongly-concave minimax optimization. We first show that local minimax points are equivalent to second-order stationary points of a certain envelope function. Then, inspired by the classic cubic regularization algorithm, we propose Cubic-GDA—a cubic-regularized GDA algorithm for finding local minimax points, and provide a comprehensive convergence analysis by leveraging its intrinsic potential function. Specifically, we establish the global convergence of Cubic-GDA to a local minimax point at a sublinear convergence rate. Moreover, we propose a stochastic variant of Cubic-GDA for large-scale minimax optimization, and characterize its sample complexity under stochastic sub-sampling.

1 Introduction

Minimax optimization (a.k.a. two-player sequential zero-sum games) is a popular modeling framework that has broad applications in modern machine learning, including game theory [Ferreira et al., 2012], generative adversarial networks [Goodfellow et al., 2014], adversarial training [Sinha et al., 2017], reinforcement learning [Qu et al., 2020], Ho and Ermon, 2016, Song et al., 2018, etc. A standard minimax optimization problem is shown below, where $f$ is a smooth bivariate function.

\[
\min_{x \in \mathbb{R}^m} \max_{y \in \mathbb{R}^n} f(x, y).
\] (P)

In the existing literature, many optimization algorithms have been developed to solve different types of minimax problems. Among them, a simple and popular algorithm is the gradient descent-ascent (GDA), which alternates between a gradient descent update on $x$ and a gradient ascent update on $y$ in each iteration. Specifically, the global convergence of GDA has been established for minimax problems under various types of global geometries, such as convex-concave-type geometry ($f$ is convex in $x$ and concave in $y$) [Nedić and Ozdaglar, 2009, Du and Hu, 2019, Mokhtari et al., 2020, Zhang and Wang, 2021], bi-linear geometry [Neumann, 1928, Robinson, 1951] and Polyak-Łojasiewicz geometry [Nouiehed et al., 2019, Yang et al., 2020], yet these geometries are not satisfied by general nonconvex minimax problems in modern machine learning applications. Recently, many studies proved the convergence of GDA in nonconvex minimax optimization for both nonconvex-concave problems [Lin et al., 2020, Nouiehed et al., 2019, Xu et al., 2020d] and nonconvex-strongly-concave problems [Lin et al., 2020, Xu et al., 2020d, Chen et al., 2021]. In these studies, it has been shown that GDA converges sublinearly to a stationary point where the gradient of an envelope-type function $\Phi(x) := \max_y f(x, y)$ vanishes.

Although GDA can find stationary points in nonconvex minimax optimization, the stationary points may include candidate solutions that are far more sub-optimal than global minimax points. However, finding global
minimax points is in general NP-hard [Jin et al., 2020]. Recently, [Jin et al., 2020] proposes a notion of local minimax point that is computationally tractable and is close to global minimax point. Specifically, a local minimax point \((x, y)\) is a stationary point that satisfies the following second-order non-degeneracy conditions (see Definition 11 for the formal definition).

\[
\nabla_{22} f(x, y) < 0, \quad \left[ \nabla_{11} f - \nabla_{12} f (\nabla_{22} f)^{-1} \nabla_{21} f \right] (x, y) > 0.
\]

In the existing literature, several studies have proposed Newton-type GDA algorithms for finding such local minimax points. Specifically, [Wang et al., 2020] proposed a Follow-the-Ridge (FR) algorithm, which is a variant of GDA that applies a second-order correction term to the gradient ascent update. In particular, the authors showed that any strictly stable fixed point of FR is a local minimax point, and vice versa. In another work [Zhang et al., 2021], the authors proposed two Newton-type GDA algorithms that are proven to locally converge to a local minimax point at a linear and super-linear convergence rate, respectively. However, these second-order-type GDA algorithms only have asymptotic convergence guarantees that require initializing sufficiently close to a local minimax point, and they do not have any global convergence guarantees. Therefore, we are motivated to ask the following fundamental questions.

\(\text{Q: Can we develop globally convergent GDA-type algorithms that can efficiently find local minimax points in nonconvex minimax optimization? What are their convergence rates and complexities?}\)

In this work, we provide comprehensive answers to these questions. We develop deterministic and stochastic Newton-type GDA algorithms that globally converge to local minimax points in nonconvex-strongly-concave minimax optimization, and study their convergence rates and sample complexities under standard assumptions. We summarize our contributions as follows.

### 1.1 Our Contributions

We consider the standard minimax optimization problem \((P)\), where \(f\) is twice-differentiable with Lipschitz continuous gradient and Hessian, and is nonconvex-strongly-concave. In this setting, we first show that local minimax points of \(f\) are equivalent to second-order stationary points of the envelope function \(\Phi_1(x) := \max_{y \in \mathbb{R}^n} f(x, y)\). Then, inspired by the classic cubic regularization algorithm, we propose a cubic-regularized GDA (named Cubic-GDA) algorithm to find local minimax points. The algorithm uses gradient ascent to update \(y\), which is then used to estimate the gradient and Hessian involved in the cubic regularization update for \(x\) (see Algorithm 1 for more details).

**Global convergence.** We show that Cubic-GDA admits an intrinsic potential function \(H_1\) (see Proposition 3) that monotonically decreases over the iterations. Based on this property, we prove that every limit point of \(\{x_t\}\) generated by Cubic-GDA is a local minimax point. Moreover, to achieve an \(\epsilon\)-accurate local minimax point, Cubic-GDA requires \(O(\kappa^{1.5} \epsilon^{-3})\) number of cubic updates and \(O(\kappa^{2.5} \epsilon^{-3})\) number of gradient ascent updates, where \(\kappa > 1\) denotes the problem condition number.

**Sample complexity.** We further develop a stochastic variant of Cubic-GDA that applies sub-sampling to improve the sample complexity in large-scale minimax optimization. In particular, we adopt time-varying batch sizes in a way such that the induced gradient inexactness and Hessian inexactness are adapted to the optimization increment \(\|x_t - x_{t-1}\|\) in the previous iteration. Consequently, to achieve an \(\epsilon\)-accurate local minimax point, we show that stochastic Cubic-GDA requires querying \(O(\kappa^{1.5} \epsilon^{-7})\) number of gradient samples and \(O(\kappa^{2.5} \epsilon^{-5})\) number of Jacobian samples.

### 1.2 Other Related Work

**Deterministic GDA algorithms:** [Yang et al., 2020] studied an alternating gradient descent-ascent (AGDA) algorithm in which the gradient ascent step uses the current variable \(x_{t+1}\) instead of \(x_t\). [Xu et al., 2020d] studied an alternating gradient projection algorithm which applies \(\ell_2\) regularizer to the local objective function of GDA followed by projection onto the constraint sets. [Daskalakis and Panageas, 2018] [Mokhtari et al., 2020] [Zhang and Wang, 2021] analyzed optimistic gradient descent-ascent (OGDA). [Mokhtari et al., 2020] also studied an extra-gradient algorithm which applies two-step GDA in each iteration. [Nouiehed et al., 2019] studied multi-step GDA where multiple gradient ascent steps are performed, and they also studied the momentum-accelerated version. [Cherukuri et al., 2017] [Daskalakis and Panageas, 2018] [Jin et al., 2020] studied GDA in continuous time dynamics using differential equations. [Adolphs et al., 2019] analyzed a second-order variant of the GDA algorithm. In a concurrent work [Luo and Chen, 2021], the authors proposed a Minimax Cubic-Newton algorithm.
that is similar to our Cubic-GDA but adopts a different output rule. Specifically, in iteration \( t \), they terminate the algorithm based on the current increment \( \|s_t\| \). As a comparison, our Algorithm 1 terminates the algorithm based on \( \|s_t\| + \|s_{t+1}\| \). This output rule is critical to develop the practical and adaptive sub-sampling scheme for our stochastic Cubic-GDA. Moreover, we also develop a stochastic version of Cubic-GDA and analyze its sample complexity.

Stochastic GDA algorithms: [Lin et al., 2020] [Yang et al., 2020] analyzed stochastic GDA and stochastic AGDA, which are direct extension of GDA and AGDA to the stochastic setting. Variance reduction techniques have been applied to stochastic minimax optimization, including SVRG-based [Du and Hu, 2019] [Yang et al., 2020], SPIDER-based [Xu et al., 2020], SREDA [Xu et al., 2020b], STORM [Qu et al., 2020] and its gradient free version [Huang et al., 2020]. [Xie et al., 2020] studied the complexity lower bound of first-order stochastic algorithms for finite-sum minimax problem.

Cubic regularization (CR): The CR algorithm dates back to [Griewank, 1981], where global convergence is established. In [Nesterov and Polyak, 2006], the author analyzed the convergence rate of CR to second-order stationary points in nonconvex optimization. In [Nesterov, 2008], the authors established the sub-linear convergence of CR for solving convex smooth problems, and they further proposed an accelerated version of CR with improved sub-linear convergence. [Yue et al., 2019] studied the asymptotic convergence properties of CR under the error bound condition, and established the quadratic convergence of the iterates. Recently, [Hallak and Teboulle, 2020] proposed a framework of two directional method for finding second-order stationary points in general smooth nonconvex optimization. The main idea is to search for a feasible direction toward the solution and is not based on cubic regularization. Several other works proposed different methods to solve the cubic subproblem of CR, e.g., [Agarwal et al., 2017] [Carmon and Duchi, 2016] [Cartis et al., 2011b]. Another line of work aimed at improving the computation efficiency of CR by solving the cubic subproblem with inexact gradient and Hessian information. In particular, [Chadimi et al., 2017] proposed an inexact CR for solving convex problem. Also, [Cartis et al., 2011a] proposed an adaptive inexact CR for nonconvex optimization, whereas [Jiang et al., 2017] further studied the accelerated version for convex optimization. Several studies explored subsampling schemes to implement inexact CR algorithms, e.g., [Kohler and Lucchi, 2017] [Xu et al., 2020a] [Zhou and Liang, 2018] [Wang et al., 2018b].

Notation: For a bivariate function \( f(x, y) \), we denote \( \nabla_1 f, \nabla_2 f \) as the partial gradients with respect to the first and the second input arguments of \( f \), respectively. We also denote the Jacobian blocks with regard to the two input arguments as \( \nabla_{11} f, \nabla_{22} f, \nabla_{12} f, \nabla_{21} f \). We denote the max and min operators as \( \lor \) and \( \land \), respectively.

2 Problem Formulation and Preliminaries

We consider the following standard minimax optimization problem (P), where \( f \) is a nonconvex-strongly-concave bivariate function and is twice-differentiable. Throughout the paper, we define the envelope function \( \Phi(x) := \max_{y \in \mathbb{R}^n} f(x, y) \).

\[
\min_{x \in \mathbb{R}^m} \max_{y \in \mathbb{R}^n} f(x, y). \tag{P}
\]

Our goal is to develop algorithms that globally converge to local minimax points of (P), which are defined to satisfy the following set of conditions [Jin et al., 2020].

**Definition 1 (Local minimax point).** A point \((x, y)\) is a local minimax point of (P) if it satisfies

1. **Stationary:** \( \nabla_1 f(x, y) = 0, \nabla_2 f(x, y) = 0 \);
2. **Non-degeneracy:** \( \nabla_{22} f(x, y) \prec 0 \), and \( \nabla_{11} f - \nabla_{12} f (\nabla_{22} f)^{-1} \nabla_{21} f \) \( (x, y) \succ 0 \).

This definition has been proved to be a second-order sufficient condition of the local minimax point defined in [Jin et al., 2020]. Local minimax points are different from global minimax points, which require \( x \) and \( y \) to be the global minimizer and global maximizer of the functions \( \Phi(x) \) and \( f(x, \cdot) \), respectively. In general minimax optimization, global minimax points can be neither local minimax points nor even stationary points [Jin et al., 2020]. However, it has been shown that global minimax necessarily implies local minimax in nonconvex-strongly-concave optimization [Jin et al., 2020]. Moreover, under mild conditions, many machine learning problems have been shown to possess local minimax points, e.g., generative adversarial networks (GANs) [Nagarajan and Kolter, 2017] [Zhang et al., 2021], distributional robust machine learning [Sinha et al., 2018], etc.

