The Complexity of Nonconvex-Strongly-Concave Minimax Optimization

Siqi Zhang*  Junchi Yang*  Cristóbal Guzmán  Negar Kiyavash  Niao He

1University of Illinois at Urbana-Champaign (UIUC)  2University of Twente & Pontificia Universidad Católica de Chile  3École polytechnique fédérale de Lausanne (EPFL)  4ETH Zürich

Abstract

This paper studies the complexity for finding approximate stationary points of nonconvex-strongly-concave (NC-SC) smooth minimax problems, in both general and averaged smooth finite-sum settings. We establish nontrivial lower complexity bounds of $\Omega(\sqrt{n}\Delta L e^{-2})$ and $\Omega(n + \sqrt{n}\Delta L e^{-2})$ for the two settings, respectively, where $\kappa$ is the condition number, $L$ is the smoothness constant, and $\Delta$ is the initial gap. Our result reveals substantial gaps between these limits and best-known upper bounds in the literature. To close these gaps, we introduce a generic acceleration scheme that deploys existing gradient-based methods to solve a sequence of crafted strongly-convex-strongly-concave subproblems. In the general setting, the complexity of our proposed algorithm nearly matches the lower bound; in particular, it removes an additional poly-logarithmic dependence on accuracy present in previous works. In the averaged smooth finite-sum setting, our proposed algorithm improves over previous algorithms by providing a nearly-tight dependence on the condition number.

1 INTRODUCTION

In this paper, we consider general minimax problems of the form $(n, d_1, d_2) \in \mathbb{N}^+$:

$$\min_{x \in \mathbb{R}^{d_1}} \max_{y \in \mathbb{R}^{d_2}} f(x, y),$$

as well as their finite-sum counterpart:

$$\min_{x \in \mathbb{R}^{d_1}} \max_{y \in \mathbb{R}^{d_2}} f(x, y) \equiv \frac{1}{n} \sum_{i=1}^{n} f_i(x, y),$$

where $f, f_i$ are continuously differentiable and $f$ is $L$-Lipschitz smooth jointly in $x$ and $y$. We focus on the setting when $f$ is $\mu$-strongly concave in $y$ and perhaps nonconvex in $x$, i.e., $f$ is nonconvex-strongly-concave (NC-SC). Such problems arise ubiquitously in machine learning, e.g., GANs with regularization [Sanjabi et al., 2018, Lei et al., 2020], Wasserman robust models [Sinha et al., 2018], robust learning over multiple domains [Qian et al., 2019], and off-policy reinforcement learning [Dai et al., 2017, 2018, Huang and Jiang, 2020]. Since the problem is nonconvex in general, a natural goal is to find an approximate stationary point $\hat{x}$, such that $\|\nabla_x \Phi(\hat{x})\| \leq \epsilon$, for a given accuracy $\epsilon$, where $\Phi(\hat{x}) \equiv \max_y f(x, y)$ is the primal function. This goal is meaningful for the aforementioned applications, e.g., in adversarial models the primal function quantifies the worst-case loss for the learner, with respect to adversary’s actions.

There exists a number of algorithms for solving NC-SC problems in the general setting, including GDmax [Nouiehed et al., 2019], GDA [Lin et al., 2020a], alternating GDA [Yang et al., 2020a, Boj and Böhm, 2020], Xi et al., 2020], Minimax-PPA [Lin et al., 2020b]. Specifically, GDA and its alternating variant both achieve the complexity of $O(\kappa^2 \Delta L e^{-2})$ [Lin et al., 2020a, Yang et al., 2020a], where $\kappa \equiv \frac{n}{L}$ is the condition number and $\Delta \equiv \Phi(x_0) - \inf_x \Phi(x)$ is the initial function gap. Recently, [Lin et al., 2020b] provided the best-known complexity of $O(\sqrt{n}\Delta L e^{-2} \cdot \log^2(\frac{\kappa L}{\epsilon}))$ achieved by Minimax-PPA, which improves the dependence on the condition number but suffers from an extra poly-logarithmic factor in $\frac{1}{\epsilon}$.

In the finite-sum setting, several algorithms have been proposed recently, e.g., PGSMD [Rafique et al., 2018], SGDMX [Lin et al., 2020b], Stochastic GDA [Lin et al., 2020a], SREDAs [Luo et al., 2020]. In particular, [Lin et al., 2020a] proved that Stochastic GDA attains the complexity of $O(n^2 \kappa^{-2} e^{-4})$. [Luo et al., 2020] recently showed the state-of-the-art result achieved by SREDA: when $n \geq \kappa^2$, the complexity is $O(n \log \frac{\kappa}{\epsilon} + \sqrt{n}\kappa^2 \Delta L e^{-2})$, which is sharper than the batch Minimax-PPA algorithm; when $n \leq \kappa^2$, the complexity is $O((nk + \kappa^2) \Delta L e^{-2})$, which is sharper than...
Despite this active line of research, whether these state-of-the-art complexity bounds can be further improved remains elusive. As a special case by restricting the domain of y to a singleton, lower bounds for nonconvex smooth minimization, e.g., [Carmon et al., 2019a,b, Fang et al., 2018; Zhou and Gu, 2019; Arjevani et al., 2019], hardly capture the dependence on the condition number $\kappa$, which plays a crucial role in the complexity for general NC-SC smooth minimax problems. In many of the aforementioned machine learning applications, the condition number is inversely proportional the regularization parameter, and can be very large in practice. For example, in statistical learning, where $n$ represents the sample size, the optimal regularization parameter (i.e., with optimal empirical/generalization trade-off) leads to $\kappa = \Omega(\sqrt{n})$ [Shalev-Shwartz and Ben-David, 2014].

This motivates the following fundamental questions: What is the complexity limit for NC-SC problems in the general and finite-sum settings? Can we design new algorithms to meet the performance limits and attain optimal dependence on the condition number?

### 1.1 CONTRIBUTIONS

Our contributions, summarized in Table 1, are as follows:

- We establish nontrivial lower complexity bounds for finding an approximate stationary point of nonconvex-strongly-concave (NC-SC) minimax problems. In the general setting, we prove an $\Omega(\sqrt{\kappa}L\epsilon^{-2})$ lower complexity bound which applies to arbitrary deterministic linear-span algorithms interacting with the classical first-order oracle. In the finite-sum setting, we prove an $\Omega(n^{1/3}\sqrt{\kappa}DL\epsilon^{-2})$ lower complexity bound (when $\kappa = \Omega(n)$) for the class of averaged smooth functions and arbitrary linear-span algorithms interacting with a (randomized) incremental first-order oracle (precise definitions in Sections 2 and 3).

