SECOND-ORDER MIN-MAX OPTIMIZATION WITH LAZY HESSIANS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies second-order methods for convex-concave minimax optimization. Monteiro & Svaiter (2012) proposed a method to solve the problem with an optimal iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ to find an ϵ-saddle point. However, it is unclear whether the computational complexity, $\mathcal{O}((N + d^2)d\epsilon^{-2/3})$, can be improved. In the above, we follow Doikov et al. (2023) and assume the complexity of obtaining a first-order oracle as N and the complexity of obtaining a second-order oracle as dN. In this paper, we show that the computation cost can be reduced by reusing Hessian across iterations. Our methods take the overall computational complexity of $\tilde{\mathcal{O}}((N + d^2)(d + d^{2/3}\epsilon^{-2/3}))$, which improves those of previous methods by a factor of $d^{1/3}$. Furthermore, we generalize our method to strongly-convex-strongly-concave minimax problems and establish the complexity of $\tilde{\mathcal{O}}((N+d^2)(d+d^{2/3}\kappa^{2/3}))$ when the condition number of the problem is κ, enjoying a similar speedup upon the state-of-the-art method. Numerical experiments on both real and synthetic datasets also verify the efficiency of our method.

1 INTRODUCTION

We consider the following minimax optimization problem:

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

where we suppose f is (strongly-)convex in $\boldsymbol{x}$ and (strongly-)concave in $\boldsymbol{y}$. This setting covers many useful applications, including functionally constrained optimization (Xu, 2020), game theory (Von Neumann & Morgenstern, 1947), robust optimization (Ben-Tal et al., 2009), fairness-aware machine learning (Zhang et al., 2018), reinforcement learning (Du et al., 2017; Wang, 2017; Paternain et al., 2022; Wai et al., 2018), decentralized optimization (Kovalev et al., 2021; 2020), AUC maximization (Ying et al., 2016; Hanley & McNeil, 1982; Yuan et al., 2021).

First-order methods are widely studied for this problem. Classical algorithms include ExtraGradient (EG) (Korpelevich, 1976; Nemirovski, 2004), Optimistic Gradient Descent Ascent (OGDA) (Popov, 1980; Mokhtari et al., 2020a;b), Hybrid Proximal Extragradient (HPE) (Monteiro & Svaiter, 2010), and Dual Extrapolation (DE) (Nesterov & Scrimali, 2006; Nesterov, 2007). When the gradient of $f(\cdot,\cdot)$ is L-Lipschitz continuous, these methods attain the rate of $\mathcal{O}(\epsilon^{-1})$ under convex-concave (C-C) setting and the rate of $\mathcal{O}((L/\mu)\log(\epsilon^{-1}))$ when $f(\cdot,\cdot)$ is μ-strongly convex in $\boldsymbol{x}$ and μ-strongly-concave in $\boldsymbol{y}$ (SC-SC) for $\mu > 0$. They are all optimal under C-C and SC-SC setting due to the lower bounds reported by (Nemirovskij & Yudin, 1983; Zhang et al., 2022a).

Second-order methods usually lead to faster rates than first-order methods when the Hessian of $f(\cdot,\cdot)$ is ρ-Lipschitz continuous. A line of works (Nesterov & Scrimali, 2006; Huang et al., 2022) extended the celebrated Cubic Regularized Newton (CRN) (Nesterov & Polyak, 2006) method to minimax problems with local superlinear convergence rates and global convergence guarantee. However, the established global convergence rates of $\mathcal{O}(\epsilon^{-1})$ by Nesterov & Scrimali (2006) and $\mathcal{O}((L\rho/\mu^2)\log(\epsilon^{-1}))$ by Huang et al. (2022) under C-C and SC-SC conditions are no better than the optimal first-order methods. Another line of work generalizes the optimal first-order methods to higher-order methods. Monteiro & Svaiter (2012) proposed the Newton Proximal Extragradient (NPE) method with a global convergence rate of $\mathcal{O}(\epsilon^{-2/3}\log\log(\epsilon^{-1}))$ under the C-C conditions.

This result nearly matches the lower bounds (Adil et al., 2022; Lin & Jordan, 2024), except an additional $\mathcal{O}(\log\log(\epsilon^{-1}))$ factor which is caused by the implicit binary search at each iteration. Recently, Alves & Svaiter (2023) proposed a search-free NPE method to remove this $\mathcal{O}(\log\log(\epsilon^{-1}))$ factor based on ideas from homotopy. Over the past decade, researchers also proposed various second-order methods, in addition to the NPE framework (Monteiro & Svaiter, 2012), that achieve the same convergence rate. This algorithms include the second-order extension of EG (Bullins & Lai, 2022; Adil et al., 2022; Lin et al., 2022; Huang & Zhang, 2022) (which we refer to as EG-2, Huang & Zhang (2022) call it ARE), that of OGDA (Jiang & Mokhtari, 2022; Jiang et al., 2024) (which we refer to as OGDA-2), and that of DE (Lin & Jordan, 2024) (they name the method Perseus). For SC-SC problems, Jiang & Mokhtari (2022) proved the OGDA-2 can converge at the rate of $\mathcal{O}((\rho/\mu)^{2/3} + \log\log(\epsilon^{-1}))$, and Huang & Zhang (2022) proposed the ARE-restart with the rate of $\mathcal{O}((\rho/\mu)^{2/3}\log\log(\epsilon^{-1}))$.

Although the aforementioned second-order methods Adil et al. (2022); Lin & Jordan (2024); Lin et al. (2022); Jiang & Mokhtari (2022); Monteiro & Svaiter (2012) enjoy an improved convergence rate over the first-order methods and have achieved optimal iteration complexities, they require querying one new Hessian at each iteration and solving a matrix inversion problem at each Newton step, which leads to a $\mathcal{O}(d^3)$ computational cost per iteration. This becomes the main bottleneck that limits the applicability of second-order methods. Liu & Luo (2022a) proposed quasi-Newton methods for saddle point problems that access one Hessian-vector product instead of the exact Hessian for each iteration. The iteration complexity is $\mathcal{O}(d^2)$ for quasi-Newton methods. However, their methods do not have a global convergence guarantee under general (S)C)-(S)C conditions.