Throughout the paper, we adopt the following standard assumptions on the minimax optimization problem (P) [Lin et al., 2020] [Jin et al., 2020] [Zhang et al., 2021].
**Assumption 1.** The minimax problem \((P)\) satisfies:
1. Function \(f(\cdot, \cdot)\) is \(L_1\)-smooth and function \(f(x, \cdot)\) is \(\mu\)-strongly concave for any fixed \(x\);
2. The Jacobian mappings \(\nabla_{11} f, \nabla_{12} f, \nabla_{21} f, \nabla_{22} f\) are \(L_2\)-Lipschitz continuous;
3. Function \(\Phi(x) := \max_{y\in\mathbb{R}^n} f(x, y)\) is bounded below and has compact sub-level sets.

To elaborate, item 1 considers the class of nonconvex-strongly-concave functions \(f\) that has been widely studied in the minimax optimization literature [Lin et al., 2020] [Jin et al., 2020] [Xu et al., 2020a] [Lu et al., 2020], and it is also satisfied by many machine learning applications. Item 2 assumes that the block Jacobian matrices of \(f\) are Lipschitz continuous, which is a standard assumption for analyzing second-order optimization algorithms [Nesterov and Polyak, 2006] [Agarwal et al., 2017]. Moreover, item 3 guarantees that the minimax problem has at least one solution.

Under Assumption 1, the following properties regarding the gradient of the minimax problem \((P)\) have been proved in the literature. Throughout, we denote \(\kappa = L_1/\mu\) as the condition number.

**Proposition 1.** [Lin et al., 2020] Let Assumption 1 hold. Then, the mapping \(y^*(x) := \arg\max_{y\in\mathbb{R}^n} f(x, y)\) is unique for every fixed \(x\). Moreover, it holds that
1. Function \(\Phi(x)\) is \(L_1(1 + \kappa)\)-smooth and \(\nabla\Phi(x) = \nabla_1 f(x, y^*(x))\).

## 3 Cubic-Regularized GDA

In this section, we propose a GDA-type algorithm that leverages the cubic regularization technique to find local minimax points of the nonconvex minimax problem \((P)\). We first relate local minimax points to certain second-order stationary points in Section 3.1, based on which we further develop the algorithm in Section 3.2.

### 3.1 Local Minimax and Second-Order Stationary

Regarding the conditions of local minimax points listed in Definition 1, note that the stationary conditions in item 1 are easy to achieve, e.g., by performing standard GDA updates. For the non-degeneracy conditions listed in item 2, the first condition is guaranteed as \(f(x, \cdot)\) is strongly concave. Therefore, the major challenge is to achieve the other non-degeneracy condition \([\nabla_{11} f - \nabla_{12} f(\nabla_{22} f)^{-1}\nabla_{21} f](x, y) \succ 0\). Interestingly, in nonconvex-strongly-concave minimax optimization, such a non-degeneracy condition has close connections to a certain second-order stationary condition on the envelope function \(\Phi(x)\), as formally stated in the following proposition.

**Proposition 2.** Let Assumption 1 hold. Then, the following statements hold.
1. Define \(G(x, y) := [\nabla_{11} f - \nabla_{12} f(\nabla_{22} f)^{-1}\nabla_{21} f](x, y)\). Then, \(G\) is a Lipschitz continuous mapping with constant \(L_G = L_2(1 + \kappa)^2\);
2. The Hessian of \(\Phi\) satisfies \(\nabla^2\Phi(x) = G(x, y^*(x))\), and it is Lipschitz continuous with constant \(L_{\Phi} = L_G(1 + \kappa)^3\).

The above proposition points out that the non-degeneracy condition \(G(x, y) \succ 0\) actually corresponds to a second-order stationary condition of the envelop function \(\Phi(x)\). To explain more specifically, consider a pair of points \((x, y^*(x))\), in which \(y^*(x) := \arg\max_y f(x, y)\). Since \(f(x, \cdot)\) is strongly concave and \(y^*(x)\) is the maximizer, we know that \(y^*(x)\) must satisfy the stationary condition \(\nabla_2 f(x, y^*(x)) = 0\) and the non-degeneracy condition \(\nabla_{22} f(x, y^*(x)) \succ 0\). Therefore, in order to be a local minimax point, \(x\) must satisfy the stationary condition \(\nabla_1 f(x, y^*(x)) = 0\) and the non-degeneracy condition \(G(x, y^*(x)) \succ 0\), which, by item 2 of Proposition 1 and item 2 of Proposition 2, are equivalent to the set of second-order stationary conditions stated in the following fact.

**Fact 1.** Let Assumption 1 hold. Then, \((x, y^*(x))\) is a local minimax point of \((P)\) if and only if \(x\) satisfies the following set of second-order stationary conditions

\[(\text{Second-order stationary}): \quad \nabla\Phi(x) = 0, \quad \nabla^2\Phi(x) \succ 0.\]
To summarize, in nonconvex-strongly-concave minimax optimization, finding a local minimax point is equivalent to finding a second-order stationary point of the smooth nonconvex envelope function $\Phi(x)$. Such a key observation is the basis for developing our proposed algorithm in the next subsection.

We also note that the proof of Proposition 2 is not trivial. Specifically, we need to first develop bounds for the spectrum norm of the block Jacobian matrices in Lemma 2 (see the first page of the appendix), which helps prove the Lipschitz continuity of the $G$ mapping in item 1. Moreover, we leverage the optimality condition of $f(x, \cdot)$ to derive an expression for the maximizer mapping $y^*(x)$ (see (14) in the appendix), which is used to further prove item 2.

### 3.2 Cubic-Regularized Gradient Descent-Ascent

The standard GDA algorithm can only find stationary points, i.e., $\nabla \Phi(x) = 0$, in nonconvex-strongly-concave minimax optimization [Lin et al., 2020]. Such a type of convergence guarantee does not rule out the possibility that GDA may get stuck at suboptimal saddle points of the envelope function $\Phi$, which are known to be the major challenge for training high-dimensional machine learning models [Dauphin et al., 2014; Jin et al., 2017; Zhou and Liang, 2018]. Therefore, we are motivated to escape the saddle points and target at finding second-order stationary points. Importantly, the previous Fact 1 shows that these second-order stationary points of $\Phi$ are equivalent to local minimax points in nonconvex-strongly-concave minimax optimization.

In the existing literature, many second-order optimization algorithms have been developed for finding second-order stationary points of nonconvex minimization problems [Nesterov and Polyak, 2006; Agarwal et al., 2017; Yue et al., 2019; Zhou et al., 2018]. Hence, one may want to apply them to minimize the nonconvex function $\Phi(x)$ and find local minimax points of the minimax problem (P). However, these algorithms are not directly applicable, as the function $\Phi(x)$ involves a special maximization structure and hence its specific function form $\Phi$ as well as the gradient $\nabla \Phi$ and Hessian $\nabla^2 \Phi$ are implicit. Instead, our algorithm design can only leverage information of the bi-variate function $f$.

Our algorithm design is inspired by the classic cubic regularization algorithm [Nesterov and Polyak, 2006]. Specifically, to find a second-order stationary point of the envelope function $\Phi(x)$, the conventional cubic regularization algorithm would perform the following iterative update.

$$\begin{align*}
s_{t+1} &\in \arg \min_s \nabla \Phi(x_t)^\top s + \frac{1}{2} s^\top \nabla^2 \Phi(x_t) s + \frac{1}{6 \eta_t} \|s\|^3, \\
x_{t+1} &= x_t + s_{t+1},
\end{align*}$$

where $\eta_t > 0$ is a proper learning rate. However, due to the special maximization structure of $\Phi$, its gradient and Hessian have complex formulas (see Propositions 1 and 2) that involve the mapping $y^*(x)$, which cannot be computed exactly in practice. Hence, we aim to develop an algorithm that efficiently computes approximations of $\nabla \Phi(x), \nabla^2 \Phi(x)$, and use them to perform the cubic regularization update.

To perform the cubic regularization update in eq. (1), we need to compute $\nabla \Phi(x_t) = \nabla_1 f(x_t, y^*(x_t))$ (by Proposition 1) and $\nabla^2 \Phi(x_t) = G(x_t, y^*(x_t))$ (by Proposition 2), both of which depend on the maximizer $y^*(x_t)$ of the function $f(x_t, \cdot)$. Since $f(x_t, \cdot)$ is strongly-concave, we can run $N_t$ iterations of gradient ascent to obtain an approximated maximizer $\tilde{y}_{N_t} \approx y^*(x_t)$, and then approximate $\nabla \Phi(x_t), \nabla^2 \Phi(x_t)$ using $\nabla_1 f(x_t, \tilde{y}_{N_t})$ and $G(x_t, \tilde{y}_{N_t})$, respectively. Intuitively, these are good approximations due to two reasons: (i) $\tilde{y}_{N_t}$ converges to $y^*(x_t)$ at a fast linear convergence rate; and (ii) both $\nabla_1 f$ and $G$ are shown to be Lipschitz continuous in their second argument. We refer to this algorithm as Cubic-Regularized Gradient Descent-Ascent (Cubic-GDA), and summarize its update rule in Algorithm 1 below.

Algorithm 1 terminates whenever the maximum of the previous two increments $\|s_{t-1}\| \vee \|s_t\|$ is below a certain threshold $\epsilon$. Such an output rule helps characterize the computation complexity of the algorithm. Moreover, in Section 5 we leverage this output rule to develop a practical and adaptive sub-sampling scheme for the stochastic version of Cubic-GDA and analyze its sample complexity. We also note that the cubic regularization sub-problem can be efficiently solved by gradient-based algorithms [Carmon and Duchi, 2016; Tripuraneni et al., 2018], which involve computation of Jacobian-vector product that can be efficiently implemented by the existing machine learning platforms such as TensorFlow [Abadi, 2015] and PyTorch [Paszke, 2019]. Please refer to Appendix G for more discussion about how to compute the Jacobian-vector product involved in Cubic-GDA.
**Algorithm 1** Cubic-Regularized GDA (Cubic-GDA)

**Input:** Initialize $x_0, y_0$, learning rates $\eta_x, \eta_y$, threshold $\epsilon'$

Define $\|s_0\| = \epsilon'$

for $t = 0, 1, 2, \ldots, T - 1$ do

Initialize $y_0 = y_t$

for $k = 0, 1, 2, \ldots, N_t - 1$ do

$\hat{y}_{k+1} = \hat{y}_k + \eta_y \nabla_2 f(x_t, \hat{y}_k)$

end

Set $y_{t+1} = \hat{y}_{N_t}$. Solve the cubic problem for $s_{t+1}$:

$\arg\min_s \nabla_1 f(x_t, y_{t+1})^T s + \frac{1}{2} s^T G(x_t, y_{t+1}) s + \frac{1}{6\eta_y} \|s\|^3$

Update $x_{t+1} = x_t + s_{t+1}$

end

**Output:** $x_T, y_T, T' = \min\{t : \|s_{t-1}\| \vee \|s_t\| \leq \epsilon'\}$

### 4 Global Convergence of Cubic-GDA

In this section, we study the global convergence properties of Cubic-GDA. The key to our convergence analysis is characterizing an intrinsic potential function of Cubic-GDA in nonconvex minimax optimization. We formally present this result in the following proposition.