Our lower bounds build upon two main ideas: first, we start from an NC-SC function whose primal function mimics the lower bound construction in smooth nonconvex minimization [Carmon et al., 2019a]. Crucially, the smoothness parameter of this primal function is boosted by an $\Omega(\kappa)$ factor, which strengthens the lower bound. Second, the function has an alternating zero-chain structure, as utilized in lower bounds for convex-concave settings [Ouyang and Xu, 2019]. The combination of these features leads to a hard instance for our problem.

- To bridge the gap between the lower bounds and existing upper bounds in both settings, we introduce a generic Cat-

1A concurrent work by [Han et al., 2021] provided a similar lower bound result for finite-sum NC-SC problems under probabilistic arguments based on geometric random variables.

### 1.2 RELATED WORK

**Lower bounds for minimax problems.** Information-based complexity (IBC) theory [Traub et al., 1988], which derives the minimal number of oracle calls to attain an approximate solution with a desired accuracy, is often used in lower bound analysis of optimization algorithms. Unlike the case of minimization, e.g., [Nemirovskii and Yudin, 1983; Carmon et al., 2019a,b; Arjevani et al., 2019], lower bounds for minimax optimization are far less understood; only a few recent works provided lower bounds for finding an approximate saddle point of (strongly)-convex-(strongly)-concave minimax problems [Ouyang and Xu, 2019; Zhang et al., 2019; Ibrahim et al., 2020; Xie et al., 2020; Yoon and Ryu, 2021]. Instead, this paper considers lower bounds for NC-SC problems of finding an stationary point, which requires different techniques for constructing zero-chain properties. Note that there exists another line of research on the purely stochastic setting, e.g., [Rafique et al., 2018; Luo et al., 2020]; constructing lower bounds in that setting is out of the scope of this paper.

**Complexity of making gradient small.** In nonconvex optimization, most lower and upper complexity bound results are presented in terms of the gradient norm (see a recent survey [Danilova et al., 2020] and references therein for more details). For convex optimization, the optimality gap based on the objective value is commonly used as the convergence criterion. The convergence in terms of gradient norm, albeit easier to check, are far less studied in the literature until recently; see e.g., [Nesterov, 2012; Allen-Zhu, 2018; Diakonikolas and Guzman, 2021] for convex minimization and [Diakonikolas, 2020; Yoon and Ryu, 2021] for convex-concave smooth minimax problems.

**Nonconvex minimax optimization.** In NC-SC setting, as we mentioned, there has been several substantial works. Among them, [Lin et al., 2020b] achieved the best depen-
we say $f = O(g)$ if $f(x) \leq cg(x)$ for some $c > 0$, and further write $f = \tilde{O}(g)$ to omit poly-logarithmic terms on constants $L, \mu$ and $\kappa$, while $f = \Omega(g)$ if $f(x) \geq cg(x)$ (see more in Appendix A).

We introduce definitions and assumptions used throughout.

**Definition 2.1 (Primal and Dual Functions)** For a function $f(x,y)$, we define $\Phi(x) \triangleq \max_y f(x,y)$ as the primal function, and $\Psi(y) \triangleq \min_x f(x,y)$ as the dual function. We also define the primal-dual gap at a point $(\bar{x}, \bar{y})$ as $\text{gap}_f(\bar{x}, \bar{y}) \triangleq \max_{y \in \mathbb{R}^d} f(\bar{x}, y) - \min_{x \in \mathbb{R}^d} f(x, \bar{y})$.

**Definition 2.2 (Lipschitz Smoothness)** We say a function $f(x,y)$ is $L$-Lipschitz smooth ($L$-S) jointly in $x$ and $y$ if it is differentiable and for any $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, $\|\nabla_x f(x_1, y_1) - \nabla_x f(x_2, y_2)\| \leq L(\|x_1 - x_2\| + \|y_1 - y_2\|)$ and $\|\nabla_y f(x_1, y_1) - \nabla_y f(x_2, y_2)\| \leq L(\|x_1 - x_2\| + \|y_1 - y_2\|)$, for some $L > 0$.

**Definition 2.3 (Average / Individual Smoothness)** We say $f(x,y) = \frac{1}{n} \sum_{i=1}^n f_i(x,y)$ or $\{f_i\}_{i=1}^n$ is $L$-averaged smooth ($L$-AS) if each $f_i$ is differentiable, and for any $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, we have

$$\frac{1}{n} \sum_{i=1}^n \|\nabla f_i(x_1, y_1) - \nabla f_i(x_2, y_2)\|^2 \leq L^2(\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2).$$

We say $f$ or $\{f_i\}_{i=1}^n$ is $L$-individually smooth ($L$-IS) if each $f_i$ is $L$-Lipschitz smooth.

Average smoothness is a weaker condition than the common Lipschitz smoothness assumption of each component in finite-sum / stochastic minimization [Yang et al. 2018, Zhou and Gu, 2019]. Similarly in minimax problems, the following proposition summarizes the relationship among these different notions of smoothness.