In this paper, we propose a computation-efficient second-order method, which we call LEN (Lazy Extra Newton method). In contrast to all existing second-order methods or quasi-Newton methods for minimax optimization problems that always access new second-order information for the coming iteration, LEN reuses the second-order information from past iterations. Specifically, LEN solves a cubic regularized sub-problem using the Hessian from the snapshot point that is updated every m iteration, then conducts an extra-gradient step by the gradient from the current iteration. We provide a rigorous theoretical analysis of LEN to show it maintains fast global convergence rates and goes beyond the optimal second-order methods Adil et al. (2022); Lin & Jordan (2024); Huang & Zhang (2022); Lin et al. (2022); Alves & Svaiter (2023); Jiang et al. (2024) in terms of the overall computational complexity. We summarize our contributions as follows.

- When the object function $f(\cdot,\cdot)$ is convex in $\boldsymbol{x}$ and concave in $\boldsymbol{y}$, we propose LEN and prove that it finds an ϵ-saddle point in $\mathcal{O}(m^{2/3}\epsilon^{-2/3})$ iterations. Under Assumption 3.4, where the complexity of calculating $\boldsymbol{F}(\boldsymbol{z})$ is N and the complexity of calculating $\nabla\boldsymbol{F}(\boldsymbol{z})$ is dN, the optimal choice is $m = \Theta(d)$. In this case, LEN only requires a computational complexity of $\tilde{\mathcal{O}}((N+d^2)(d+d^{2/3}\epsilon^{-2/3}))$, which is strictly better than $\mathcal{O}((N+d^2)d\epsilon^{-2/3})$ for the existing optimal second-order methods by a factor of $d^{1/3}$.

- When the object function $f(\cdot,\cdot)$ is μ-strongly-convex in $\boldsymbol{x}$ and μ-strongly-concave in $\boldsymbol{y}$, we apply the restart strategy on LEN and propose LEN-restart. We prove the algorithm can find an ϵ-root with $\tilde{\mathcal{O}}((N+d^2)(d+d^{2/3}(\rho/\mu)^{2/3}))$ computational complexity, where ρ means the Hessian of $f(\cdot,\cdot)$ is ρ Lipschitz-continuous. Our result is strictly better than the $\tilde{\mathcal{O}}((N+d^2)d(\rho/\mu)^{2/3})$ in prior works.

We compare our results with the prior works in Table 1.

2 Related Works and Technical Challenges

Lazy Hessian in minimization problems. The idea of reusing Hessian was initially presented by Shamanskii (1967) and later incorporated into the Levenberg-Marquardt method, damped Newton method, and proximal Newton method (Fan, 2013; Lampariello & Sciandrone, 2001; Wang et al., 2006; Adler et al., 2020). However, the explicit advantage of lazy Hessian update over the ordinary Newton(-type) update was not discovered until the recent work of (Doikov et al., 2023; Chayti et al., 2023). Let $M > 0$ be a constant and $\pi(t) = t - t \mod m$. They applied the following lazy Hessian

Table 1: We compare the required computational complexity to achieve an ϵ-saddle point of the proposed LEN with the optimal choice $m = \Theta(d)$ and other existing algorithms on both convex-concave (C-C) and strongly-convex-strongly-concave (SC-SC) problems. Here, $d = d_x + d_y$ is the dimension of the problem. We assume the gradient is L-Lipschitz continuous for EG and the Hessian is ρ-Lipschitz continuous for others. We count each gradient oracle call with N computational complexity, and each Hessian oracle with dN computational complexity.

Setup	Method	Computational Cost
C-C	EG (Korpelevich, 1976)	$\mathcal{O}((N+d)\epsilon^{-1})$
	NPE (Monteiro & Svaiter, 2012)	$\tilde{\mathcal{O}}((N+d^2)d\epsilon^{-2/3})$
	search-free NPE (Alves & Svaiter, 2023)	$\mathcal{O}((N+d^2)d\epsilon^{-2/3})$
	EG-2 (Adil et al., 2022)	$\tilde{\mathcal{O}}((N+d^2)d\epsilon^{-2/3})$
	Perseus (Lin & Jordan, 2024)	$\tilde{\mathcal{O}}((N+d^2)d\epsilon^{-2/3})$
	OGDA-2 (Jiang & Mokhtari, 2022)	$\mathcal{O}((N+d^2)d\epsilon^{-2/3})$
	LEN (Theorem 4.3)	$\tilde{\mathcal{O}}((N+d^2)(d+d^{2/3}\epsilon^{-2/3}))$
SC-SC	EG (Korpelevich, 1976)	$\tilde{\mathcal{O}}((N+d)(L/\mu))$
	OGDA-2 (Jiang & Mokhtari, 2022)	$\mathcal{O}((N+d^2)d(\rho/\mu)^{2/3})$
	ARE-restart (Huang & Zhang, 2022)	$\tilde{\mathcal{O}}((N+d^2)d(\rho/\mu))^{2/3})$
	Perseus-restart (Lin & Jordan, 2024)	$\tilde{\mathcal{O}}((N+d^2)d(\rho/\mu)^{2/3})$
	LEN-restart (Corollary 4.1)	$\tilde{\mathcal{O}}((N+d^2)(d+d^{2/3}(\rho/\mu)^{2/3}))$

update on cubic regularized Newton (CRN) methods (Nesterov & Polyak, 2006):

$$\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z}\in\mathbb{R}^d} \left\{ \langle \boldsymbol{F}(\boldsymbol{z}_t), \boldsymbol{z}-\boldsymbol{z}_t\rangle + \frac{1}{2}\langle \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}-\boldsymbol{z}_t), \boldsymbol{z}-\boldsymbol{z}_t\rangle + \frac{M}{6}\|\boldsymbol{z}-\boldsymbol{z}_t\|^3 \right\}, \tag{2}$$

where $\boldsymbol{F}: \mathbb{R}^d \to \mathbb{R}^d$ is the gradient field of a convex function. They establish the convergence rates of $\mathcal{O}(\sqrt{m}\epsilon^{-3/2})$ for nonconvex optimization (Doikov et al., 2023), and $\mathcal{O}(\sqrt{m}\epsilon^{-1/2})$ for convex optimization (Chayti et al., 2023) respectively. Such rates lead to the total computational cost of $\tilde{\mathcal{O}}((N+d^2)(d+\sqrt{d}\epsilon^{-3/2}))$ and $\tilde{\mathcal{O}}((N+d^2)(d+\sqrt{d}\epsilon^{-1/2}))$ by setting $m=\Theta(d)$, which strictly improve the result by classical CRN methods by a factor of $\sqrt{d}$ on both setups.