**Proposition 3** (Potential decrease). Let Assumption[3] hold. For any $\alpha, \beta > 0$, choose $\epsilon' \leq \frac{3\alpha}{4L\Phi}$, $\eta_x \leq (9L\Phi + 18\alpha + 28\beta)^{-1}$ and $\eta_y = \frac{2}{L\Phi}$. Define the potential function $H_t := \Phi(x_t) + (L\Phi + 2\alpha + 3\beta)\|s_t\|^3$. Then, when $N_t \geq O(\kappa \ln \frac{L\alpha\|s_{t-1}\| + L\beta\|s_t\|^3}{L\sqrt{\beta} \epsilon'^2})$, the output of Cubic-GDA satisfies the following potential decrease property.

$$H_{t+1} - H_t \leq -(L\Phi + \alpha + \beta)(\|s_{t+1}\|^3 + \|s_t\|^3).$$

(2)

Proposition[3] reveals that Cubic-GDA admits an intrinsic potential function $H_t$, which takes the form of the envelope function $\Phi(x)$ plus the cubic increment term $\|s_t\|^3$. Moreover, the potential function $H_t$ is monotonically decreasing along the optimization path of Cubic-GDA, implying that the algorithm continuously makes optimization progress.

The key for establishing such a potential function is that, by running a sufficient number of inner gradient ascent iterations, we can obtain a sufficiently accurate approximated maximizer $y_{t+1} \approx y^*(x_t)$. Consequently, the $\nabla_1 f(x_t, y_{t+1})$ and $G(x_t, y_{t+1})$ involved in the cubic sub-problem are good approximations of $\nabla \Phi(x_t)$ and $\nabla^2 \Phi(x_t)$, respectively. In fact, the approximation errors are proven to satisfy the following bounds.

$$\|\nabla \Phi(x_t) - \nabla_1 f(x_t, y_{t+1})\| \leq \beta(\|s_t\|^2 + \epsilon), \quad (3)$$

$$\|\nabla^2 \Phi(x_t) - G(x_t, y_{t+1})\| \leq \alpha(\|s_t\|^3 + \epsilon'). \quad (4)$$

On one hand, the above bounds are tight enough to establish the decreasing potential function. On the other hand, they are flexible and are adapted to the increment $\|s_t\| = \|x_t - x_{t-1}\|$ produced by the previous cubic update. Therefore, when the increment is large in the initial iterations, it suffices to use coarse approximations, and hence only a few number of inner gradient ascent iterations are needed. Such an idea of adapting the inexactness to the previous increment in eqs. (3) and (4) are further leveraged to develop a practical and scalable stochastic Cubic-GDA algorithm in Section[5].

Based on Proposition[5] we obtain the following global convergence rate of Cubic-GDA to a second-order stationary point of $\Phi$. Throughout, we adopt the following standard measure of second-order stationary introduced in [Nesterov and Polyak, 2006].

$$\mu(x) = \sqrt{\|\nabla \Phi(x)\| \vee -\frac{\lambda_{\text{min}}(\nabla^2 \Phi(x))}{\sqrt{3SL\Phi}}}. \quad (5)$$

Intuitively, a smaller $\mu(x)$ means that the point $x$ is closer to being second-order stationary.

**Theorem 1** (Global convergence rate). Let the conditions of Proposition[3] hold with $\alpha = \beta = L\Phi$. For any $0 < \epsilon \leq \frac{L\sqrt{3L\Phi}}{L\alpha}$, choose $\epsilon' = \frac{\epsilon}{\sqrt{3L\Phi}}$ and $T \geq \frac{\Phi(x_0) - \Phi^* + 8L\epsilon e^2}{\lambda_{\text{min}}(\nabla^2 \Phi(x_0))}$. Then, the output of Cubic-GDA satisfies:

$$\mu(x_{T'}) \leq \epsilon.$$

(5)
Consequently, the total number of cubic iterations satisfies $T' \leq O(\sqrt{L_2 \kappa^{1.5} \epsilon^{-3}})$, and the total number of gradient ascent iterations satisfies $\sum_{t=0}^{T'-1} N_t \leq \tilde{O}(\sqrt{L_2 \kappa^{2.5} \epsilon^{-3}}).

The above theorem shows that the first-order stationary measure $\|\nabla \Phi(x_t)\|$ converges at a sublinear rate $O(T^{-\frac{1}{2}})$, and the second-order stationary measure $-\lambda_{\min}(\nabla^2 \Phi(x))$ converges at a sublinear rate $O(T^{-\frac{1}{4}})$. Both results match the convergence rates of the cubic regularization algorithm for nonconvex minimization \cite{NesterovPolyak2006}. In comparison, standard GDA cannot guarantee the convergence of $-\lambda_{\min}(\nabla^2 \Phi(x))$, and its convergence rate of $\|\nabla \Phi(x_t)\|$ is $O(T^{-\frac{1}{2}})$ \cite{LinPolyak2020}, which is slower than our Cubic-GDA. Therefore, by leveraging the curvature of the approximated Hessian matrix $G(x_t, y_{t+1})$, Cubic-GDA is able to find second-order stationary points of $\Phi$ at a fast rate.

We note that the proof of the global convergence results in Theorem \ref{thm:global-conv} is critically based on the intrinsic potential function $H$ that we characterized in Proposition \ref{prop:intrinsic}. Specifically, note that the cubic subproblem in Cubic-GDA involves an approximated gradient $\nabla_1 f(x_t, y_{t+1})$ and Hessian matrix $G(x_t, y_{t+1})$. Such inexactness of the gradient and Hessian introduces non-negligible noise to the cubic regularization update of Cubic-GDA. Consequently, Cubic-GDA cannot make monotonic progress on decreasing the function value $\Phi$, as opposed to the standard cubic regularization algorithm in nonconvex minimization (which uses exact gradient and Hessian). Instead, we take a different approach and show that as long as the gradient and Hessian approximations are involved in the cubic update by their corresponding stochastic approximations.

On the other hand, to approximate the matrix $G$, we sub-sample mini-batches of samples $B_{11}, B_{12}, B_{21}, B_{22}$ with replacement and construct approximated Jacobian matrices $\tilde{\nabla}_{11} f, \tilde{\nabla}_{12} f, \tilde{\nabla}_{21} f, \tilde{\nabla}_{22} f$ in the same way as above. Then, we construct the following approximation of $G$.

$$
\tilde{G}(x, y) = \left[ \tilde{\nabla}_{11} f - \tilde{\nabla}_{12} f (\tilde{\nabla}_{22} f)^{-1} \tilde{\nabla}_{21} f \right](x, y). \tag{7}
$$

We summarize the update rule of stochastic Cubic-GDA in Algorithm \ref{alg:stoc-cubic-gda} below. In particular, we run stochastic gradient ascent in the inner iterations to obtain the approximated maximizer $y_{t+1}$, and its high-probability convergence rate has been established in the stochastic optimization literature \cite{HarveyPolyakWen2019}.
The following lemma characterizes the sample complexities of all the stochastic approximators for achieving a certain approximation accuracy.

**Lemma 1.** Fix any $0 < \epsilon_1 \leq 2L_0$, $0 < \epsilon_2 \leq 4L_1$ and choose the following batch sizes

$$|B_1| \geq \mathcal{O}\left(\frac{L_0^2}{\epsilon_1^2} \ln \frac{m}{\delta}\right), \quad (8)$$

$$|B_{11}|, |B_{12}|, |B_{21}|, |B_{22}| \geq \mathcal{O}\left(\frac{L_0^2}{\epsilon_2^2} \ln \frac{m+n}{\delta}\right). \quad (9)$$

Then, the stochastic approximators satisfy the following error bounds with probability at least $1 - \delta$.

$$\|\hat{\nabla}_1 f(x, y) - \nabla f(x, y)\| \leq \epsilon_1, \quad (10)$$

$$\|\hat{\nabla}_k^2 f(x, y) - \nabla_k^2 f(x, y)\| \leq \epsilon_2, \quad \forall k, \ell \in \{1, 2\}, \quad (11)$$

$$\|G(x, y) - G(x, y)\| \leq (\kappa + 1)^2 \epsilon_2. \quad (12)$$

Therefore, by choosing proper batch sizes, the inexactness of the stochastic gradient, Jacobian and Hessian estimators can be controlled within a desired range. From this perspective, stochastic Cubic-GDA can be viewed as an inexact version of the Cubic GDA algorithm.

To characterize the convergence and sample complexity of stochastic Cubic-GDA, we adopt an adaptive inexactness criterion for the sub-sampling scheme. Specifically, we choose time-varying batch sizes in a way such that the gradient and Jacobian inexactness in iteration $t$ are proportional to the previous increment, i.e., $\epsilon_1(t) \propto \|s_t\|^2, \epsilon_2(t) \propto \|s_t\|$. Such an adaptive inexact criterion is justified in the cubic regularization literature (Wang et al., 2018a; Wang et al., 2019) with the following advantages: 1) it is adapted to the optimization increment and hence leads to reduced batch sizes when the increment is large in the early iterations; 2) it makes the batch size scheduling scheme in Lemma practical, as the batch sizes in iteration $t$ now depend on the increment $\|s_t\|$ obtained in the previous iteration $t - 1$. We also note that since the sub-sampling scheme is adapted to the previous increment, the output rule of Algorithm is designed to control the value of both the current and the previous increments. This termination rule is critical to bound the adapted gradient and Jacobian inexactness in the analysis.

We obtain the following global convergence and sample complexity result of stochastic Cubic-GDA.

**Theorem 2 (Convergence and sample complexity).** Let Assumption 2 and Theorem 3 hold. For $0 < \epsilon \leq L_1 \sqrt{mL^2}, \frac{L_0}{\sqrt{3L^2}}, \eta_0 \leq \frac{1}{36L_0}, T \geq \Phi((\epsilon_0^2 + \epsilon^2) + 3L_0^2 \epsilon^2) / L_0^2, N_t \geq \mathcal{O}\left(\frac{L_0 \ln(1/\delta) + L_0}{\kappa^2 \|s_t \|^2 + \epsilon^2 / L_0}\right)$. Moreover, in iteration $t$, choose the batch sizes according to eqs. (8) and (9) with the inexactness given by

$$\epsilon_1(t) = \frac{L_0}{2} \left(\|s_t\|^2 + \frac{\epsilon^2}{33L_0}\right) \land 2L_0, \quad \epsilon_2(t) = \frac{L_0}{2(\kappa + 1)^2} \left(\|s_t\| + \frac{\epsilon}{\sqrt{33L_0}}\right) \land 4L_1.$$
Then, the output of Stochastic Cubic-GDA satisfies
\[ \mu(x_{T'}) \leq \epsilon. \]  
(13)

Consequently, the total number of cubic iterations satisfies
\[ T' \leq O(\sqrt{L^2K^2\epsilon^{-3}}), \]
the total number of queried gradient samples satisfies
\[ \sum_{t' = 0}^{T'} (N_t + |B_1(t)|) \leq O\left(\frac{L^2K^2\epsilon^{-3}}{\epsilon^5} \ln \frac{n}{\delta}\right), \]
and the total number of queried Jacobian samples satisfies
\[ \sum_{t = 0}^{T'-1} \sum_{k=1}^2 \sum_{\ell=1}^2 |B_{k,\ell}(t)| \leq O\left(\frac{L^2K^2\epsilon^{-5}}{\sqrt{\epsilon}} \ln \frac{n}{\delta}\right). \]

Therefore, under adaptive sub-sampling, the induced gradient and Jacobian inexactness $\epsilon_1(t), \epsilon_2(t)$ are properly controlled so that the iteration complexity $T'$ of stochastic Cubic-GDA remains in the same level as that of Cubic-GDA. Moreover, as opposed to the sample complexity of Cubic-GDA that scales linearly with regard to the data size $N$, the sample complexity of stochastic Cubic-GDA is independent of $N$. For example, by comparing the more expensive Jacobian sample complexity between Cubic-GDA and stochastic Cubic-GDA, we conclude that stochastic sub-sampling helps reduce the sample complexity so long as the data size is large enough, i.e., $N \geq \tilde{O}\left(\frac{L^2}{\epsilon^7 n}\right)$.