**Proposition 2.1** Let $f(x,y) = \frac{1}{n} \sum_{i=1}^n f_i(x,y)$. Then we have: (a) If the function $f$ is $L$-IS or $L$-AS, then it is $L$-S. (b) If $f$ is $L$-IS, then it is $(2L)$-AS. (c) If $f$ is $L$-AS, then $f(x,y) + \frac{\tau_x}{2} \|x - \bar{x}\|^2 - \frac{\tau_y}{2} \|y - \bar{y}\|^2 \leq \nabla f(x,y)^T (\bar{x} - x) + \nabla f(x,y)^T (\bar{y} - y)$ for any $\bar{x}$ and $\bar{y}$.
We study the lower bound of finding primal stationary point
Another commonly used criterion is the stationarity of
• For the general setting, we consider the
Definition 2.5 (Convergence Criterion) For a differen-
We assume that
Under Assumption 2.1, the primal function
Definition 2.4 (Strong Convexity) A differentiable func-
1.1, we will use the zero-chain argument for the analysis.
Algorithm class In this work, we consider the class of
Algorithm class In this work, we consider the class of
Complexity measures The efficiency of algorithms is
Complexity measures The efficiency of algorithms is
3 LOWER BOUNDS FOR NC-SC MINIMAX PROBLEMS
In this section, we establish lower complexity bounds (LB)
3.1 FRAMEWORK AND SETUP
We study the lower bound of finding primal stationary point
Function class We consider the nonconvex-strongly-
Oracle class We consider different oracles for the general
• For the finite-sum setting, incremental first-order oracle (IFO) is often used in lower bound analysis [Agarwal
For the finite-sum setting, incremental first-order oracle (IFO) is often used in lower bound analysis [Agarwal
Algorithm class In this work, we consider the class of
Linear-span algorithms existing deterministic and randomized linear-span algorithms. We distinguish the general and finite-sum setting by specifying the used oracle, which is \( O_{IFO} \) or \( O_{IFO} \), respectively. Most existing first-order algorithms, including simultaneous and alternating update algorithms, can be formulated as linear-span algorithms. It is worth pointing out that typically the linear span assumption is used without loss of generality, since there is a standard reduction from deterministic linear-span algorithms to arbitrary oracle based deterministic algorithms [Nemirovsky, 1991, 1992, Ouyang and Xu, 2019]. We defer this extension for future work.

3.1 FRAMEWORK AND SETUP
We study the lower bound of finding primal stationary point
Function class We consider the nonconvex-strongly-
Oracle class We consider different oracles for the general
• For the general setting, we consider the first-order oracle (FO), denoted as \( O_{FO}(f, \cdot) \), that for each query on point \( z \), it returns the gradient \( O_{FO}(f, z) \triangleq (\nabla_x f(x, y), \nabla_y f(x, y)) \).

For the finite-sum setting, \( f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x, y) \) is such that for each query on point \( z \) and given \( i \in [n] \), it returns the gradient of the \( i \)-th component, i.e., \( O_{IFO}(f, z, i) \triangleq (\nabla_{x_i} f_i(x, y), \nabla_{y_i} f_i(x, y)) \). Here, we consider averaged smooth IFO and individually smooth IFO, denoted as \( O_{IFO}^{\text{AS}}(f) \) and \( O_{IFO}^{\text{AS}}(f) \), where \( \{f_i\}_{i=1}^n \) is L-AS or L-IS, respectively.

Algorithm class In this work, we consider the class of
Complexity measures The efficiency of algorithms is
Complexity measures The efficiency of algorithms is

\[ T_\epsilon(f, A) \triangleq \inf \{ T \in \mathbb{N} \mid \|\nabla \Phi(x^T)\| \leq \epsilon \} \]

as the minimum number of oracle calls \( A \) makes to reach stationarity convergence. For the general case, we define the worst-case complexity

\[ \text{Compl} \_{c}(f, g) \triangleq \sup \left\{ f \in F, \; g \in A(\Omega) \right\} \inf \left\{ T_\epsilon(f, A) \right\} \]

For finite-sum cases, we lower bound the randomized complexity by the distributional complexity \([Braun et al., 2017]\):

\[ \text{Compl}_r(f, g) \triangleq \sup \left\{ f \in F, \; g \in A(\Omega) \right\} \inf \left\{ \mathbb{E} T_\epsilon(f, A) \right\} \]

Following the motivation of analysis discussed in Section 1.1, we will use the zero-chain argument for the analysis. First we define the notion of (first-order) zero-chain \([Carmon et al., 2019b]\) and activation as follows.

Definition 3.1 (Zero Chain, Activation) A function \( f : \mathbb{R}^d \to \mathbb{R} \) is a first-order zero-chain if for any \( x \in \mathbb{R}^d \),

\[ \text{supp}(x) \subseteq \{1, \ldots, i-1\} \Rightarrow \text{supp}(\nabla f(x)) \subseteq \{1, \ldots, i\}, \]

\[ T_\epsilon(f, A) \triangleq \inf \{ T \in \mathbb{N} \mid \|\nabla \Phi(x^T)\| \leq \epsilon \} \]

as the minimum number of oracle calls \( A \) makes to reach stationarity convergence. For the general case, we define the worst-case complexity

\[ \text{Compl} \_{c}(f, A, \Omega) \triangleq \sup \left\{ f \in F, \; A \in A(\Omega) \right\} \inf \left\{ T_\epsilon(f, A) \right\} \]

For finite-sum cases, we lower bound the randomized complexity by the distributional complexity \([Braun et al., 2017]\):

\[ \text{Compl}_r(f, A, \Omega) \triangleq \sup \left\{ f \in F, \; A \in A(\Omega) \right\} \inf \left\{ \mathbb{E} T_\epsilon(f, A) \right\} \]

Following the motivation of analysis discussed in Section 1.1, we will use the zero-chain argument for the analysis. First we define the notion of (first-order) zero-chain \([Carmon et al., 2019b]\) and activation as follows.

Definition 3.1 (Zero Chain, Activation) A function \( f : \mathbb{R}^d \to \mathbb{R} \) is a first-order zero-chain if for any \( x \in \mathbb{R}^d \),

\[ \text{supp}(x) \subseteq \{1, \ldots, i-1\} \Rightarrow \text{supp}(\nabla f(x)) \subseteq \{1, \ldots, i\}, \]
where \( \text{supp}(x) \triangleq \{ i \in [d] \mid x_i \neq 0 \} \) and \([d] = \{ 1, \cdots, d \}\).

For an algorithm initialized at \( o \in \mathbb{R}^d \), with iterates \( \{ x^t \}_t \), we say coordinate \( i \) is activated at \( x^t \), if \( x^t_i \neq 0 \) and \( x^t_s = 0 \), for any \( s < t \).

### 3.2 General NC-SC Problems

First we consider the general NC-SC (Gen-NC-SC) minimax optimization problems. Following the above framework, we choose function class \( \mathcal{F}_{\text{NCSC}} \), oracle \( \mathcal{O}_F \), linear-span algorithms \( \mathcal{A} \), and we analyze the complexity defined in (5).