We have also observed that the idea of the "lazy Hessian" is widely used in practical second-order methods. Sophia (Liu et al., 2023) estimates a diagonal Hessian matrix as a pre-conditioner, and to reduce the complexity, the pre-conditioner is updated in a lazy manner. KFAC (Martens & Grosse, 2015; Grosse & Martens, 2016) approximates the Fisher information matrix, and it also uses an exponential moving average (EMA) to update the estimate of the Fisher information matrix, which can be viewed as a soft version of the lazy update.

Challenge of using lazy Hessian updates in minimax problems. In comparison to the previous works on lazy Hessian, our methods LEN and LEN-restart demonstrate the advantage of using lazy Hessian for a broader class of optimization problems, the *minimax* problems. Our analysis differs from the ones in Doikov et al. (2023); Chayti et al. (2023). Their methods only take a lazy CRN update (2) at each iteration, which makes it easy to bound the error of lazy Hessian updates using Assumption 3.1 and the triangle inequality in the following way:

$$\|\nabla \boldsymbol{F}(\boldsymbol{z}_t) - \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})\| \le \rho\|\boldsymbol{z}_{\pi(t)} - \boldsymbol{z}_t\| \le \rho \sum_{i=\pi(t)}^{t-1} \|\boldsymbol{z}_i - \boldsymbol{z}_{i+1}\|.$$

Our method, on the other hand, does not only take a lazy (regularized) Newton update but also requires an extra gradient step (Line 3 in Algorithm 1). Thus, the progress of one Newton update

$\{\|z_{i+1/2} - z_i\|\}_{i=\pi(t)}^{t}$ cannot directly bound the error term $\|z_t - z_{\pi(t)}\|$ introduced by lazy Hessian update. Moreover, for minimax problems the matrix $\nabla F(z_{\pi(t)})$ is no longer symmetric, which leads to different analysis and implementation of sub-problem solving (Section 4.3). We refer the readers to Section 4.1 for more detailed discussions.

Notations. Throughout this paper, $\log$ is base 2 and $\log_+(\,\cdot\,) := 1 + \log(\,\cdot\,)$. We use $\|\cdot\|$ to denote the spectral norm and Euclidean norm of matrices and vectors, respectively. We denote $\pi(t) = t - (t \mod m)$ when presenting the lazy updates.

3 Preliminaries

In this section, we introduce notations and basic assumptions used in our work. We start with several standard definitions for Problem (1).

Definition 3.1. *We call a function $f(\boldsymbol{x}, \boldsymbol{y}) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ has ρ-Lipschitz Hessians if we have*

$$\|\nabla^2 f(\boldsymbol{x}, \boldsymbol{y}) - \nabla^2 f(\boldsymbol{x}', \boldsymbol{y}')\| \le \rho \left\| \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}' \\ \boldsymbol{y} - \boldsymbol{y}' \end{bmatrix} \right\|, \quad \forall (\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{x}', \boldsymbol{y}') \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}.$$

Definition 3.2. *A differentiable function $f(\cdot, \cdot)$ is μ-strongly-convex-μ-strongly-concave for some $\mu > 0$ if*

$$f(\boldsymbol{x}', \boldsymbol{y}) \ge f(\boldsymbol{x}, \boldsymbol{y}) + (\boldsymbol{x}' - \boldsymbol{x})^\top \nabla_x f(\boldsymbol{x}, \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2, \quad \forall \boldsymbol{x}', \boldsymbol{x} \in \mathbb{R}^{d_x}, \boldsymbol{y} \in \mathbb{R}^{d_y};$$
$$f(\boldsymbol{x}, \boldsymbol{y}') \le f(\boldsymbol{x}, \boldsymbol{y}) + (\boldsymbol{y}' - \boldsymbol{y})^\top \nabla_y f(\boldsymbol{x}, \boldsymbol{y}) - \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{y}'\|^2, \quad \forall \boldsymbol{y}', \boldsymbol{y} \in \mathbb{R}^{d_y}, \boldsymbol{x} \in \mathbb{R}^{d_x}.$$

We say f is convex-concave if $\mu = 0$.

We are interested in finding a saddle point of Problem (1), formally defined as follows.

Definition 3.3. *We call a point $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ a saddle point of a function $f(\cdot, \cdot)$ if we have*

$$f(\boldsymbol{x}^*, \boldsymbol{y}) \le f(\boldsymbol{x}^*, \boldsymbol{y}^*) \le f(\boldsymbol{x}, \boldsymbol{y}^*), \quad \forall \boldsymbol{x} \in \mathbb{R}^{d_x},\ \boldsymbol{y} \in \mathbb{R}^{d_y}.$$

Next, we introduce all the assumptions made in this work. In this paper, we focus on Problem (1) that satisfies the following assumptions.

Assumption 3.1. *We assume the function $f(\cdot, \cdot)$ is twice continuously differentiable, has ρ-Lipschitz continuous Hessians, and has at least one saddle point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$.*

We will study convex-concave problems and strongly-convex-strongly-concave problems.