6 Conclusion

In this work, we develop a Cubic-GDA algorithm that leverages the second-order information and the cubic regularization technique to effectively find local minimax points in nonconvex minimax optimization. Our key observation is that Cubic-GDA has an intrinsic potential function that monotonically decreases in the optimization process, and this leads to a guaranteed global convergence of the algorithm. Moreover, we propose a stochastic variance of Cubic-GDA for large-scale minimax optimization, and characterize its sample complexity under stochastic sub-sampling. Limitations: The cubic regularization step of Cubic-GDA requires computing Jacobian-vector products, which can be computationally intensive for some applications. An interesting future direction is to reduce the amount of such computations by using zero-th order algorithms that use function values to approximate gradients and Jacobian-vector products. Negative social impacts: This is a fundamental theoretical study and does not have any potential negative social impact.

References

[Abadi, 2015] Abadi, M. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Adolphs et al., 2019] Adolphs, L., Daneshmand, H., Lucchi, A., and Hofmann, T. (2019). Local saddle point optimization: A curvature exploitation approach. In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), pages 486–495.

[Agarwal et al., 2017] Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. (2017). Finding approximate local minima faster than gradient descent. In Proc. Annual ACM SIGACT Symposium on Theory of Computing (STOC), pages 1195–1199.

[Carmon and Duchi, 2016] Carmon, Y. and Duchi, J. C. (2016). Gradient descent efficiently finds the cubic-regularized non-convex Newton step. ArXiv: 1612.00547.

[Cartis et al., 2011a] Cartis, C., Gould, N. I. M., and Toint, P. (2011a). Adaptive cubic regularization methods for unconstrained optimization. part ii: worst-case function- and derivative-evaluation complexity. Mathematical Programming, 130(2):295–319.

[Cartis et al., 2011b] Cartis, C., Gould, N. I. M., and Toint, P. L. (2011b). Adaptive cubic regularization methods for unconstrained optimization. part i : motivation, convergence and numerical results. Mathematical Programming.

[Chen et al., 2021] Chen, Z., Zhou, Y., Xu, T., and Liang, Y. (2021). Proximal gradient descent-ascent: Variable convergence under kl geometry. In Proc. International Conference on Learning Representations (ICLR).
[Cherukuri et al., 2017] Cherukuri, A., Gharesifard, B., and Cortes, J. (2017). Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511.

[Daskalakis and Panageas, 2018] Daskalakis, C. and Panageas, I. (2018). The limit points of (optimistic) gradient descent in min-max optimization. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 9236–9246.

[Dauphin et al., 2014] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, page 2933–2941.

[Du and Hu, 2019] Du, S. S. and Hu, W. (2019). Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 196–205.

[Ferreira et al., 2012] Ferreira, M. A. M., Andrade, M., Matos, M. C. P., Filipe, J. A., and Coelho, M. P. (2012). Minimax theorem and nash equilibrium. *International Journal of Latest Trends in Finance & Economics*.

[Ghadimi et al., 2017] Ghadimi, S., Liu, H., and Zhang, T. (2017). Second-order methods with cubic regularization under inexact information. *ArXiv: 1710.05782*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.

[Greiwank, 1981] Griewank, A. (1981). The modification of newton’s method for unconstrained optimization by bounding cubic terms. *Technical Report*.

[Hallak and Teboulle, 2020] Hallak, N. and Teboulle, M. (2020). Finding second-order stationary points in constrained minimization: A feasible direction approach. *Journal of Optimization Theory and Applications*, 186(2):480–503.

[Harvey et al., 2019] Harvey, N. J., Liaw, C., and Randhawa, S. (2019). Simple and optimal high-probability bounds for strongly-convex stochastic gradient descent. *ArXiv:1909.00843*.

[Ho and Ermon, 2016] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573.

[Huang et al., 2020] Huang, F., Gao, S., Pei, J., and Huang, H. (2020). Accelerated zeroth-order momentum methods from mini to minimax optimization. *ArXiv:2008.08170*.

[Jiang et al., 2017] Jiang, B., Lin, T., and Zhang, S. (2017). A unified scheme to accelerate adaptive cubic regularization and gradient methods for convex optimization. *ArXiv:1710.04788*.

[Jin et al., 2017] Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017). How to escape saddle points efficiently. In *Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 1724–1732.

[Jin et al., 2020] Jin, C., Netrapalli, P., and Jordan, M. I. (2020). What is local optimality in nonconvex-nonconcave minimax optimization? In *Proc. International Conference on Machine Learning (ICML)*.

[Kohler and Lucchi, 2017] Kohler, J. M. and Lucchi, A. (2017). Sub-sampled cubic regularization for non-convex optimization. In *Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 1895–1904.

[Lin et al., 2020] Lin, T., Jin, C., and Jordan, M. I. (2020). On gradient descent ascent for nonconvex-concave minimax problems. In *Proc. International Conference on Machine Learning (ICML)*.

[Lu et al., 2020] Lu, S., Tsaknakis, I., Hong, M., and Chen, Y. (2020). Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing*. 

10
[Luo and Chen, 2021] Luo, L. and Chen, C. (2021). Finding second-order stationary point for nonconvex-strongly-concave minimax problem. arXiv:2110.04814.

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

[Nagarajan and Kolter, 2017] Nagarajan, V. and Kolter, J. Z. (2017). Gradient descent gan optimization is locally stable. In Proc. Neural Information Processing Systems, page 5591–5600.

[Nedić and Ozdaglar, 2009] Nedić, A. and Ozdaglar, A. (2009). Subgradient methods for saddle-point problems. Journal of optimization theory and applications, 142(1):205–228.

[Nesterov, 2008] Nesterov, Y. (2008). Accelerating the cubic regularization of newton’s method on convex problems. Mathematical Programming, 112(1):159–181.

[Nesterov and Polyak, 2006] Nesterov, Y. and Polyak, B. (2006). Cubic regularization of newton’s method and its global performance. Mathematical Programming.

[Neumann, 1928] Neumann, J. v. (1928). Zur theorie der gesellschaftsspiele. Mathematische annalen, 100(1):295–320.

[Nouiehed et al., 2019] Nouiehed, M., Sanjabi, M., Huang, T., Lee, J. D., and Razaviyayn, M. (2019). Solving a class of non-convex min-max games using iterative first order methods. In Proc. Advances in Neural Information Processing Systems (NeurIPS), pages 14934–14942.

[Paszke, 2019] Paszke, A. (2019). Pytorch: An imperative style, high-performance deep learning library. In Proc. Advances in Neural Information Processing Systems (Neurips), pages 8024–8035.

[Qiu et al., 2020] Qiu, S., Yang, Z., Wei, X., Ye, J., and Wang, Z. (2020). Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear td learning. ArXiv:2008.10103.

[Robinson, 1951] Robinson, J. (1951). An iterative method of solving a game. Annals of mathematics, 54(2):296–301.

[Sinha et al., 2018] Sinha, A., Namkoong, H., and Duchi, J. (2018). Certifiable distributional robustness with principled adversarial training. In Proc. International Conference on Learning Representations (ICLR).

[Sinha et al., 2017] Sinha, A., Namkoong, H., and Duchi, J. C. (2017). Certifying some distributional robustness with principled adversarial training. In Proc. International Conference on Learning Representations (ICLR).

[Song et al., 2018] Song, J., Ren, H., Sadigh, D., and Ermon, S. (2018). Multi-agent generative adversarial imitation learning. In Proc. Advances in Neural Information Processing Systems (NeurIPS), pages 7461–7472.

[Tripuraneni et al., 2018] Tripuraneni, N., Stern, M., Jin, C., Regier, J., and Jordan, M. I. (2018). Stochastic cubic regularization for fast nonconvex optimization. In Proc. International Conference on Neural Information Processing Systems, NIPS’18, page 2904–2913.

[Wang et al., 2020] Wang, Y., Zhang, G., and Ba, J. (2020). On solving minimax optimization locally: A follow-the-ridge approach. In Proc. International Conference on Learning Representations.

[Wang et al., 2018a] Wang, Z., Zhou, Y., Liang, Y., and Lan, G. (2018a). A note on inexact condition for cubic regularized newton’s method. ArXiv:1808.07384.

[Wang et al., 2018b] Wang, Z., Zhou, Y., Liang, Y., and Lan, G. (2018b). Sample Complexity of Stochastic Variance-Reduced Cubic Regularization for Nonconvex Optimization. ArXiv:1802.07372v1.

[Wang et al., 2019] Wang, Z., Zhou, Y., Liang, Y., and Lan, G. (2019). Stochastic variance-reduced cubic regularization for nonconvex optimization. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 2731–2740.
[Xie et al., 2020] Xie, G., Luo, L., Lian, Y., and Zhang, Z. (2020). Lower complexity bounds for finite-sum convex-concave minimax optimization problems. In Proc. International Conference on Machine Learning (ICML).

[Xu et al., 2020a] Xu, P., Roosta, F., and Mahoney, M. W. (2020a). Newton-type methods for non-convex optimization under inexact hessian information. Mathematical Programming, 184(1):35–70.

[Xu et al., 2020b] Xu, T., Wang, Z., Liang, Y., and Poor, H. V. (2020b). Enhanced first and zeroth order variance reduced algorithms for min-max optimization. ArXiv:2006.09361.

[Xu et al., 2020c] Xu, T., Wang, Z., Liang, Y., and Poor, H. V. (2020c). Gradient free minimax optimization: Variance reduction and faster convergence. ArXiv:2006.09361.

[Xu et al., 2020d] Xu, Z., Zhang, H., Xu, Y., and Lan, G. (2020d). A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. ArXiv:2006.02032.

[Yang et al., 2020] Yang, J., Kiyavash, N., and He, N. (2020). Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems. In Proc. Advances in Neural Information Processing Systems (NeurIPS).

[Yue et al., 2019] Yue, M.-C., Zhou, Z., and Man-Chu So, A. (2019). On the quadratic convergence of the cubic regularization method under a local error bound condition. SIAM Journal on Optimization, 29(1):904–932.

[Zhang and Wang, 2021] Zhang, G. and Wang, Y. (2021). On the suboptimality of negative momentum for minimax optimization. In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), pages 2098–2106.

[Zhang et al., 2021] Zhang, G., Wu, K., Poupart, P., and Yu, Y. (2021). Newton-type methods for minimax optimization. arXiv:2006.14592.

[Zhou and Liang, 2018] Zhou, Y. and Liang, Y. (2018). Critical points of linear neural networks: Analytical forms and landscape properties. In Proc. International Conference on Learning Representations (ICLR).

[Zhou et al., 2018] Zhou, Y., Wang, Z., and Liang, Y. (2018). Convergence of cubic regularization for nonconvex optimization under kl property. In Proc. Advances in Neural Information Processing Systems (NeurIPS), pages 3760–3769.
Appendix

Table of Contents

A Supporting Lemmas 13
B Proof of Proposition 2 16
C Proof of Proposition 3 17
D Proof of Theorem 1 18
E Proof of Theorem 2 19
F Convergence Rate of SGA 22
G Compute Jacobian-Vector Product 22

A Supporting Lemmas

We first prove the following auxiliary lemma that bounds the spectral norm of the Jacobian matrices.

Lemma 2. Let Assumption 1 hold. Then, for any \( x \in \mathbb{R}^m \) and \( y \in \mathbb{R}^n \), the Jacobian matrices of \( f(x, y) \) and \( G(x, y) = [\nabla_{11} f - \nabla_{12} f(\nabla_{22} f)^{-1} \nabla_{21} f](x, y) \) satisfy the following bounds.

\[
\begin{align*}
\| [\nabla_{22} f(x, y)]^{-1} \| & \leq \mu^{-1}, \\
\| \nabla_{12} f(x, y) \| & = \| \nabla_{21} f(x, y) \| \leq L_1, \\
\| \nabla_{11} f(x, y) \| & \leq L_1, \\
\| G(x, y) \| & \leq L_1(1 + \kappa).
\end{align*}
\] (14)(15)(16)(17)

The same bounds also hold for \( \hat{\nabla}_{11} f, \hat{\nabla}_{12} f, \hat{\nabla}_{21} f, \hat{\nabla}_{22} f \) and \( \hat{G} \) defined in Section 5 under Assumption 2.