#### Hard Instance Construction

Inspired by the hard instances constructed in [Ouyang and Xu 2019 Carmon et al. 2019b], we introduce the following function \( F_d : \mathbb{R}^{d+1} \times \mathbb{R}^{d+2} \rightarrow \mathbb{R} \) (\( d \in \mathbb{N}^+ \)) and

\[
F_d(x, y; \lambda, \alpha) \triangleq \lambda_1 (B_d x, y - \lambda_2 x_j^2 - \alpha x_j^2) + \lambda_3 \sqrt{\alpha} \langle e_1, x \rangle,
\]

where \( \lambda = (\lambda_1, \lambda_2) \in \mathbb{R}^2 \) is the parameter vector, \( e_1 \in \mathbb{R}^{d+1} \) is the unit vector with the only non-zero element in the first dimension, \( \Gamma : \mathbb{R} \rightarrow \mathbb{R} \) and \( B_d \in \mathbb{R}^{(d+2) \times (d+1)} \) are

\[
B_d = \begin{bmatrix}
1 & & & & \cdots & & 1 \\
& 1 & & & & \cdots & & 1 \\
& & \ddots & & & & \ddots & & 1 \\
1 & & & & \cdots & & 1 \\
\end{bmatrix},
\]

\[
\Gamma(x) = 120 \int_{t \in [t]} \frac{t^2 (t-1)}{1 + t^2} dt.
\]

Matrix \( B_d \) essentially triggers the activation of variables at each iteration, and function \( \Gamma \) introduces nonconvexity in \( x \) to the instance. By the first-order optimality condition of \( F_d(x, \cdot ; \lambda, \alpha) \), we can compute its primal function, \( \Phi_d \):

\[
\Phi_d(x; \lambda, \alpha) = \max_{y \in \mathbb{R}^{d+2}} F_d(x, y; \lambda, \alpha)
\]

\[
= \frac{\lambda_3^2}{2\lambda_2} \left( \sum_{i=1}^{d} -1 \right) + \frac{\sqrt{\alpha}}{2} \left( \sum_{i=1}^{d} \Gamma(x_i) + \frac{1 - \alpha}{2} x_i^2 \right),
\]

where \( A_d \in \mathbb{R}^{(d+1) \times (d+1)} \) is

\[
A_d = \left[ \begin{array}{cccc}
1 + \sqrt{\alpha} & -1 & \cdots & -1 \\
-1 & 2 & \cdots & -1 \\
\cdots & \cdots & \cdots & \cdots \\
-1 & 2 & \cdots & -1 \\
1 & -1 & \cdots & 1 \\
\end{array} \right].
\]

The resulting primal function resembles the worst-case functions used in lower bound analysis of minimization problems [Nesterov 2018 Carmon et al. 2019b].

#### Zero-Chain Construction

First we summarize key properties of the instance and its zero-chain mechanism. We further denote \( \hat{e}_d \in \mathbb{R}^{d+2} \) as the unit vector for the variable \( y \) and define \( k \geq 1 \)

\[
X_k \triangleq \text{Span}\{ e_1, e_2, \cdots, e_k \}, \quad X_0 \triangleq \{ 0 \},
\]

\[
Y_k \triangleq \text{Span}\{ \hat{e}_{d+2}, \hat{e}_{d+1}, \cdots, \hat{e}_{d-k+2} \}, \quad Y_0 \triangleq \{ 0 \},
\]

then we have the following properties for \( F_d \).

**Lemma 3.1 (Properties of \( F_d \))** For any \( d \in \mathbb{N}^+ \) and \( \alpha \in [0, 1] \), \( F_d(x, y; \lambda, \alpha) \) in (7) satisfies:

(i) The function \( F_d(x, \cdot ; \lambda, \alpha) \) is \( L_F \)-Lipschitz smooth where \( L_F = \max \left\{ \frac{200 \lambda_3^2}{\lambda_2}, 2 \lambda_1, 2 \lambda_2 \right\} \).

(ii) For each fixed \( x \in \mathbb{R}^{d+1} \), \( F_d(x, \cdot ; \lambda, \alpha) \) is \( \mu_F \)-strongly concave where \( \mu_F = 2 \lambda_2 \).

The proof of Lemma 3.1 is deferred to Appendix C.1.1.

The first two properties show that function \( F_d \) is Lipschitz smooth and NC-SC; the third property above suggests that, starting from \( (x, y) = (0, 0) \), the activation process follows an “alternating zero-chain” form [Ouyang and Xu 2019]. That is, for a linear-span algorithm, when \( x \in X_k \), \( y \in Y_k \), the next iterate will at most activate the \((k+1)\)-th coordinate of \( x \) while keeping \( y \) fixed; similarly when \( x \in X_{k+1} \), \( y \in Y_{k+1} \), the next iterate will at most activate the \((d-k+1)\)-th element of \( y \). We need the following properties of \( \Phi_d \) for the lower bound argument.

**Lemma 3.2 (Properties of \( \Phi_d \))** For any \( \alpha \in [0, 1] \) and \( x \in \mathbb{R}^{d+1} \), if \( x_d = x_{d+1} = 0 \), we have:

(i) \( \| \nabla \Phi_d(x; \lambda, \alpha) \| \geq \frac{\lambda_3}{\alpha \lambda_2} \alpha^{3/4} \).

(ii) \( \Phi_d(0; \lambda, \alpha) - \inf_{x \in \mathbb{R}^{d+1}} \Phi_d(x; \lambda, \alpha) \leq \frac{\lambda_3}{\alpha \lambda_2} \left( \sqrt{\alpha} + 10 \alpha d \right) \).

We defer the proof of Lemma 3.2 to Appendix C.1.2. This lemma indicates that, starting from \( (x, y) = (0, 0) \) with appropriate parameter settings, the primal function \( \Phi_d \) will not approximate stationarity until the last two coordinates are activated. Now we are ready to present our final lower bound result for the general NC-SC case.
Theorem 3.1 (LB for Gen-NC-SC) For the linear-span first-order algorithm class $\mathcal{A}$, parameters $L, \mu, \Delta > 0$, and accuracy $\epsilon$ satisfying $\epsilon^2 \leq \min \left(\frac{\Delta}{5400}, \frac{\Delta \sqrt{\epsilon}}{38400}\right)$, we have

$$\text{Compl}_\epsilon \left(\mathcal{F}^{L, \Delta}_{\text{NCSC}}, \mathcal{A}, \Omega_{\text{IFO}}\right) = \Omega\left(\sqrt{\kappa} \Delta \Lambda \epsilon^{-2}\right).$$  \hspace{1cm} (12)

The hard instance in the proof is established based on $\mathcal{F}_d$ in (7). We choose the scaled function $f(x, y) = \eta^2 F_d (\frac{x}{\eta}, \frac{y}{\eta}; \lambda^*, \alpha)$ as the final hard instance, which preserves the smoothness and strong convexity (by Lemma B.3), while appropriate setting of $\eta$ will help to fulfill the requirements on the initial gradient and large gradient norm (before thorough activation) of the primal function. The detailed statement and proof of Theorem 3.1 are presented in Appendix C.1.3.