Assumption 3.2 (C-C setting)**.** *We assume the function $f(\cdot, \cdot)$ is convex in $\boldsymbol{x}$ and concave in $\boldsymbol{y}$.*

Assumption 3.3 (SC-SC setting)**.** *We assume the function $f(\cdot, \cdot)$ is μ-strongly-convex-μ-strongly-concave. We denote the condition number as $\kappa := \rho/\mu$*

We let $d := d_x + d_y$ and denote the aggregated variable $\boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^d$. We also denote the GDA field of f and its Jacobian as

$$\boldsymbol{F}(\boldsymbol{z}) := \begin{bmatrix} \nabla_x f(\boldsymbol{x}, \boldsymbol{y}) \\ -\nabla_y f(\boldsymbol{x}, \boldsymbol{y}) \end{bmatrix}, \quad \nabla \boldsymbol{F}(\boldsymbol{z}) := \begin{bmatrix} \nabla^2_{xx} f(\boldsymbol{x}, \boldsymbol{y}) & \nabla^2_{xy} f(\boldsymbol{x}, \boldsymbol{y}) \\ -\nabla^2_{yx} f(\boldsymbol{x}, \boldsymbol{y}) & -\nabla^2_{yy} f(\boldsymbol{x}, \boldsymbol{y}) \end{bmatrix}. \tag{3}$$

The GDA field of $f(\cdot, \cdot)$ has the following properties.

Lemma 3.1 (Lemma 2.7 Lin et al. (2022))**.** *Under Assumptions 3.1 and 3.2, we have*

1. *$\boldsymbol{F}$ is monotone, i.e. $\langle \boldsymbol{F}(\boldsymbol{z}) - \boldsymbol{F}(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}' \rangle \ge 0,\ \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^d$.*
2. *$\nabla \boldsymbol{F}$ is ρ-Lipschitz continuous, i.e. $\|\nabla \boldsymbol{F}(\boldsymbol{z}) - \nabla \boldsymbol{F}(\boldsymbol{z}')\| \le \rho\|\boldsymbol{z} - \boldsymbol{z}'\|,\ \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^d$.*
3. *$\boldsymbol{F}(\boldsymbol{z}^*) = 0$ if and only if $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a saddle point of function $f(\cdot, \cdot)$.*

Furthermore, if Assumption 3.3 holds, we have $\boldsymbol{F}(\cdot)$ is μ-strongly-monotone, i.e.

$$\langle \boldsymbol{F}(\boldsymbol{z}) - \boldsymbol{F}(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}' \rangle \ge \mu\|\boldsymbol{z} - \boldsymbol{z}'\|^2,\ \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^d.$$

For the C-C case, the commonly used optimality criterion is the following restricted gap.

Definition 3.4 (Nesterov (2007)). *Let $\mathbb{B}_\beta(\boldsymbol{w})$ be the ball centered at $\boldsymbol{w}$ with radius β. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point of function f. For a given point $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, we let β sufficiently large such that it holds*

$$\max\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|,\ \|\hat{\boldsymbol{y}} - \boldsymbol{y}^*\|\} \le \beta,$$

we define the restricted gap function as

$$\mathrm{Gap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \beta) := \max_{\boldsymbol{y} \in \mathbb{B}_\beta(\boldsymbol{y}^*)} f(\hat{\boldsymbol{x}}, \boldsymbol{y}) - \min_{\boldsymbol{x} \in \mathbb{B}_\beta(\boldsymbol{x}^*)} f(\boldsymbol{x}, \hat{\boldsymbol{y}}),$$

We call $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ an ϵ-saddle point if $\mathrm{Gap}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \beta) \le \epsilon$ and $\beta = \Omega(\max\{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|, \|\boldsymbol{y}_0 - \boldsymbol{y}^*\|\})$.

For the SC-SC case, we use the following stronger notion.

Definition 3.5. *Suppose that Assumption 3.3 holds. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the unique saddle point of function f. We call $\hat{\boldsymbol{z}} = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ an ϵ-root if $\|\hat{\boldsymbol{z}} - \boldsymbol{z}^*\| \le \epsilon$.*

Most previous works only consider the complexity metric as the number of oracle calls, where an oracle takes a point $\boldsymbol{z} \in \mathbb{R}^d$ as the input and returns a tuple $(\boldsymbol{F}(\boldsymbol{z}), \nabla \boldsymbol{F}(\boldsymbol{z}))$ as the output. The existing algorithms (Monteiro & Svaiter, 2012; Lin & Jordan, 2022; Adil et al., 2022; Alves & Svaiter, 2023) have achieved optimal complexity regarding the number of oracle calls. In this work, we focus on the computational complexity of the oracle. More specifically, we distinguish between the different computational complexities of calculating the Hessian matrix $\nabla \boldsymbol{F}(\boldsymbol{z})$ and the gradient $\boldsymbol{F}(\boldsymbol{z})$. Formally, we make the following assumption as Doikov et al. (2023).

Assumption 3.4. *We count the computational complexity of computing $\boldsymbol{F}(\cdot)$ as N and the computational complexity of $\nabla \boldsymbol{F}(\cdot)$ as Nd.*

Remark 3.1. *Assumption 3.4 supposes the cost of computing $\nabla \boldsymbol{F}(\cdot)$ is d times that of computing $\boldsymbol{F}(\cdot)$. It holds in many practical scenarios as one Hessian oracle can be computed via d Hessian-vector products on standard basis vectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_d$, and one Hessian-vector product oracle is typically as expensive as one gradient oracle (Wright, 2006):*

1. *When the computational graph of f is obtainable, both $\boldsymbol{F}(\boldsymbol{z})$ and $\nabla \boldsymbol{F}(\boldsymbol{z})\boldsymbol{v}$ can be computed using automatic differentiation with the same cost for any $\boldsymbol{z}, \boldsymbol{v} \in \mathbb{R}^d$.*
2. *When f is a black box function, we can estimate the Hessian-vector $\nabla \boldsymbol{F}(\boldsymbol{z})\boldsymbol{v}$ via the finite-difference $\boldsymbol{u}_\delta(\boldsymbol{z}; \boldsymbol{v}) = \frac{1}{\delta}(\boldsymbol{F}(\boldsymbol{z} + \delta \boldsymbol{v}) - \boldsymbol{F}(\boldsymbol{z} - \delta \boldsymbol{v}))$ and we have $\lim_{\delta \to 0} \boldsymbol{u}_\delta(\boldsymbol{z}; \boldsymbol{v}) = \nabla \boldsymbol{F}(\boldsymbol{z})\boldsymbol{v}$ under mild conditions on $\boldsymbol{F}(\cdot)$.*

4 Algorithms and Convergence Analysis

In this section, we present novel second-order methods for solving minimax optimization problems (1). We present LEN and its convergence analysis for convex-concave minimax problems in Section 4.1. We generalize LEN for strongly-convex-strongly-concave minimax problems by presenting LEN-restart in Section 4.2. We discuss the details of solving minimax cubic-regularized sub-problem, present detailed implementation of LEN, and give the total computational complexity of proposed methods in Section 4.3.