Proof. We first prove eq. (14). Consider any \( x \in \mathbb{R}^m \) and \( y \in \mathbb{R}^n \). By Assumption 1 we know that \( f(x, \cdot) \) is \( \mu \)-strongly concave, which implies that \( -\nabla_{22} f(x, y) \succeq \mu I \). Thus, we further conclude that

\[
\| [\nabla_{22} f(x, y)]^{-1} \| = \lambda_{\text{max}}( -\nabla_{22} f(x, y) )^{-1} = \left( \lambda_{\text{min}}( -\nabla_{22} f(x, y) ) \right)^{-1} \leq \mu^{-1}.
\] (18)

Next, we prove eq. (15). Consider any \( x, u \in \mathbb{R}^m \) and \( y \in \mathbb{R}^n \), we have

\[
\| \nabla_{21} f(x, y) \| u \| = \| \frac{d}{dt} \nabla_{2} f(x + tu, y) \|_{t=0} \|
= \| \lim_{t \to 0} \frac{1}{t} [\nabla_{2} f(x + tu, y) - \nabla_{2} f(x, y)] \|
= \lim_{t \to 0} \frac{1}{t} \| \nabla_{2} f(x + tu, y) - \nabla_{2} f(x, y) \|
\leq \lim_{t \to 0} \frac{L_1}{t} \| tu \| = L_1 \| u \|,
\] (19)

which implies that \( \| \nabla_{21} f(x, y) \| \leq L_1 \). Since \( f \) is twice differentiable and has continuous second-order derivative, we have \( \nabla_{12} f(x, y)^\top = \nabla_{21} f(x, y) \), and hence eq. (15) follows. The proof of eq. (16) is similar.

Finally, eq. (17) can be proved as follows using eqs. (14) & (15).

\[
\| G(x, y) \| \leq \| \nabla_{11} f(x, y) \| + \| \nabla_{12} f(x, y) \| \| \nabla_{22} f^{-1}(x, y) \| \| \nabla_{21} f(x, y) \| \leq L_1 + L_1 \mu^{-1} L_1 = L_1(1 + \kappa).
\] (20)

The proof is similar for the stochastic minimax optimization problem (Q) in Section 5 under Assumption 2.
The following lemma restates Lemma 3 of [Wang et al., 2018a].

**Lemma 3.** The solution \( s_{k+1} \) of the cubic regularization problem in Algorithm 1 satisfies the following conditions,

\[
\nabla f(x_t, y_{t+1}) + G(x_t, y_{t+1})s_{t+1} + \frac{1}{2t_x} \| s_{t+1} \| s_{t+1} = 0, \tag{19}
\]

\[
G(x_t, y_{t+1}) + \frac{1}{2t_x} \| s_{t+1} \| I \succeq 0, \tag{20}
\]

\[
\nabla f(x_t, y_{t+1})^\top s_{t+1} + \frac{1}{2} s_{t+1}^\top G(x_t, y_{t+1}) s_{t+1} \leq -\frac{1}{4t_x} \| s_{t+1} \|^3. \tag{21}
\]

**Lemma 4.** Suppose the gradient \( \nabla f(x_t, y_{t+1}) \) and Hessian \( G(x_t, y_{t+1}) \) involved in the cubic-regularization step in Algorithm 1 satisfy the following bounds for all \( t \leq T' - 1 \) with \( T' = \min\{t : \| s_{t-1} \| \vee \| s_t \| \leq \epsilon'\} \):

\[
\| \nabla \Phi(x_t) - \nabla f(x_t, y_{t+1}) \| \leq \beta(\| s_t \|^2 + \epsilon'), \tag{22}
\]

\[
\| \nabla^2 \Phi(x_t) - G(x_t, y_{t+1}) \| \leq \alpha(\| s_t \|^3 + \epsilon'). \tag{23}
\]

Then, choosing \( \eta_x \leq (9L_x + 18\alpha + 28\beta)^{-1} \), the sequence \( \{x_t\} \) generated by Algorithm 1 satisfies that for all \( t \leq T' - 2 \),

\[
H_{t+1} - H_t \leq -(L_x + \alpha + \beta)(\| s_{t+1} \|^3 + \| s_t \|^3), \tag{24}
\]

where \( H_t = \Phi(x_t) + (L_x + 2\alpha + 3\beta)\| x_t - x_{t-1} \|^3 \). The same conclusion holds for Algorithm 2 by replacing \( \nabla f(x_t, y_{t+1}), G(x_t, y_{t+1}) \) with their stochastic estimators \( \tilde{\nabla} f(x_t, y_{t+1}), \tilde{G}(x_t, y_{t+1}) \) respectively.

**Proof.** The proof logic follows that of the proof of Theorem 1 in [Wang et al., 2019]. By the Lipschitz continuity of the Hessian of \( \Phi \), we obtain that

\[
\Phi(x_{t+1}) - \Phi(x_t)
\]

\[
\leq \nabla \Phi(x_t)^\top (x_{t+1} - x_t) + \frac{1}{2} (x_{t+1} - x_t)^\top \nabla^2 \Phi(x_t) (x_{t+1} - x_t) + \frac{L_x}{6} \| x_{t+1} - x_t \|^3
\]

\[
= \nabla f(x_t, y_{t+1})^\top s_{t+1} + \frac{1}{2} s_{t+1}^\top G(x_t, y_{t+1}) s_{t+1} + \frac{L_x}{6} \| s_{t+1} \|^3
\]

\[
+ (\nabla \Phi(x_t) - \nabla f(x_t, y_{t+1}))^\top s_{t+1} + \frac{1}{2} s_{t+1}^\top (\nabla^2 \Phi(x_t) - G(x_t, y_{t+1})) s_{t+1}
\]

\[
\leq (\frac{L_x}{6} - \frac{1}{4t_x}) \| s_{t+1} \|^3 + \beta(\| s_t \|^2 + \epsilon')^2 + \frac{\alpha}{2} \| s_{t+1} \|^2 (\| s_t \| + \epsilon')
\]

\[
\leq (\frac{L_x}{6} - \frac{1}{4t_x}) \| s_{t+1} \|^3 + (\frac{\alpha}{2} + \beta)(2 \| s_{t+1} \|^3 + \| s_t \|^3 + \epsilon'^3)
\]

\[
\leq (\frac{L_x}{6} - \frac{1}{4t_x}) \| s_{t+1} \|^3 + (\frac{\alpha}{2} + \beta)(3 \| s_{t+1} \|^3 + 2 \| s_t \|^3)
\]

\[
\leq -(\frac{L_x}{6} - \frac{3\alpha}{2} + 3\beta) \| s_{t+1} \|^3 + (\alpha + 2\beta) \| s_t \|^3,
\]

where (i) uses eqs. (21), (22) & (23), (ii) uses the inequality that \( ab^2 \leq a^3 \vee b^3 \leq a^3 + b^3, \forall a, b \geq 0 \), (iii) uses \( \epsilon'^3 \leq \| s_t \|^3 \vee \| s_{t+1} \|^3 \leq \| s_t \|^3 + \| s_{t+1} \|^3, \forall 0 \leq t \leq T' - 2 \) based on the termination criterion of \( T' \), and (iv) uses \( \eta_x \leq (9L_x + 18\alpha + 28\beta)^{-1} \). Eq. (24) follows from the above inequality by defining \( H_t = \Phi(x_t) + (L_x + 2\alpha + 3\beta)\| x_t - x_{t-1} \|^2 \).

Note that the cubic-regularization step in Algorithm 2 simply replaces \( \nabla f(x_t, y_{t+1}), G(x_t, y_{t+1}) \) in Algorithm 1 with their stochastic estimators \( \tilde{\nabla} f(x_t, y_{t+1}), \tilde{G}(x_t, y_{t+1}) \) respectively. Hence, eq. (24) holds for Algorithm 2 after such replacement in eqs. (22) & (23).
Lemma 5. Suppose all the conditions of Lemma 4 hold. If $T' \geq \frac{\Phi(x_0) - \Phi^* + (L_\Phi + 3\alpha + 4\beta)e^2}{(L_\Phi + \alpha + \beta)e^3}$ in Algorithm 4 then $T' = \min\{t \geq 1 : \|s_{t-1}\| \lor \|s_t\| \leq \epsilon'\} \leq \frac{\Phi(x_0) - \Phi^* + (L_\Phi + 3\alpha + 4\beta)e^2}{(L_\Phi + \alpha + \beta)e^3} \leq T$. Consequently, the output of Algorithm 4 has the following convergence rate

$$\|\nabla \Phi(x_T)\| \leq \left(\frac{1}{2\eta_x} + L_\Phi + 2\alpha + 2\beta\right)\epsilon^2, \quad (25)$$

$$\nabla^2 \Phi(x_T) \succeq -\left(\frac{1}{2\eta_x} + L_\Phi + 2\alpha\right)\epsilon I. \quad (26)$$

The same conclusion holds for Algorithm 5 by replacing $\nabla f(x_t, y_{t+1})$, $G(x_t, y_{t+1})$ in the conditions 22, 23 with their stochastic estimators $\tilde{\nabla} f(x_t, y_{t+1})$, $\tilde{G}(x_t, y_{t+1})$ respectively.

Proof. The proof logics follows that of the proof of Theorem 1 in Wang et al., 2019.

Suppose $T' \leq T$ does not hold, i.e., $\|s_{t-1}\| \lor \|s_t\| > \epsilon'$, $\forall 1 \leq t \leq T$, which implies that eq. (24) holds for all $0 \leq t \leq T - 1$ based on Lemma 4.

On one hand, telescoping eq. (24) over $t = 0, 1, \ldots, T - 1$ yields

$$H_0 - H_T \geq (L_\Phi + \alpha + \beta) \sum_{t=0}^{T-1} (\|s_{t+1}\|^3 + \|s_t\|^3)$$

$$\geq (L_\Phi + \alpha + \beta) \sum_{t=0}^{T-1} (\|s_t\| \lor \|s_{t+1}\|)^3$$

$$> T(L_\Phi + \alpha + \beta)\epsilon^3$$

$$\geq \Phi(x_0) - \Phi^* + (L_\Phi + 3\alpha + 4\beta)e^2. \quad (27)$$

On the other hand, recalling the definition of $H_t$ in Lemma 4, we have

$$H_0 - H_T = \Phi(x_0) - \Phi(x_T) + (L_\Phi + 2\alpha + 3\beta)(\|s_0\|^2 - \|s_T\|^2)$$

$$\leq (\Phi(x_0) - \Phi^*) + (L_\Phi + 3\alpha + 4\beta)e^2, \quad (28)$$

where (i) uses $\|s_0\| = \epsilon'$ and $\Phi(x_T) \geq \Phi^* = \min_{x \in \mathbb{R}} \Phi(x)$. Note that eqs. (27) & (28) contradict. Therefore, we must have $1 \leq T' \leq T$ for any $T \geq \frac{\Phi(x_0) - \Phi^* + (L_\Phi + 3\alpha + 4\beta)e^2}{(L_\Phi + \alpha + \beta)e^3}$, which implies that $T' \leq \frac{\Phi(x_0) - \Phi^* + (L_\Phi + 3\alpha + 4\beta)e^2}{(L_\Phi + \alpha + \beta)e^3} \leq T$.