Remark 3.1 (Tightness) The best-known upper bounds for general NC-SC problems are $O(\Delta \Lambda \kappa^{2} \epsilon^{-2})$ [Lin et al., 2020a; Bot and Böhm, 2020] and $O(\Delta \sqrt{\kappa} \Lambda \epsilon^{-2} \log \frac{1}{\epsilon})$ [Lin et al., 2020b]. Therefore, our result exhibits significant gaps in terms of the dependence on $\epsilon$ and $\kappa$. In order to mitigate these gaps, we propose faster algorithms in Section 4. On the other hand, compared to the $\Omega(\Delta \Lambda \epsilon^{-2})$ lower bound for nonconvex smooth minimization [Carmon et al., 2019a], our result reveals an explicit dependence on $\kappa$.

3.3 FINITE-SUM NC-SC PROBLEMS

The second case we consider is finite-sum NC-SC (FS-NC-SC) minimax problems, for the function class $\mathcal{F}^{L, \Delta}_{\text{NCSC}}$ , the linear-span algorithm class $\mathcal{A}$ and the averaged smooth IFO class $\Omega_{\text{IFO}}^{L, \Lambda, \kappa}$. The complexity is defined in (6).

Hard instance construction To derive the finite-sum hard instance, we modify $F_d$ in (7) with orthogonal matrices defined as follows.

Definition 3.2 (Orthogonal Matrices) For positive integers $a, b, n \in \mathbb{N}^+$, we define a matrix sequence $\{U(i)\}^n_{i=1} \in \text{Orth}(a, b, n)$ if for each $i, j \in \{1, \cdots, n\}$ and $i \neq j$, $U(i)$, $U(j)$ are in $\mathbb{R}^{a \times b}$ and $U(i)(U(j))^\top = \mathbb{I} \in \mathbb{R}^{a \times a}$ and $U(i)(U(i))^\top = 0 \in \mathbb{R}^{a \times a}$.

Here the intuition for the finite-sum hard instance is combining $n$ independent copies of the hard instance in the general case (7), then appropriate orthogonal matrices will convert the $n$ independent variables with dimension $d$ into one variable with dimension $n \times d$, which results in the desired hard instance. To preserve the zero property, for $\{U(i)\}^n_{i=1} \in \text{Orth}(d+1, n(d+1), n)$, $\{V(i)\}^n_{i=1} \in \text{Orth}(d+2, n(d+2), n)$, $\forall n, d \in \mathbb{N}^+$ and $x \in \mathbb{R}^{n(d+1)}$, $y \in \mathbb{R}^{n(d+2)}$, we set $U(i)$ and $V(i)$ by concatenating $n$ matrices:

$$U(i) = \begin{bmatrix} 0_{d+1} & \cdots & 0_{d+1} & 1_{d+1} & 0_{a+1} & \cdots & 0_{d+1} \end{bmatrix},$$

$$V(i) = \begin{bmatrix} 0_{d+2} & \cdots & 0_{d+2} & 1_{d+2} & 0_{d+2} & \cdots & 0_{d+2} \end{bmatrix},$$ \hspace{1cm} (13)

where $0_d, I_d \in \mathbb{R}^{d \times d}$ are the zero and identity matrices respectively, while the $i$-th matrix above is the identity matrix. Hence, $U(i)x$ will be the $(id - d + 1)$-th to the $(id)$-th elements of $x$, similar property also holds for $V(i)y$.

The hard instance construction here follows the idea of that in the deterministic hard instance (7), the basic motivation is that its primal function will be a finite-sum form of the primal function $\Phi_d$ defined in the deterministic case (9). We choose the following functions $H_d : \mathbb{R}^{d+1} \times \mathbb{R}^{d+2} \rightarrow \mathbb{R}$, $\Gamma_d : \mathbb{R}^{n(d+1)} \rightarrow \mathbb{R}$ and

$$H_d(x, y; \lambda, \alpha) \triangleq \lambda_1 \langle B_{\alpha} x, y \rangle - \lambda_2 \langle y, 2 \lambda_3 \alpha \rangle - \frac{\lambda_2^2 \alpha}{4 \lambda_2} x^2 + \frac{\lambda_2^2 \alpha}{4 \lambda_2},$$

$$\Gamma_d^n(x) \triangleq \sum_{i=1}^{n} \sum_{j=(i(d+1)-d)}^{i(d+1)-1} \Gamma(x_j),$$

Note that by denoting $\Gamma_d(x) \triangleq \sum_{i=1}^{d} \Gamma_d^n(x_i)$, it is easy to find that

$$\Gamma_d^n(x) = \sum_{i=1}^{n} \sum_{j=(i(d+1)-d)}^{i(d+1)-1} \Gamma(x_j) = \sum_{i=1}^{n} \sum_{j=1}^{d} \Gamma\left(\left(U(i)x\right)_j\right).$$ \hspace{1cm} (16)

Define $u(i) \triangleq U(i)x$, we summarize the properties of the above functions in the following lemma.

Lemma 3.3 (Properties of $f$) For the above functions $\{f_i\}_i$ and $f$ in (15), they satisfy that:

(i) $\{f_i\}_i$ is $L_F$-average smooth where $L_F = \sqrt{\frac{n}{2}} \max \left\{16 \lambda_1^2 + 8 \lambda_2^2, \frac{C_2 \lambda_1^2 \alpha^2}{n \lambda_2^2} + \frac{\lambda_1 \alpha^2}{2 \lambda_2} + 8 \lambda_1^2\right\}$.

(ii) $f$ is $\mu_F$-strongly concave on $y$ where $\mu_F = \frac{2 \lambda_2}{n}$.