4.1 The LEN algorithm for convex-concave problems

We propose LEN for convex-concave problems in Algorithm 1. Our method builds on the recently proposed optimal second-order methods ARE (Adil et al., 2022; Huang & Zhang, 2022) / Newton-Minimax (Lin et al., 2022). The only change is that we reuse the Hessian from previous iterates, as colored in blue. Each iteration of LEN contains the following two steps:

$$\begin{cases} \boldsymbol{F}(\boldsymbol{z}_t) + \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t) + M\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t\|(\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t) = \boldsymbol{0}, & \text{(Implicit Step)} \\ \boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \dfrac{\boldsymbol{F}(\boldsymbol{z}_{t+1/2})}{M\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t\|}. & \text{(Explicit Step)} \end{cases} \tag{4}$$

Algorithm 1 LEN$(\boldsymbol{z}_0, T, m, M)$

1: **for** $t = 0, \cdots, T-1$ **do**
2: Compute lazy cubic step, *i.e.* find $\boldsymbol{z}_{t+1/2}$ that satisfies
$$\boldsymbol{F}(\boldsymbol{z}_t) = (\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + M\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|\boldsymbol{I}_d)(\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}).$$
3: Compute $\gamma_t = M\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|$.
4: Compute extra-gradient step $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \gamma_t^{-1}\boldsymbol{F}(\boldsymbol{z}_{t+1/2})$.
5: **end for**
6: **return** $\bar{z}_T = \frac{1}{\sum_{t=0}^{T-1}\gamma_t^{-1}}\sum_{t=0}^{T-1}\gamma_t^{-1}\boldsymbol{z}_{t+1/2}$.

The first step (implicit step) solves a cubic regularized sub-problem based on the $\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})$ computed at the latest snapshot point and $\boldsymbol{F}(\boldsymbol{z}_t)$ at the current iteration point. This step is often viewed as an oracle (Lin & Jordan, 2024; Nesterov, 2023; Lin et al., 2022; Adil et al., 2022) as there exists efficient solvers, which will also be discussed in Section 4.3. The second one (explicit step) conducts an extra gradient step based on $\boldsymbol{F}(\boldsymbol{z}_{t+1/2})$.

Reusing the Hessian in the implicit step makes each iteration much cheaper, but would cause additional errors compared to previous methods (Huang & Zhang, 2022; Lin et al., 2022; Adil et al., 2022; Nesterov, 2023). The error resulting from the lazy Hessian updates is formally characterized by the following theorem.

Lemma 4.1. *Suppose that Assumption 3.1 and 3.2 hold. For any $\boldsymbol{z} \in \mathbb{R}^d$, Algorithm 1 ensures*
$$\begin{aligned}\gamma_t^{-1}\langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z}\rangle \leq \frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_{t+1} - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\|^2 \\ - \frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2 + \frac{\rho^2}{2M^2}\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2 + \underbrace{\frac{2\rho^2}{M^2}\|\boldsymbol{z}_{\pi(t)} - \boldsymbol{z}_t\|^2}_{(*)}.\end{aligned}$$

Above, (*) is the error from lazy Hessian updates. Note that (*) vanishes when the current Hessian is used. For lazy Hessian updates, the error would accumulate in the epoch.

The key step in our analysis shows that we can use the negative terms in the right-hand side of the inequality in Lemma 4.1 to bound the accumulated error by choosing M sufficiently large, with the help of the following technical lemma.

Lemma 4.2. *For any sequence of positive numbers $\{r_t\}_{t\geq 0}$, it holds for any $m \geq 2$ that $\sum_{t=1}^{m-1}\left(\sum_{i=0}^{t-1} r_i\right)^2 \leq \frac{m^2}{2}\sum_{t=0}^{m-1} r_t^2$.*

When $m = 1$, the algorithm reduces to the EG-2 algorithm (Huang & Zhang, 2022; Lin et al., 2022; Adil et al., 2022) without using lazy Hessian updates. When $m \geq 2$, we use Lemma 4.2 to upper bound the error that arises from lazy Hessian updates. Finally, we prove the following guarantee for our proposed algorithm.

Theorem 4.1 (C-C setting)**.** *Suppose that Assumption 3.1 and 3.2 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point and $\beta = \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$. Set $M \geq 3\rho m$. The sequence of iterates generated by Algorithm 1 is bounded $\boldsymbol{z}_t \in \mathbb{B}_\beta(\boldsymbol{z}^*)$, $\boldsymbol{z}_{t+1/2} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*)$, $\forall t = 0, \cdots, T-1$, and satisfies the following ergodic convergence:*
$$\mathrm{Gap}(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T; 3\beta) \leq \frac{16M\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^3}{T^{3/2}}.$$
Let $M = 3\rho m$. Algorithm 1 finds an ϵ-saddle point within $\mathcal{O}(m^{2/3}\epsilon^{-2/3})$ iterations.