Finally, we conclude that

$$||\nabla \Phi(x_{T'})||$$

$$\leq ||\nabla \Phi(x_{T'}) - \nabla f(x_{T'-1}, y_{T'}) - G(x_{T'-1}, y_{T'})||_{s_{T'}} - \frac{1}{2\eta_x}||s_{T'}||_{s_{T'}}$$

$$\leq ||\nabla \Phi(x_{T'}) - \nabla \Phi(x_{T'-1}) - \nabla^2 \Phi(x_{T'-1})||_{s_{T'}} + ||\nabla \Phi(x_{T'-1}) - \nabla f(x_{T'-1}, y_{T'})||$$

$$+ ||\nabla^2 \Phi(x_{T'-1})||_{s_{T'}} - G(x_{T'-1}, y_{T'})||_{s_{T'}} + \frac{1}{2\eta_x}||s_{T'}||^2$$

$$\leq (L_\Phi + \frac{1}{2\eta_x})||s_{T'}||^2 + \beta(||s_{T-1}||^2 + \epsilon^2) + \alpha(||s_{T-1}|| + \epsilon')||s_{T'}||$$

$$\leq \left(\frac{1}{2\eta_x} + L_\Phi + 2\alpha + 2\beta\right)\epsilon^2, \quad (29)$$

where (i) uses eq. (19), (ii) uses eqs. (22, 23) and the item 2 of Proposition 2 that $\nabla^2 \Phi$ is $L_\Phi$-Lipschitz, and (iii) uses $\|s_{T-1}\| \lor \|s_{T'}\| \leq \epsilon'$. Also,

$$\nabla^2 \Phi(x_{T'}) \succeq (G(x_{T'-1}, y_{T'}) - \|G(x_{T'-1}, y_{T'}) - \nabla^2 \Phi(x_{T'})\| I$$

$$\succeq (\frac{1}{2\eta_x} - \|s_{T'}\| I - \|G(x_{T'-1}, y_{T'}) - \nabla^2 \Phi(x_{T'-1})\| I - ||\nabla^2 \Phi(x_{T'}) - \nabla^2 \Phi(x_{T'-1})|| I$$
Then, the stochastic approximators satisfy the following error bounds with probability at least 

where (i) uses Weyl’s inequality, (ii) uses eq. (20), (iii) uses eq. (22) and the item of Proposition that \( \nabla^2 \Phi \) is \( L_\Phi \)-Lipschitz, and (iv) uses \( \| s_{T-1} \| \vee \| s_T \| \leq \epsilon' \).

For the stochastic minimax optimization problem (Q) in Section 5 we prove the following supporting lemma on the error of the stochastic estimators.

**Lemma 1.** Fix any \( 0 < \epsilon_1 \leq 2L_\Phi, 0 < \epsilon_2 \leq 4L_1 \) and choose the following batch sizes

\[
|B_1| \geq \Omega \left( \frac{L_\Phi^2}{\epsilon_1^2} \ln \frac{m}{\delta} \right),
\]

\[
|B_{11}|, |B_{12}|, |B_{21}|, |B_{22}| \geq \Omega \left( \frac{L_1^2}{\epsilon_2^2} \ln \frac{m + n}{\delta} \right).
\]

Then, the stochastic approximators satisfy the following error bounds with probability at least \( 1 - \delta \).

\[
\| \hat{\nabla}_1 f(x,y) - \nabla_1 f(x,y) \| \leq \epsilon_1,
\]

\[
\| \nabla_{k,\ell}^2 f(x,y) - \nabla_{k,\ell}^2 f(x,y) \| \leq \epsilon_2, \quad \forall k, \ell \in \{1,2\},
\]

\[
\| \hat{G}(x,y) - G(x,y) \| \leq (\kappa + 1)^2 \epsilon_2.
\]

**Proof.** Based on Lemmas 6 & 8 in [Kohler and Lucchi, 2017], we obtain that with probability at least \( 1 - \delta \), the following bounds hold. (We replaced \( \delta \) in [Kohler and Lucchi, 2017] with \( \delta/5 \) by applying union bound to the following 5 events.)

\[
\| \hat{\nabla}_1 f(x,y) - \nabla_1 f(x,y) \| \leq 4\sqrt{2}L_\Phi \sqrt{\frac{\ln(10m/\delta) + 1/4}{|B_1|}} \leq \epsilon_1,
\]

\[
\| \nabla_{k,\ell}^2 f(x,y) - \nabla_{k,\ell}^2 f(x,y) \| \leq 4L_1 \sqrt{\frac{\ln(10(m + n)/\delta)}{|B_{k,\ell}|}} \leq \epsilon_2; k, \ell \in \{1,2\}.
\]

Note that here we only consider the cases that \( |B_1|, |B_{k,\ell}| < N \) for all \( k, \ell \in \{1,2\} \). Otherwise, \( |B_1| = N \) yields \( \| \nabla_1 f(x,y) - \nabla_1 f(x,y) \| = 0 < \epsilon_1 \) and \( |B_{k,\ell}| = N \) yields \( \| \nabla_{k,\ell}^2 f(x,y) - \nabla_{k,\ell}^2 f(x,y) \| = 0 < \epsilon_2 \). Hence, in both cases, the above high probability bounds hold, which further implies eq. (12) following the argument below.

\[
\| \hat{G}(x,y) - G(x,y) \|
\leq \| \hat{\nabla}_1 f(x,y) - \nabla_1 f(x,y) \| + \| (\nabla_{12} f(\nabla_{22} f)^{-1} \hat{\nabla}_{21} f)(x,y) - (\nabla_{12} f(\nabla_{22} f)^{-1} \nabla_{21} f)(x,y) \|
\leq \epsilon_2 + \| (\nabla_{12} f - \nabla_{12} f)(\hat{\nabla}_{21} f)(x,y) \|
+ \| \nabla_{12} f((\hat{\nabla}_{22} f)^{-1} - (\nabla_{22} f)^{-1}) \hat{\nabla}_{21} f(x,y) \| + \| \nabla_{12} f(\nabla_{22} f)^{-1} (\hat{\nabla}_{21} f - \nabla_{21} f)(x,y) \|
\leq \epsilon_2 + \epsilon_2 \mu L_1 + L_1^2 \| \nabla_{22} f(x,y) \| \| (\nabla_{22} f - \hat{\nabla}_{22} f)(x,y) \| \| \nabla_{22} f(x,y)^{-1} \| L_1 \mu^{-1} \epsilon_2
\leq \epsilon_2 + 2\kappa \epsilon_2 + L_1^2 \mu^{-2} \epsilon_2 \leq (\kappa + 1)^2 \epsilon_2.
\]

where (i) and (ii) use Lemma 2.

Regarding the high-probability convergence rate of the inner stochastic gradient ascent (SGA) in Algorithm 2 the following result is a direct application of Theorem 3.1 in [Harvey et al., 2019].

**B Proof of Proposition 2**

**Proposition 2.** Let Assumption 1 hold. Then, the following statements hold.
1. Define $G(x, y) = [\nabla_1 f - \nabla_2 f (\nabla_2 f)^{-1} \nabla_2 f] (x, y)$. Then, $G$ is a Lipschitz continuous mapping with constant $L_G = L_2 (1 + \kappa)^2$;

2. The Hessian of $\Phi$ satisfies $\nabla^2 \Phi(x) = G(x, y^*(x))$, and it is Lipschitz continuous with constant $L_\Phi = L_2 (1 + \kappa)^2$.

**Proof.** We first prove the item 1. Consider any $x, x' \in \mathbb{R}^m$ and $y, y' \in \mathbb{R}^n$. For convenience we denote $z = (x, y)$ and $z' = (x', y')$. Then, by Assumption 1, and using the bounds of Lemma 2, we have that

$$
\|G(x', y') - G(x, y)\| \\
\leq \|\nabla_1 f(x', y') - \nabla_1 f(x, y)\| + \|\nabla_2 f(x', y') - \nabla_2 f(x, y)\| \|\nabla_2 f(x', y')\|^{-1} \|\nabla_2 f(x', y')\| \\
+ \|\nabla_2 f(x, y)\| \|\nabla_2 f(x', y')\|^{-1} \|\nabla_2 f(x', y')\| \\
+ \|\nabla_2 f(x, y)\| \|\nabla_2 f(x', y')\|^{-1} \|\nabla_2 f(x', y')\| \\
\leq L_2 \|z' - z\| + L_2 \|z' - z\| \mu^{-1} L_1 \\
+ L_2^2 \|\nabla_2 f(x', y')\|^{-1} \|\nabla_2 f(x, y)\| \|\nabla_2 f(x', y')\|^{-1} \|\nabla_2 f(x', y')\| + L_1 \mu^{-1} (L_2 \|z' - z\|) \\
\leq L_2 (1 + 2 \kappa) \|z' - z\| + L_1 \mu^{-1} (L_2 \|z' - z\|) \mu^{-1} \\
\leq L_2 (1 + \kappa)^2 \|z' - z\|.
$$

Next, we prove the item 2. Consider any fixed $x \in \mathbb{R}^m$, we know that $f(x, \cdot)$ achieves its maximum at $y^*(x)$, where the gradient vanishes, i.e., $\nabla_2 f(x, y^*(x)) = 0$. Thus, we further obtain that

$$
0 = \nabla_2 \nabla_2 f(x, y^*(x)) = \nabla_2 \nabla_2 f(x, y^*(x)) + \nabla_2 f(x, y^*(x)) \nabla y^*(x),
$$

which implies that

$$
\nabla y^*(x) = -\nabla_2 f(x, y^*(x)) \nabla_2 f(x, y^*(x)).
$$

With the above equation, we take derivative of $\nabla \Phi(x) = \nabla_1 f(x, y^*(x))$ and obtain that

$$
\nabla_2 \Phi(x) = \nabla_1 f(x, y^*(x)) + \nabla_2 f(x, y^*(x)) \nabla y^*(x) \\
= \nabla_1 f(x, y^*(x)) - \nabla_2 f(x, y^*(x)) \nabla_2 f(x, y^*(x)) \nabla_2 f(x, y^*(x)) \\
=G(x, y^*(x)).
$$

Moreover, we have that

$$
\|\nabla_2 \Phi(x') - \nabla_2 \Phi(x)\| = \|G(x', y^*(x')) - G(x, y^*(x))\| \\
\leq L_G \|x' - x\| + \|y^*(x') - y^*(x)\| \\
\leq L_G (1 + \kappa) \|x' - x\|,
$$

where the last step uses the item 1 of Proposition 1. This proves the item 2. \hfill \Box

**C** Proof of Proposition 3

**Proposition 3** (Potential decrease). Let Assumption 1 hold. For any $\alpha, \beta > 0$, choose $\epsilon' \leq \frac{\alpha}{3 \beta L_G}, \eta_x \leq (9L_\Phi + 18\alpha + 28\beta)^{-1}$ and $\eta_y = \frac{2}{L_1^2 \mu}$. Define the potential function $H_t := \Phi(x_t) + (L_\Phi + 2\alpha + 3\beta)\|s_t\|^3$. Then, when $N_t \geq \mathcal{O}(\kappa \ln \frac{L_1 \alpha \|s_t\|^2 L_2^4 (\alpha + 2\alpha + 3\beta)\|s_t\|^3}{L_G^2 \beta \epsilon' \kappa})$, the output of Cubic-GDA satisfies the following potential decrease property.

$$
H_{t+1} - H_t \leq -(L_\Phi + \alpha + \beta)(\|s_{t+1}\|^3 + \|s_t\|^3).
$$

**Proof.** The required number of inner gradient ascent steps is shown below

$$
N_0 \geq \kappa \ln \left( \frac{L_1 \|y_0 - y^*(x_0)\|/(2\beta \epsilon'^2)}{1}\right)
$$
\[ N_t \geq \kappa \ln \left( \frac{2L_1\alpha \|s_{t-1}\| + L_1(\alpha + L_GK)\|s_t\|}{L_G\beta \epsilon^2} \right) \]
\[ = O\left( \kappa \ln \frac{L_1\alpha \|s_{t-1}\| + L_1(\alpha + L_GK)\|s_t\|}{L_G\beta \epsilon^2} \right); \quad 1 \leq t \leq T'. \]  
(37)

To prove eq. (2), we first prove by induction that for any \( t \geq 0 \),
\[ \|y_{t+1} - y^*(x_t)\| \leq \frac{\alpha(\|s_t\| + \epsilon') \wedge \beta(\|s_t\|^2 + \epsilon'^2)}{L_G} \cdot \|y_0 - y^*(x_0)\| \exp \left( - \frac{1}{2\beta \epsilon'^2} L_1 \|y_0 - y^*(x_0)\| \right) \]
\[ \leq \sum_{i=0}^{t-1} \|y_{i+1} - y^*(x_i)\| \exp \left( - \frac{1}{2\beta \epsilon'^2} L_1 \|y_i - y^*(x_i)\| \right) \]
\[ \leq \frac{2\beta \epsilon'^2}{L_1} \left( \frac{\alpha(\|s_0\| + \epsilon') \wedge \beta(\|s_0\|^2 + \epsilon'^2)}{L_G} \right) \]
\[ \leq \frac{2\beta \epsilon'^2}{L_1} \left( \frac{\alpha(\|s_0\| + \epsilon') \wedge \beta(\|s_0\|^2 + \epsilon'^2)}{L_G} \right) \]
(38)

where (i) uses eq. (37) and \( \ln(1 - x) \leq -x < 0 \) for \( x = \kappa^{-1} \in (0, 1) \) and (ii) uses \( \|s_0\| = \epsilon' \leq \frac{\alpha L_1}{\beta L_G} \). Hence, eq. (38) holds when \( t = 0 \).