(iii) For $n \in \mathbb{N}^+$, $L \geq 2n \mu > 0$, if we set $\lambda = \lambda^* \approx (\lambda_1^* + \lambda_2^* = \sqrt{\frac{20L}{n \mu}} + \frac{n \mu}{50L} \in [0, 1]$, then $\{f_i\}_i$ is L-AS and $f$ is $\mu$-strongly concave on $y$. 


(iv) With $\Phi_d$ is defined in (9), let $\Phi(x)\triangleq \max_y \tilde{f}(x, y)$, then we have
\[ \Phi(x) = \frac{1}{n} \sum_{i=1}^{n} \Phi_i(x), \quad \Phi_i(x) \triangleq \Phi_d(U(i))x. \]  
We defer the proof of Lemma 3.3 to Appendix C.2. From (15), combined with a scaling trick similar to the one in the hard instance construction is based on $\bar{\Omega}$ explicitly the dependence on $\Omega(\cdot)$, Note that there is still a large gap be-
to those in existing literature, e.g. [Zhou and Gu, 2019, 2021]. Our scheme is distinct from these work in two aspects: (i) we introduce a new subroutine to solve regularized subproblems of the form:
\[ \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^d} f(x, y) + L\|x - \bar{x}\|^2 - \frac{\tau}{2} \|y - \bar{y}\|^2, \]
where $\bar{x}$ and $\bar{y}$ are carefully chosen prox-centers, and the parameter $\tau \geq 0$ is selected such that the condition numbers for $x$-component and $y$-component of these subproblems are well-balanced. Since $f$ is $L$-Lipschitz smooth and $\mu$-strongly concave in $y$, the above auxiliary problem is $L$-strongly convex in $x$ and $(\mu + \tau)$-strongly concave in $y$. Therefore, it can be solved by a wide family of off-the-shelf first-order algorithms with linear convergence rate.

Now we arrive at our final theorem for the averaged smooth FS-NC-SC case as follows.

**Theorem 3.2 (LB for AS FS-NC-SC)** For the linear-span algorithm class $A$, parameters $L, \mu, \Delta > 0$ and component size $n \in \mathbb{N}^+$, if $L \geq 2\mu > 0$, the accuracy $\epsilon$ satisfies that $\epsilon^2 \leq \min\left(\frac{\sqrt{n}L^2\Delta}{2880\mu^2}, \frac{\alpha L^2}{128\mu^2}, \frac{L^2\Delta}{\mu}\right)$ where $\alpha = \frac{\mu}{n\kappa}$, then we have
\[ \text{Compl} \left( F_{\mathbb{R}^d}, A, \Omega_{\text{IFO}}^{L, \Delta} \right) = \Omega(n + \sqrt{n}\kappa\Delta\epsilon^{-2}). \]  

Theorem above indicates that for any $A \in \mathcal{A}$, we can construct a function $f(x, y) = \frac{1}{n} \sum_{i=1}^{n} f_x(x, y)$, such that $f \in F_{\mathbb{R}^d}$ and $\{f_x\}$ is $L$-ASF, and $A$ requires at least $\Omega(n + \sqrt{n}\kappa\Delta\epsilon^{-2})$ IFO calls to attain an approximate stationary point of its primal function (in terms of expectation). The hard instance construction is based on $f$ and $f_x$ above [15], combined with a scaling trick similar to the one in the general case. Also we remark that lower bound holds for small enough $\epsilon$, while the require on $\epsilon$ is comparable to those in existing literature, e.g. [Zhou and Gu, 2019, 2021]. The detailed statement and proof of the theorem are deferred to Appendix C.2.2.

**Remark 3.2 (Tightness)** The state-of-the-art upper bound for NC-SC finite-sum problems is $O(n + \sqrt{n}\kappa^2\Delta\epsilon^{-2})$ when $n \geq \kappa^2$ and $O((nk + \kappa^2)\Delta\epsilon^{-2})$ when $n \leq \kappa^2$ [Luo et al., 2020]. Note that there is still a large gap between upper and lower bounds on the dependence in terms of $\kappa$ and $n$, which motivates the design of faster algorithms for FS-NC-SC case, we address this in Section 4. Note that a weaker result on the lower bound of nonconvex finite-sum averaged smooth minimization is $\Omega(\sqrt{n}\Delta\epsilon^{-2})$ [Fang et al., 2018, Zhou and Gu, 2019]; here, our result presents explicitly the dependence on $\kappa$.

## 4 Faster Algorithms for NC-SC Minimax Problems

In this section, we introduce a generic Catalyst acceleration scheme that turns existing optimizers for (finite-sum) SC-SC minimax problems into efficient, near-optimal algorithms for (finite-sum) NC-SC minimax optimization. Rooted in the inexact accelerated proximal point algorithm, the idea of Catalyst acceleration was introduced in [Lin et al., 2020] for convex minimization and later extended to nonconvex minimization in [Paquette et al., 2018] and nonconvex-concave minimax optimization in [Yang et al., 2020b]. In stark contrast, we focus on NC-SC minimax problems.

The backbone of our Catalyst framework is to repeatedly solve regularized subproblems of the form:
\[ \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^d} f(x, y) + L\|x - \bar{x}\|^2 - \frac{\tau}{2} \|y - \bar{y}\|^2, \]
where $\bar{x}$ and $\bar{y}$ are carefully chosen prox-centers, and the parameter $\tau \geq 0$ is selected such that the condition numbers for $x$-component and $y$-component of these subproblems are well-balanced. Since $f$ is $L$-Lipschitz smooth and $\mu$-strongly concave in $y$, the above auxiliary problem is $L$-strongly convex in $x$ and $(\mu + \tau)$-strongly concave in $y$. Therefore, it can be solved by a wide family of off-the-shelf first-order algorithms with linear convergence rate.

Our Catalyst framework, presented in Algorithm 1, consists of three crucial components: an inexact proximal point step for primal update, an inexact accelerated proximal point step for dual update, and a linear-convergent algorithm for solving the subproblems.