Discussion on the computational complexity of the oracles. Theorem 4.1 indicates that LEN requires $\mathcal{O}(m^{2/3}\epsilon^{-2/3})$ calls to $\boldsymbol{F}(\cdot)$ and $\mathcal{O}(m^{2/3}\epsilon^{-2/3}/m + 1)$ calls to $\nabla\boldsymbol{F}(\cdot)$ to find the ϵ-saddle point. Under Assumption 3.4, the computational cost to call the oracles $\boldsymbol{F}(\cdot)$ and $\nabla\boldsymbol{F}(\cdot)$ is
$$\text{Oracle Computational Cost} = \mathcal{O}\left(N \cdot m^{2/3}\epsilon^{-2/3} + (Nd)\cdot\left(\frac{\epsilon^{-2/3}}{m^{1/3}} + 1\right)\right). \tag{5}$$

Algorithm 2 LEN-restart($\boldsymbol{z}_0, T, m, M, S$)

1: $\boldsymbol{z}^{(0)} = \boldsymbol{z}_0$
2: **for** $s = 0, \cdots, S-1$
3: $\quad \boldsymbol{z}^{(s+1)} = \text{LEN}(\boldsymbol{z}^{(s)}, T, m, M)$
end for

Taking $m = \Theta(d)$ minimizes (5) to $\mathcal{O}(Nd + Nd^{2/3}\epsilon^{-2/3})$. In comparison to the state-of-the-art second-order methods (Huang & Zhang, 2022; Lin et al., 2022; Adil et al., 2022), whose computational cost in terms of the oracles is $\mathcal{O}(Nd\epsilon^{-2/3})$ since they require to query $\nabla \boldsymbol{F}(\cdot)$ at each iteration, our methods significantly improve their results by a factor of $d^{1/3}$.

It is worth noticing that the computational cost of an algorithm includes both the computational cost of calling the oracles, which we have discussed above, and the computational cost of performing the updates (*i.e.* solving auxiliary problems) after accessing the required oracles. We will give an efficient implementation of LEN and analyze the total computational cost later in Section 4.3.

4.2 THE LEN-RESTART ALGORITHM FOR STRONGLY-CONVEX-STRONGLY-CONCAVE PROBLEMS

We generalize LEN to solve the strongly-convex-strongly-concave minimax optimization by incorporating the restart strategy introduced by Huang & Zhang (2022). We propose the LEN-restart in Algorithm 2, which works in epochs. Each epoch of LEN-restart invokes LEN (Algorithm 1), which gets $\boldsymbol{z}^{(s)}$ as inputs and outputs $\boldsymbol{z}^{(s+1)}$.

The following theorem shows that the sequence $\{\boldsymbol{z}^{(s)}\}$ enjoys a superlinear convergence in epochs. Furthermore, the required number of iterations in each epoch to achieve such a superlinear rate is only a constant.

Theorem 4.2 (SC-SC setting)**.** *Suppose that Assumptions 3.1 and 3.3 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the unique saddle point. Set $M = 3\rho m$ as Theorem 4.1 and $T = \left(\frac{2M\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|}{\mu}\right)^{2/3}$. Then the sequence of iterates generated by Algorithm 2 converge to $\boldsymbol{z}^*$ superlinearly as $\|\boldsymbol{z}^{(s)} - \boldsymbol{z}^*\|^2 \le \left(\frac{1}{2}\right)^{(3/2)^s} \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2$. In particular, Algorithm 2 with $M = 3\rho m$ finds a point $\boldsymbol{z}^{(s)}$ such that $\|\boldsymbol{z}^{(s)} - \boldsymbol{z}^*\| \le \epsilon$ within $S = \log_{3/2}\log_2(1/\epsilon)$ epochs. The total number of inner loop iterations is given by*

$$TS = \mathcal{O}\left(m^{2/3}\kappa^{2/3}\log\log(1/\epsilon)\right).$$

Under Assumption 3.4, Algorithm 2 with $m = \Theta(d)$ takes the computational complexity of $\mathcal{O}((Nd + Nd^{2/3}\kappa^{2/3})\log\log(1/\epsilon))$ to call the oracles $\boldsymbol{F}(\cdot)$ and $\nabla \boldsymbol{F}(\cdot)$.

4.3 IMPLEMENTATION DETAILS AND COMPUTATIONAL COMPLEXITY ANALYSIS

We provide details of implementing the cubic regularized Newton oracle (Implicit Step, (4)). Inspired by Monteiro & Svaiter (2012); Adil et al. (2022); Lin et al. (2022), we transform the subproblem into a root-finding problem for a univariate function.

Lemma 4.3 (Section 4.3 Lin et al. (2022))**.** *Suppose Assumption 3.1 and 3.2 hold for function $f : \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ and let $\boldsymbol{F}$ be its GDA field. Define $\gamma_t = M\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t\|$. The cubic regularized Newton oracle (Implicit Step, (4)) can be rewritten as:*

$$\boldsymbol{z}_{t+1/2} = \boldsymbol{z}_t - (\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + \gamma_t \boldsymbol{I}_d)^{-1}\boldsymbol{F}(\boldsymbol{z}_t),$$

which can be implemented by finding the root of the following univariate function:

$$\phi(\gamma) := M \left\|\left(\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + \gamma \boldsymbol{I}_d\right)^{-1}\boldsymbol{F}(\boldsymbol{z}_t)\right\| - \gamma. \tag{6}$$

Furthermore, the function $\phi(\gamma)$ is strictly decreasing when $\lambda > 0$.

From the above lemma, to implement the cubic regularized Newton oracle, it suffices to find the root of a strictly decreasing function $\phi(\gamma)$, which can be solved within $\tilde{\mathcal{O}}(1)$ iteration. The main operation is to solve the following linear system:

$$(\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + \gamma \boldsymbol{I}_d)\boldsymbol{h} = \boldsymbol{F}(\boldsymbol{z}_t). \tag{7}$$

Naively solving this linear system at every iteration still results in an expensive computational complexity of $\mathcal{O}(d^3)$ per iteration.

We present a computationally efficient way to implement LEN by leveraging the Schur factorization at the snapshot point $\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{Q}^{-1}$, where $\boldsymbol{Q} \in \mathbb{C}^d$ is a unitary matrix and $\boldsymbol{U} \in \mathbb{C}^d$ is an upper-triangular matrix. Then we apply the following update

$$\boldsymbol{h} = \boldsymbol{Q}(\boldsymbol{U} + \gamma \boldsymbol{I}_d)^{-1}\boldsymbol{Q}^{-1}\boldsymbol{F}(\boldsymbol{z}_t) \tag{8}$$

instead of solving the linear system (7) at each iteration. The final implementable algorithm is presented in Algorithm 3.