If eq. (38) holds for \( t = k-1 \in [0, T'-2] \), then
\[ \|y_{k+1} - y^*(x_k)\| \leq (1 - \kappa^{-1})^N_k \|y_k - y^*(x_{k-1})\| + (1 - \kappa^{-1})^N_k \|y^*(x_{k-1}) - y^*(x_k)\| \]
\[ \leq \exp \left( N_k \ln(1 - \kappa^{-1}) \right) \left( \frac{\alpha(\|s_{k-1}\| + \epsilon') \wedge \beta(\|s_{k-1}\|^2 + \epsilon'^2)}{L_G} \right) \|y_{k-1} - y^*(x_{k-1})\| \]
\[ \leq \exp \left( - \frac{2L_1\alpha \|s_{k-1}\| + L_1(\alpha + L_GK)\|s_k\|}{L_G\beta \epsilon'^2} \right) \left( \frac{\alpha(\|s_{k-1}\| + \epsilon') \wedge \beta(\|s_k\|^2 + \epsilon'^2)}{L_G} \right) \]
\[ \leq \frac{2\beta \epsilon'^2}{L_1} \left( \frac{\alpha(\|s_{k-1}\| + \epsilon') \wedge \beta(\|s_k\|^2 + \epsilon'^2)}{L_G} \right) \]
where (i) uses eq. (38) for \( t = k-1 \) and the fact that \( y^* \) is \( \kappa \)-Lipschitz mapping (see [Lin et al., 2020] [Chen et al., 2021] for the proof), (ii) uses \( \ln(1 - \kappa^{-1}) \leq -\kappa^{-1} \) and eq. (47), (iii) uses \( \epsilon' \leq \|s_{k-1}\| \vee \|s_k\| \leq \|s_{k-1}\| + \|s_k\| \) for \( k \leq T' - 1 \), and (iv) uses the condition that \( \epsilon' \leq \frac{\alpha L_1}{\beta L_G} \). This proves eq. (38) holds for \( t = k \) and thus for all \( t \in [0, T' - 1] \), which further implies eqs. (22) & (23). Hence, by Lemma 4 we prove that eq. (2) holds for all \( 0 \leq t \leq T' - 1 \).

**D Proof of Theorem 1**

**Theorem 1** (Global convergence rate). Let the conditions of Proposition 3 hold with \( \alpha = \beta = L_G \). For any \( 0 < \epsilon \leq \frac{L_1\alpha}{L_G} \), choose \( \epsilon' = \frac{\epsilon}{\sqrt{3L_G}} \) and \( T \geq \frac{\Phi(x_0) - \Phi^* + 8L_1\epsilon'^2}{3L_G\epsilon'^2} \). Then, the output of Cubic-GDA satisfies:
\[ \mu(x_T') \leq \epsilon. \]  
(5)

Consequently, the total number of cubic iterations satisfies \( T' \leq O(\sqrt{2\kappa^3}\epsilon^{-3}) \), and the total number of gradient ascent iterations satisfies \( \sum_{t=0}^{T'-1} N_t \leq \tilde{O}(\sqrt{L_1\kappa^2}\epsilon^{-3}) \).
Proof. Substituting $\alpha = \beta = L_B = L_2(1 + \kappa)^3$ and $\epsilon' = \frac{\epsilon}{\sqrt{33L_\Phi}}$ into Lemma 5 yields that when $T \geq \frac{\sqrt{33L_\Phi}3^3\theta(x_0) - \Phi^2 + 8L_\Phi^2}{3\epsilon^2}$ and $\eta_x = (55L_\Phi)^{-1}$, we have $T' \leq \frac{\sqrt{33L_\Phi}3^3\theta(x_0) - \Phi^2 + 8L_\Phi^2}{3\epsilon^2} = O(\sqrt{2}k^{1.5}\epsilon^{-3}) \leq T$, and moreover,

$$\|\nabla \Phi(x_{T'})\| \leq \left(\frac{1}{2\eta_x} + L_\Phi + 2\alpha + 2\beta\right)\epsilon^2 \leq \epsilon^2,$$

$$\lambda_{\text{min}}(\nabla^2 \Phi(x_{T'})) \geq -\left(\frac{1}{2\eta_x} + L_\Phi + 2\alpha\right)\epsilon' \geq -\sqrt{33L_\Phi}\epsilon.$$

This proves eq. (5).

Note that the number of gradient ascent iterations $N_t$ should satisfy eq. (37). Substituting $\alpha = \beta = L_B = L_2(1 + \kappa)^3$ and $\epsilon' = \frac{\epsilon}{\sqrt{33L_\Phi}}$ into eq. (37) yields that

$$N_0 \geq \kappa \ln \left(\frac{L_1\|y_0 - y^*(x_0)\|/(2\beta\epsilon^2)}{\sqrt{L_2\alpha(\|s_{t-1}\| + L_1\alpha + L_2\kappa)\|s_t\|}}\right)$$

$$N_t \geq \kappa \ln \left(\frac{2L_1\alpha\|s_{t-1}\| + L_1(\alpha + L_2\kappa)\|s_t\|}{L_2\beta\epsilon^2}\right)$$

$$\sum_{t=0}^{T'-1} N_t = \kappa \ln \left(\frac{66L_1(1 + \kappa)\|s_{t-1}\| + 33L_1(2 + \kappa)\|s_t\|}{\epsilon^2}\right)$$

$$\leq \kappa \ln \left(\frac{33L_1\|y_0 - y^*(x_0)\|/(2\epsilon^2)}{\sqrt{L_2\alpha(\|s_{t-1}\| + L_1\alpha + L_2\kappa)\|s_t\|}}\right)$$

$$+ \kappa \sum_{t=0}^{T'-1} \ln \left(\|s_{t-1}\| + \|s_t\|\right)$$

$$+ \kappa T' \ln \left(\frac{4(\delta_0 - H^*)}{3L_2T'}\right)$$

$$+ \kappa T' \ln \left(\frac{66L_1\epsilon^{-2}(1 + \kappa)}{\epsilon^2}\right)$$

$$= O(\kappa T') \leq O(\sqrt{2k^{2.5}\epsilon^{-3}}) \quad (41)$$

where (i) uses $(a + b)^3 \leq 4(a^3 + b^3)$ for any $a, b \geq 0$, (ii) applies Jensen’s inequality to the concave function $\ln(.)$, and (iii) telescopes eq. (2) over $t = 0, 1, \ldots, T' - 1$.\hfill $\square$

E Proof of Theorem 2

Theorem 2 (Convergence and sample complexity). Let Assumption 5 and Theorem 3 hold. For $0 < \epsilon \leq \frac{L_B}{\sqrt{33L_\Phi}}$, choose $\epsilon' = \frac{\epsilon}{\sqrt{33L_\Phi}}$, $\eta_x \leq \frac{1}{\sqrt{33L_\Phi}}$, $T \geq \frac{\theta(x_0) - \Phi^2 + 8L_\Phi^2}{3\epsilon^2}$ and $N_t \geq O(\sqrt{\frac{k^{1.5}\epsilon^{-3}}{L_2\epsilon^2}})$. Moreover, in iteration $t$, choose the batch sizes according to eqs. (3) and (9) with the inexactness given by

$$\epsilon_1(t) = \frac{L_\Phi}{2}\left(\|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi}\right) \land 2L_0, \quad \epsilon_2(t) = \frac{L_\Phi}{2(\kappa + 1)^2}\left(\|s_{t-1}\|^2 + \frac{\epsilon}{\sqrt{33L_\Phi}}\right) \land 4L_1.$$

19
Then, the output of Stochastic Cubic-GDA satisfies
\[ \mu(x_{T'}) \leq \epsilon. \] (13)

Consequently, the total number of cubic iterations satisfies \( T' \leq O(\sqrt{L_2}\kappa^{1.5}\epsilon^{-3}) \), the total number of queried gradient samples satisfies \( \sum_{t'=0}^{T'-1} (N_t + |B_1(t)|) \leq O\left(\frac{L_2^2\kappa^2}{\sqrt{L_2}} \ln \frac{m}{\eta} \right) \), and the total number of queried Jacobian samples satisfies \( \sum_{t'=0}^{T'-1} \sum_{k=1}^{2} \sum_{\ell=1}^{2} |B_{k,\ell}(t)| \leq O\left(\frac{L_2^2\kappa^2}{\sqrt{L_2}} \ln \frac{m+n}{\delta} \right) \).

**Proof.** In Lemma 1, replace \( x, y \) with \( x_t, y_t, B_1 \) with \( B_1(t) \), and \( B_{k,\ell} \) with \( B_{k,\ell}(t) \) for any \( k, \ell \in \{1, 2\} \), and substitute the following hyperparameters,

\[ \epsilon_1(t) = \frac{L_\Phi}{2} \left( \|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi} \right) \wedge 2L_0 = O(L_\Phi\|s_t\|^2 + \epsilon^2), \]

\[ \epsilon_2(t) = \frac{L_\Phi}{2(\kappa + 1)^2} \left( \|s_t\|^2 + \frac{\epsilon}{33L_\Phi} \right) \wedge 4L_1 = O(\kappa^{-2}(L_\Phi\|s_t\|^2 + \sqrt{L_\Phi}\epsilon)). \]

Then, we obtain that using the following batch sizes,

\[
\begin{align*}
|B_1(t)| &\geq O\left(\frac{L_0^2}{L_\Phi\|s_t\|^4 + \epsilon^2} \ln \frac{m}{\delta} \right), \\
|B_{k,\ell}(t)| &\geq O\left(\frac{L_\Phi^2\kappa^4}{L_\Phi(\|s_t\|^2 + \epsilon^2)} \ln \frac{m+n}{\delta} \right),
\end{align*}
\] (42)

the stochastic approximators satisfy the following error bounds with probability at least \( 1 - \delta \).

\[ \|\tilde{G}_1f(x_t, y_t) - \nabla f(x_t, y_t)\| \leq \epsilon_1(t) \leq \frac{L_\Phi}{2} \left( \|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi} \right), \] (43)

\[ \|\tilde{G}(x_t, y_t) - G(x_t, y_t)\| \leq \epsilon_2(t) \leq \frac{L_\Phi}{2} \left( \|s_t\|^2 + \frac{\epsilon}{\sqrt{33L_\Phi}} \right). \] (44)