**Inexact proximal point step in the primal.** For the $x$-update in the outer loop, $\{x_t\}^\infty_{t=0}$, can be viewed as applying an inexact proximal point method to the primal function $\Phi(x)$, requiring to solve a sequence of auxiliary problems:
\[ \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^d} \bar{f}_t(x, y) \triangleq f(x, y) + L\|x - x_0\|^2. \]  

Inexact proximal point methods have been explored in minimax optimization in several work, e.g. [Lin et al., 2020, Rafique et al., 2018]. Our scheme is distinct from these work in two aspects: (i) we introduce a new subroutine to approximately solve the auxiliary problems with near-optimal complexity, and (ii) the inexactness is measured by an adaptive stopping criterion using the gradient norms:
\[ \|\nabla \bar{f}_t(x_t + 1, y_t + 1)\|^2 \leq \alpha_t \|\nabla \bar{f}_t(x_0, y_0)\|^2, \]  
where $\{\alpha_t\}_t$ is carefully chosen. Using the adaptive stopping criterion significantly reduces the complexity of solving the auxiliary problems. We will show that the number of steps required is only logarithmic in $L, \mu$ without any dependence on target accuracy $\epsilon$. Although the auxiliary
Algorithm 1 Catalyst for NC-SC Minimax Problems

**Input:** objective $f$, initial point $(x_0, y_0)$, smoothness constant $L$, strong-concavity constant $\mu$, and a parameter $\tau > 0$.

1. Let $(x_0, y_0) = (x_0, y_0)$ and $q = \frac{\mu}{\mu + \tau}$.
2. **for all** $t = 0, 1, \ldots, T$ **do**
3.   Let $z_1 = y_0$ and $k = 1$.
4.   Let $\hat{f}_t(x, y) \triangleq f(x, y) + L\|x - x_0\|^2$.
5.   **repeat**
6.     Find inexact solution $(x_k^{i}, y_k^{i})$ to the problem below by algorithm $\mathcal{M}$ with initial point $(x_{k-1}^{i}, y_{k-1}^{i})$:
7.     $\min_{x \in \mathbb{R}^d} \max_{y \in \mathbb{R}^d} \left[ \hat{f}_t(x, y) \right] \triangleq f(x, y) + L\|x - x_0\|^2 - \frac{\tau}{2}\|y - z_k\|^2$.
8.     **such that** $\|\nabla \hat{f}_t(x_k^{i}, y_k^{i})\| \leq \epsilon_k$.
9.     **until** $\|\nabla \hat{f}_t(x_k^{i+1}, y_k^{i+1})\| \leq \alpha_t \|\nabla \hat{f}_t(x_0^2, y_0^2)\|$.
10. **set** $(x_{k+1}^{i+1}, y_{k+1}^{i+1}) = (x_k^{i+1}, y_k^{i+1})$.
11. **end for**

**Output:** $\hat{x}_T$, which is uniformly sampled from $x_1^0, \ldots, x_T^0$.

If $\mathcal{M}$ is a deterministic algorithm; or taking expectation to the left-hand side above if $\mathcal{M}$ is randomized. The choices for $\mathcal{M}$ include, but are not limited to, extragradient (EG) [Tseng, 1995], optimistic gradient descent ascent (OGDA) [Gidel et al., 2018], SVRG [Balamurugan and Bach, 2016], SPIDER [Tan et al., 2018], SVRE [Chavdarova et al., 2019], Point-SAGA [Luo et al., 2019], and variance reduced prox-method [Carmon et al., 2019c]. For example, in the case of EG, $\mathbb{E}[\mathcal{M}_\mu, L](\tau) = \frac{L + \max\{2L, \tau\}}{4\min\{L, \mu + \tau\}}$ [Tseng, 1995].

### 4.1 CONVERGENCE ANALYSIS

In this section, we analyze the complexity of each of the three components we discussed. Let $T$ denote the outer-loop complexity, $K$ the inner-loop complexity, and $N$ the number of iterations for $\mathcal{M}$ (expected number if $\mathcal{M}$ is randomized) to solve subproblem $(\star\star)$. The total complexity of Algorithm 1 is computed by multiplying $K$, $T$, and $N$. Later, we will provide a guideline for choosing parameter $\tau$ to achieve the best complexity, given an algorithm $\mathcal{M}$.

**Theorem 4.1 (Outer loop)** Suppose function $f$ is NC-SC with strong convexity parameter $\mu$ and $L$-Lipschitz smooth. If we choose $\alpha_t = \frac{\mu^2}{526\max\{1, L\}^2}$ for $t > 0$ and $\alpha_0 = \frac{\mu^3}{576\max\{1, L\}^2}$, the output $\hat{x}_T$ from Algorithm 1 satisfies

$$
\mathbb{E}[\|\nabla \Phi(\hat{x}_T)\|^2] \leq \frac{268L}{5T} \Delta + \frac{28L}{5T} D_{y}^0,
$$

where $\Delta = \Phi(x_0) - \inf_x \Phi(x)$, $D_{y}^0 = \|y_0 - y^*(x_0)\|^2$ and $y^*(x_0) = \arg \max_{y \in \mathbb{R}^d} f(x_0, y)$.

This theorem implies that the algorithm finds an $\epsilon$ stationary point of $\Phi$ after inexact solving $(\star\star)$ for $T = O \left( \frac{L(\Delta + D_{y}^0)}{\epsilon^2} \right)$ times. The dependency on $D_{y}^0$ can be eliminated if we select the initialization $y_0$ close enough to $y^*(x_0)$, which only requires an additional logarithmic cost by maximizing a strongly concave function.

**Theorem 4.2 (Inner loop)** Under the same assumptions in Theorem 4.1 if we choose $\epsilon_k^t = \frac{x_0 - x_k^{i} \|y_k - y^*\|^2}{\Delta} \leq \frac{1}{\mu^2(\sqrt{q} - \rho)} \|\nabla \hat{f}_t(x_k, y_k)\|^2$.

Particularly, setting $\rho = 0.9\sqrt{q}$, Theorem 4.2 implies after inexact solving $(\star\star)$ for $K = \tilde{O} \left( \left( \frac{\tau + \mu}{\mu} \right) \log \frac{1}{\alpha_0} \right)$ times, the stopping criterion $21$ is satisfied. This complexity decreases with $\tau$. However, we should not choose $\tau$ too small, because the smaller $\tau$ is, the harder it is for $\mathcal{M}$ to solve $(\star\star)$. The following theorem captures the complexity for Algorithm $\mathcal{M}$ to solve the subproblem.