Now, we are ready to analyze the total computational complexity of LEN, which can be divided into the following two parts:

$$\text{Computational Cost} = \text{Oracle Computational Cost} + \text{Update Computational Cost},$$

where the first part has been discussed in Section 4.1. As for the update computational cost, the Schur decomposition with an $\mathcal{O}(d^3)$ computational complexity is required once every m iterations. After the Schur decomposition has been given at the snapshot point, the dominant part of the update computational complexity is solving the linear system (7), which can be done efficiently by solving the upper-triangular linear system (8) with the back substitution algorithm within $\mathcal{O}(d^2)$ computational complexity. Thus, we have

$$\text{Update Computational Cost} = \tilde{\mathcal{O}}\left(d^2 \cdot m^{2/3}\epsilon^{-2/3} + d^3 \cdot \left(\frac{\epsilon^{-2/3}}{m^{1/3}} + 1\right)\right), \tag{9}$$

and the total computational cost of LEN is

$$\text{Computational Cost} \overset{(5),(9)}{=} \tilde{\mathcal{O}}\left((d^2 + N) \cdot m^{2/3}\epsilon^{-2/3} + (d^3 + Nd) \cdot \left(\frac{\epsilon^{-2/3}}{m^{1/3}} + 1\right)\right). \tag{10}$$

By taking $m = \Theta(d)$, we obtain the best computational complexity in (10) of LEN, which is formally stated in the following theorem.

Theorem 4.3 (C-C setting)**.** *Under the same setting of Theorem 4.1, Algorithm 3 with $m = \Theta(d)$ finds an ϵ-saddle point with $\tilde{\mathcal{O}}((N + d^2)(d + d^{2/3}\epsilon^{-2/3})$ computational complexity.*

We also present the total computational complexity of LEN-restart for SC-SC setting.

Corollary 4.1 (SC-SC setting)**.** *Under the same setting of Theorem 4.2, Algorithm 2 implemented in the same way as Algorithm 3 with $m = \Theta(d)$ finds an ϵ-root with $\tilde{\mathcal{O}}((N + d^2)(d + d^{2/3}\kappa^{2/3})$ computational complexity.*

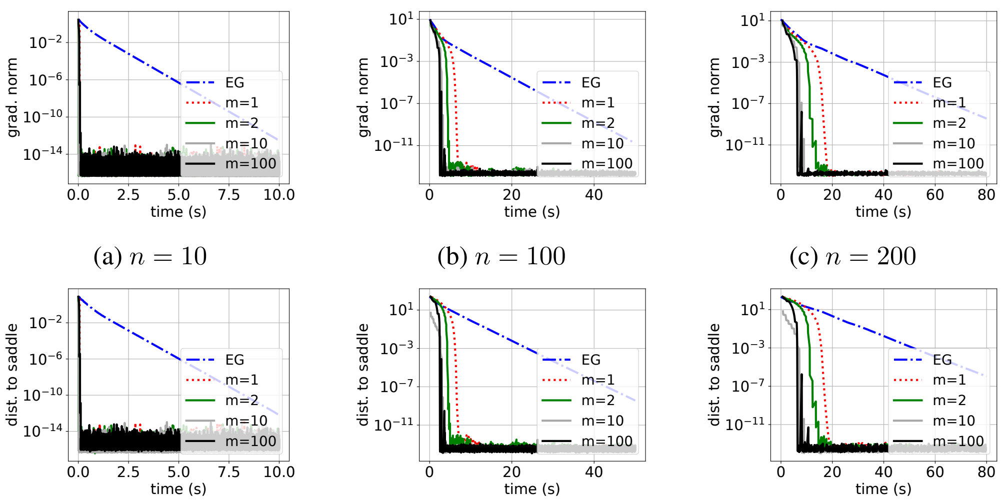
In both cases, our proposed algorithms improve the total computational cost of the optimal second-order methods (Monteiro & Svaiter, 2012; Lin & Jordan, 2024; Adil et al., 2022; Jiang & Mokhtari, 2022) by a factor of $d^{1/3}$.

Remark 4.1. *In the main text, we assume the use of the classical algorithm for matrix inversion/decomposition, which has a computational complexity of $\mathcal{O}(d^3)$. The fast matrix multiplication proposed by researchers in the field of theoretical computer science only requires a complexity of d^ω, where the best-known ω is currently around 2.371552 (Williams et al., 2024). This also implies faster standard linear algebra operators including Schur decomposition and matrix inversion (Demmel et al., 2007). However, the large hidden constant factors in these fast matrix multiplication algorithms mean that the matrix dimensions necessary for these algorithms to be superior to classical algorithms are much larger than what current computers can effectively handle. Consequently, these algorithms are not always used in practice. We present the computational complexity of using fast matrix operations in Appendix G.*

In Appendix H, we also extend our algorithms to allow inexact auxiliary CRN sub-problem solving and analyze the total complexity. Specifically, we design an efficient sub-procedure (Algorithm 5) to solve the CRN sub-problem to desired accuracy in only $\mathcal{O}(\log\log(1/\epsilon))$ number of linear system solving. It tightens the $\mathcal{O}(\log(1/\epsilon))$ iteration complexity in (Bullins & Lai, 2022; Adil et al., 2022). Additionally, (Bullins & Lai, 2022; Adil et al., 2022) requires additionally assume $\sigma_{\min}(\nabla \boldsymbol{F}(\boldsymbol{z})) \geq \mu$ for some positive constant μ, which makes the problem similar to strongly-convex(-strongly-concave) problems, while our analysis does not require such an assumption.