Based on Theorem 3 using the following number of stochastic gradient ascent steps

\[ N_t \geq O\left(\frac{L_0\ln(1/\delta) + L_0^2}{\mu^2L_\Phi^2((\|s_t\|^2 + \epsilon^2)/(33L_\Phi)) / L_1 \wedge (\|s_t\|^2 + \epsilon/(\sqrt{33L_\Phi}) / L_G) \right)^2 \]

\[ \overset{(i)}{=} O\left(\frac{L_0\ln(1/\delta) + L_0^2}{((\|s_t\|^2 + \epsilon^2) / \kappa) \wedge L_1(\|s_t\|^2 + \epsilon/\sqrt{L_\Phi})} \right)^2 \]

\[ = O\left(\kappa^{-2}(L_\Phi\|s_t\|^2 + \epsilon^2) \wedge L_1(\|s_t\|^2 + \epsilon^2/\sqrt{L_\Phi}) \right), \] (45)

where (i) uses \( \kappa = L_1 / \mu \) and \( L_\Phi = L_G(1 + \kappa) \) (Proposition 3). we have

\[ \|y_{t+1} - y^*(x_t)\| \leq O\left(\sqrt{\frac{L_0\ln(1/\delta) + L_0^2}{\mu^2N_t}} \right) \]

\[ \leq \frac{L_\Phi}{2} \min\left(\frac{1}{L_1}(\|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi}), \frac{1}{L_G}(\|s_t\|^2 + \frac{\epsilon}{\sqrt{33L_\Phi}}) \right). \] (46)

Therefore,

\[ \|\nabla \Phi(x_t) - \tilde{G}_1f(x_t, y_{t+1})\| \]

\[ \overset{(i)}{\leq} \|\nabla f(x_t, y^*(x_t)) - \nabla f(x_t, y_{t+1})\| + \|\tilde{G}_1f(x_t, y_{t+1}) - \nabla f(x_t, y_{t+1})\| \]

\[ \overset{(ii)}{\leq} L_1\|y_{t+1} - y^*(x_t)\| + \frac{L_\Phi}{2} \left( \|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi} \right) \]

\[ \overset{(iii)}{\leq} L_\Phi \left( \|s_t\|^2 + \frac{\epsilon^2}{33L_\Phi} \right), \] (47)
where (i) uses $\nabla \Phi(x) = \nabla_1 f(x, y^*(x))$ in Proposition 1, (ii) uses eq. (43) and Assumption 1, and (iii) uses eq. (46).

\[
\|\nabla^2 \Phi(x_t) - \hat{G}(x_t, y_{t+1})\| \\
\leq (i) \|G(x_t, y^*(x_t)) - G(x_t, y_{t+1})\| + \|\hat{G}(x_t, y_{t+1}) - G(x_t, y_{t+1})\| \\
\leq L_G \|y_{t+1} - y^*(x_t)\| + \frac{L_\Phi}{2} \left(\|s_t\| + \frac{\epsilon}{\sqrt{33L_\Phi}}\right) \\
\leq L_\Phi \left(\|s_t\| + \frac{\epsilon}{\sqrt{33L_\Phi}}\right),
\]

Eq. (47) & (48) imply that the conditions (22) & (23) hold with $\alpha = \beta = L_\Phi = L_2(1 + \kappa)^3$ and $\epsilon' = \frac{\epsilon}{\sqrt{33L_\Phi}}$. In Lemma 5 by substituting these values of $\alpha, \beta, \epsilon'$ and $\eta_\delta = (55L_\Phi)^{-1}$, we obtain that when $T \geq \frac{\Phi(x_0) - \Phi(\tilde{x}) + 8L_\Phi \epsilon^2}{33L_\Phi \epsilon'}$, we have $T' \leq \sqrt{\frac{33L_\Phi}{4\epsilon'}} = O\left(\sqrt{L_2 \kappa^1.5 \epsilon^{-3}}\right) \leq T$, and moreover,

\[
\|\nabla \Phi(x_{T'})\| \leq \left(\frac{1}{2\eta_\delta} + L_\Phi + 2\alpha + 2\beta\right) \epsilon'^2 \leq \epsilon^2,
\]

\[
\lambda_{\text{min}}(\nabla^2 \Phi(x_{T'})) \geq -\left(\frac{1}{2\eta_\delta} + L_\Phi + 2\alpha\right) \epsilon' \geq -\sqrt{33L_\Phi \epsilon},
\]

This proves that eq. (13) holds with probability at least $1 - \delta$.

Choosing both $N_t$ in eq. (45) and the batchsizes in eq. (42) with equality, the number of gradient computations has the following upper bound.

\[
\begin{align*}
\sum_{t=0}^{T'-1} (N_t + |B_1(t)|) \\
= \sum_{t=0}^{T'-1} O\left(\frac{L_0 \ln(1/\delta) + L_0^2}{\kappa^{-2}(L_\Phi^2 \|s_t\|^4 + \epsilon^4) \wedge L_\Phi^2 (\|s_t\|^2 + \epsilon^2/L_\Phi)} + \frac{L_2^2}{\epsilon^4 \ln m/\delta}\right) \\
\leq \sum_{t=0}^{T'-1} O\left(\frac{L_0 \ln(1/\delta) + L_0^2}{\epsilon^2 \kappa^{-2} \wedge L_\Phi^2 / \epsilon^2 L_\Phi} + \frac{L_2^2}{\epsilon^2 \ln m/\delta}\right) \\
\leq T' \cdot O\left(\frac{L_0 \ln(1/\delta) + L_0^2}{\epsilon^2 \kappa^{-2} \wedge L_\Phi^2 / \epsilon^2 L_\Phi} + \frac{L_2^2}{\epsilon^2 \ln m/\delta}\right) \\
\leq O\left(\sqrt{L_2 \kappa^1.5 \epsilon^{-3}} \cdot O\left(\frac{L_0^2 \ln^2 m}{\epsilon^2 \ln m/\delta}\right)\right) \\
\leq O\left(\frac{L_0^2 \kappa^4 \sqrt{L_2} \ln^2 m/\delta}{\epsilon^2}\right)
\end{align*}
\]

where (i) uses $\epsilon' = \frac{\epsilon}{\sqrt{33L_\Phi}} \leq \frac{\epsilon}{L_\Phi} = \frac{L_1(1+\kappa)}{L_\Phi}$ which implies that $\epsilon^4 \kappa^{-2} \leq O(L_2^2 \epsilon^2/L_\Phi)$, (ii) uses $T' \leq O\left(\sqrt{L_2 \kappa^1.5 \epsilon^{-3}}\right)$ we proved above. The number of Hessian computations has the following upper bound.

\[
\begin{align*}
\sum_{t=0}^{T'-1} \sum_{k=1}^{2} \sum_{l=1}^{2} |B_{k,l}(t)| \\
= \sum_{t=0}^{T'-1} O\left(\frac{L_2^4 \kappa^4}{L_\Phi (L_\Phi \|s_t\|^2 + \epsilon^2) \ln m/\delta}\right) \\
\leq \sum_{t=0}^{T'-1} O\left(\frac{L_2^4 \kappa^4}{L_\Phi \epsilon^2 \ln m/\delta}\right) \\
\leq T' \cdot O\left(\frac{L_2^4 \kappa^4 \ln m + n}{\epsilon^2 \ln m/\delta}\right)
\end{align*}
\]
Therefore, we can query the above Jacobian-vector product oracle to compute $G$

Then, we have

Theorem 3. The following high-probability convergence rate of SGA is a direct application of the Theorem 3.1 of [Harvey et al., 2019].

Tripuraneni et al., 2018]. In particular, when implementing these cubic-solvers to solve the above cubic sub-problem,

which by

µ

which by µ-strong concavity of $f(x_t, \cdot)$ proves the following convergence rate.

\[
\|y_{t+1} - y^*(x_t)\| \leq \mathcal{O}\left(\sqrt{\frac{L_0 \ln(1/\delta) + L_0^3}{\mu^2 N_t}}\right).
\]

Proof. Based on Theorem 3.1 of [Harvey et al., 2019], the inner SGA steps in Algorithm 2 has the following convergence rate with probability at least $1 - \delta$.

\[
f(x_t, y^*(x_t)) - f(x_t, y_{t+1}) \leq \mathcal{O}\left(\frac{L_0 \ln(1/\delta) + L_0^3}{\mu N_t}\right),
\]

which by $\mu$-strong concavity of $f(x_t, \cdot)$ proves the following convergence rate.

\[
\|y_{t+1} - y^*(x_t)\| \leq \sqrt{2(f(x_t, y^*(x_t)) - f(x_t, y_{t+1})/\mu} \leq \mathcal{O}\left(\sqrt{\frac{L_0 \ln(1/\delta) + L_0^3}{\mu^2 N_t}}\right).
\]

F Convergence Rate of SGA

The following high-probability convergence rate of SGA is a direct application of the Theorem 3.1 of [Harvey et al., 2019].

Theorem 3. Let Assumption 2 hold. For all $t, k$, assume that $\|\nabla_2 f(x_t, \tilde{y}_k)\| \leq L_0$ and $\|\nabla_2 f(x_t, \tilde{y}_k) - \nabla_2 f(x_t, \tilde{y}_k)\| \leq 1$ almost surely. The inner stochastic gradient ascent steps in Algorithm 2 converges at the following rate with probability at least $1 - \delta$.

\[
\|y_{t+1} - y^*(x_t)\| \leq \mathcal{O}\left(\sqrt{\frac{L_0 \ln(1/\delta) + L_0^3}{\mu^2 N_t}}\right).
\]

Proof. Based on Theorem 3.1 of [Harvey et al., 2019], the inner SGA steps in Algorithm 2 has the following convergence rate with probability at least $1 - \delta$.

\[
f(x_t, y^*(x_t)) - f(x_t, y_{t+1}) \leq \mathcal{O}\left(\frac{L_0 \ln(1/\delta) + L_0^3}{\mu N_t}\right),
\]

which by $\mu$-strong concavity of $f(x_t, \cdot)$ proves the following convergence rate.

\[
\|y_{t+1} - y^*(x_t)\| \leq \sqrt{2(f(x_t, y^*(x_t)) - f(x_t, y_{t+1})/\mu} \leq \mathcal{O}\left(\sqrt{\frac{L_0 \ln(1/\delta) + L_0^3}{\mu^2 N_t}}\right).
\]

G Compute Jacobian-Vector Product

In Algorithm 1 (and similarly for Algorithm 2), the key step is the cubic regularization update, which requires solving the following generic cubic sub-problem.

\[
\arg\min_s \nabla_1 f(x, y)^\top s + \frac{1}{2} s^\top G(x, y) s + \frac{1}{6\mu} \|s\|^3,
\]

where $G(x, y) = [\nabla_{11} f - \nabla_{12} f(\nabla_{22} f)^{-1}\nabla_{21} f](x, y)$.

In the literature, many gradient-based cubic-solvers have been developed to efficiently solve cubic sub-problems, and their convergence guarantee and computation complexity are well established [Carmon and Duchi, 2016, Tripuraneni et al., 2018]. In particular, when implementing these cubic-solvers to solve the above cubic sub-problem, one needs to query the Hessian-vector product $G(x, y) \cdot u$ for a general vector $u$. This quantity can be efficiently computed as we elaborate below.

First, note that for any vector $v$, the Jacobian-vector products $\nabla_{k\ell} f(x, y) \cdot v$ for all $k, \ell \in \{1, 2\}$ can be computed via standard auto-differentiation as follows

\[
\nabla_{k\ell} f(x, y) \cdot v = \nabla_{\ell}(\nabla_k f(x, y)^\top v), \quad \forall k, \ell \in \{1, 2\}.
\]

Therefore, we can query the above Jacobian-vector product oracle to compute

\[
z_1 := \nabla_{11} f(x, y) \cdot u, \quad z_2 := \nabla_{21} f(x, y) \cdot u.
\]

Then, we have $G(x, y) \cdot u = z_1 - \nabla_{12} f(\nabla_{22} f)^{-1}(x, y) \cdot z_2$. Note that the quantity $z_3 := (\nabla_{22} f(x, y))^{-1} \cdot z_2$ can be computed by solving the linear system

\[
\nabla_{22} f(x, y) \cdot z_3 = z_2
\]

using standard conjugate gradient solvers. Lastly, we query another Jacobian-vector product oracle to compute

\[
z_4 := \nabla_{12} f(x, y) \cdot z_3.
\]

Finally, the Hessian-vector product can be computed as $G(x, y) \cdot u = z_1 - z_4$. 

22