**Theorem 4.3 (Complexity of solving subproblems $(\star\star)$)**

Under the same assumptions in Theorem 4.1 and the choice
of \( c_k \) in Theorem 4.2, the number of iterations (expected number of iterations if \( M \) is stochastic) for \( M \) to solve \( \Phi \) such that \( \| \nabla \tilde{f}_{t,k}(x,y) \|^2 \leq c_k \) is

\[
N = O \left( \frac{L}{\mu, \tau} \log \left( \frac{\max\{1, L, \tau\}}{\min\{1, \mu\}} \right) \right).
\]

The above result implies that the subproblems can be solved within constant iterations that only depend on \( L, \mu, \tau \) and \( \Lambda_{M,L}^\tau \). This largely benefits from the use of warm-starting and stopping criterion with time-varying accuracy. In contrast, other inexact proximal point algorithms in minimax optimization, such as [Yang et al., 2020b, Lin et al., 2020b], fix the target accuracy, thus their complexity of solving the subproblems usually has an extra logarithmic factor in \( 1/\epsilon \).

The overall complexity of the algorithm follows immediately after combining the above three theorems:

**Corollary 4.1** Under the same assumptions in Theorem 4.1 and setting in Theorem 4.2, the total number (expected number if \( M \) is randomized) of gradient evaluations for Algorithm 7 to find an \( \epsilon \)-stationary point of \( \Phi \), is

\[
\tilde{O} \left( \frac{\lambda_{\mu,L}^\tau}{\epsilon^2} (\Delta + D_y^\mu) \sqrt{\mu + \tau} \right). \tag{24}
\]

In order to minimize the total complexity, we should choose the regularization parameter \( \tau \) that minimizes \( \lambda_{\mu,L}^\tau \sqrt{\mu + \tau} \).

### 4.2 SPECIFIC ALGORITHMS AND COMPLEXITIES

In this subsection, we discuss specific choices for \( M \) and the corresponding optimal choices of \( \tau \), as well as the resulting total complexities for solving NC-SC problems.

**Catalyst-EG/OGDA algorithm.** When solving NC-SC minimax problems in the general setting, we set \( M \) to be either extra-gradient method (EG) or optimistic gradient descent ascent (OGDA). Hence, we have \( \Lambda_{\mu,L}^\tau = \frac{L + \max\{2L, \tau\}}{4\min\{L, \mu + \tau\}} \) [Gidel et al., 2018, Azizian et al., 2020]. Minimizing \( \lambda_{\mu,L}^\tau \sqrt{\mu + \tau} \) yields that the optimal choice for \( \tau \) is \( L - \mu \). This leads to a total complexity of

\[
\tilde{O} \left( \sqrt{L} (\Delta + D_y^\mu) \epsilon^{-2} \right). \tag{25}
\]

**Remark 4.1** The above complexity matches the lower bound in Theorem 3.2 up to a logarithmic factor in \( L \) and \( \kappa \). It improves over Minimix-PPA [Lin et al., 2020b] by \( \log^2(1/\epsilon) \), GDA [Lin et al., 2020a] by \( \kappa^2 \) and therefore achieves the best of two worlds in terms of dependency on \( \kappa \) and \( \epsilon \). In addition, our Catalyst-EG/OGDA algorithm does not require the bounded domain assumption on \( y \), unlike [Lin et al., 2020b].

**Catalyst-SVRG/SAGA algorithm.** When solving NC-SC minimax problems in the averaged smooth finite-sum setting, we set \( M \) to be either SVRG or SAGA. Hence, we have \( \Lambda_{\mu,L}^\tau (\tau) \propto n + \frac{1}{\min\{L, \mu + \tau\}} \) [Balamurugan and Bach, 2016]. Minimizing \( \Lambda_{\mu,L}^\tau (\tau) \sqrt{\mu + \tau} \), the best choice for \( \tau \) is (proportional to) \( \max\{L, \frac{\mu}{\sqrt{\epsilon}} - \mu, 0\} \), which leads to the total complexity of

\[
\tilde{O} \left( \left( n + \frac{n^2}{\epsilon} \right) \sqrt{\Delta} (L + D_y^\mu) \epsilon^{-2} \right). \tag{26}
\]

**Remark 4.2** According to the lower bound established in Theorem 3.2, the dependency on \( \kappa \) in the above upper bound is nearly tight, up to logarithmic factors. Recall that SREDA [Luo et al., 2020] achieves the complexity of \( O \left( \lambda^2 \epsilon^{-2} + n + (n + \kappa) \log(\kappa) \right) \) for \( n \leq \kappa^2 \) and \( O \left( (\kappa^2 + \kappa n) \epsilon^{-2} \right) \) for \( n \leq \kappa^2 \). Hence, our SAGA-SVRG/OGDA algorithm attains better complexity in the regime \( n \leq \kappa^2 \). Particularly, in the critical regime \( \kappa = \Omega(\sqrt{n}) \) arising in statistical learning [Shalev-Shwartz and Ben-David, 2014], our algorithm performs strictly better.

### 5 CONCLUSION

In this work, we take an initial step towards understanding the fundamental limits of minimax optimization in the nonconvex-strongly-concave setting for both general and finite-sum cases, and bridge the gaps between lower and upper bounds. It remains interesting to investigate whether the dependence on \( n \) can be further tightened in the complexity for finite-sum NC-SC minimax optimization.

**Acknowledgements**

This work is supported by NSF CMMI #1761699, NSF CCF #1934986. CG’s research is partially supported by INRIA through the INRIA Associate Teams, and FONDECYT 1210362 project. NK’s research is supported by NSF CCF #1704970.

### References


---

2 Although Balamurugan and Bach [2016] assumes individual smoothness, its analysis can be extended to average smoothness.

3 SVRG in [Balamurugan and Bach, 2016] requires computing the proximal operator of an \((\mu, \mu)\)-SC function. For any \((\mu, \mu)\)-SC function in the form of \(\sum_i f_i(x, y)\), we can rewrite it as \(\sum_i \left( f_i(x, y) - \frac{1}{2} \| x \|^2 + \frac{\mu}{2} \| y \|^2 \right) + \frac{\mu}{2} \| (\| x \|^2 - \| y \|^2) \|^2\), where the first term is convex-concave, and the second term is \((\mu, \mu)\)-SC and admits a simple proximal operator.


Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. Mathematical Programming, pages 1–50, 2019a.