Algorithm 3 Implementation of LEN $(\boldsymbol{z}_0, T, m, M)$

for $t = 0, \cdots, T-1$ **do**
  if $t \mod m = 0$ **do**
    Compute the Schur decomposition such that $\nabla \boldsymbol{F}(\boldsymbol{z}_t) = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{Q}^{-1}$.
  end if
  Let $\phi(\,\cdot\,)$ defined as 6 and compute γ_t as its root by a binary search.
  Compute lazy cubic step $\boldsymbol{z}_{t+1/2} = \boldsymbol{Q}(\boldsymbol{U} + \gamma_t \boldsymbol{I}_d)^{-1}\boldsymbol{Q}^{-1}\boldsymbol{F}(\boldsymbol{z}_t)$.
  Compute extra-gradient step $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \gamma_t^{-1}\boldsymbol{F}(\boldsymbol{z}_{t+1/2})$.
end for
return $\bar{\boldsymbol{z}}_T = \frac{1}{\sum_{t=0}^{T-1}\gamma_t^{-1}} \sum_{t=0}^{T-1} \gamma_t^{-1} \boldsymbol{z}_{t+1/2}$.



(a) $n = 10$ (b) $n = 100$ (c) $n = 200$

(d) $n = 10$ (e) $n = 100$ (f) $n = 200$

Figure 1: We demonstrate running time *v.s.* gradient norm $\|\boldsymbol{F}(\boldsymbol{z})\|$ and *v.s.* distance to saddle point $\|\boldsymbol{z} - \boldsymbol{z}^*\|$ for Problem (11) with different sizes: $n \in \{10, 100, 200\}$.

5 Numerical Experiments

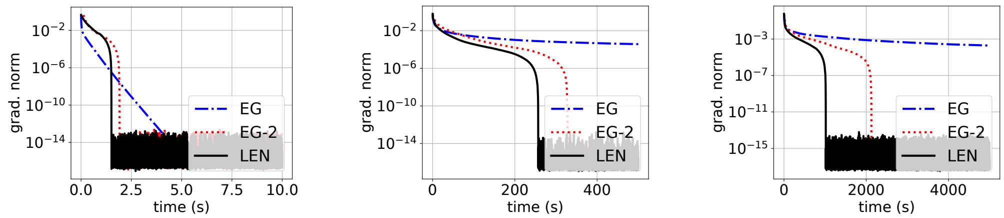
We conduct our algorithms on a regularized bilinear min-max problem and fairness-aware machine learning tasks. We include EG (Korpelevich, 1976) and second-order extension of EG (Monteiro & Svaiter, 2012; Adil et al., 2022; Bullins & Lai, 2022) (which is our algorithm with $m = 1$) as the baselines since they are the optimal first-order and second-order methods for convex-concave minimax problems, respectively. We run the programs on an AMD EPYC 7H12 64-Core Processor.

5.1 Regularized bilinear min-max problem

We first conduct numerical experiments on the cubic regularized bilinear min-max problem considered in the literature (Alves & Svaiter, 2023; Huang & Zhang, 2022; Jiang et al., 2024):

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \max_{\boldsymbol{y}\in\mathbb{R}^n} f(\boldsymbol{x}, \boldsymbol{y}) = \frac{\rho}{6}\|\boldsymbol{x}\|^3 + \boldsymbol{y}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}). \tag{11}$$

The function $f(\boldsymbol{x}, \boldsymbol{y})$ is convex-concave and has ρ-Lipschitz continuous Hessians. Moreover, the unique saddle point $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ of $f(\boldsymbol{x}, \boldsymbol{y})$ can be explicitly calculated as $\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\boldsymbol{y}^* = -\rho\|\boldsymbol{x}^*\|^2(\boldsymbol{A}^\top)^{-1}\boldsymbol{x}^*/2$, so we can compute the distance to $\boldsymbol{z}^*$ to measure the performance of algorithms. Following Lin et al. (2022), we generate each element in $\boldsymbol{b}$ as independent Rademacher



(a) heart (b) adult (c) law school

Figure 2: We demonstrate running time *v.s.* gradient norm $\|\boldsymbol{F}(\boldsymbol{z})\|$ for fairness-aware machine learning task (Problem (12)) on datasets "heart", "adult", and "law school".

variables in $\{-1,+1\}$, set the regularization coefficient $\rho = 1/(20n)$ and the matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \\ & & & 1 \end{bmatrix}.$$

We compare our methods with the baselines on different sizes of problem: $n \in \{100, 200, 500\}$. For EG, we tune the stepsize in $\{1, 0.1, 0.01, 0.001\}$. For LEN, we vary m in $\{1, 2, 10, 100\}$. The results of running time against $\|\boldsymbol{F}(\boldsymbol{z})\|$ and $\|\boldsymbol{z} - \boldsymbol{z}^*\|$ is presented in Figure 1.

5.2 Fairness-Aware Machine Learning

We then examine our algorithm in the task of fairness-aware machine learning. Let $\{\boldsymbol{a}_i, b_i, c_i\}_{i=1}^n$ be the training set, where $\boldsymbol{a}_i \in \mathbb{R}^{d_x}$ denotes the features of the i-th sample, $b_i \in \{-1,+1\}$ is the corresponding label, and $c_i \in \{-1,+1\}$ is an additional feature that is required to be protected and debiased. For example, c_i can denote the gender. Zhang et al. (2018) proposed to solve the following minimax problem to mitigate unwanted bias of c_i by adversarial learning:

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \ell(b_i \boldsymbol{a}_i^\top \boldsymbol{x}) - \beta \ell(c_i y \boldsymbol{a}_i^\top \boldsymbol{x}) + \lambda \|\boldsymbol{x}\|^2 - \gamma y^2, \tag{12}$$

where ℓ is the logit function such that $\ell(t) = \log(1 + \exp(-t))$. We set $\lambda = \gamma = 10^{-4}$ and $\beta = 0.5$. We conduct the experiments on datasets "heart" ($n = 270$, $d_x = 13$) (Chang & Lin, 2011), "adult" ($n = 32,561$, $d_x = 123$) (Chang & Lin, 2011) and "law school" ($n = 20,798$, $d_x = 380$) (Le Quy et al., 2022). For all the datasets, we choose "gender" as the protected feature. For EG, we tune the stepsize in $\{0.1, 0.01, 0.001\}$. For second-order methods (EG-2 and LEN), as we do not know the value of ρ in advance, we view it as a hyperparameter and tune it in $\{1, 10, 100\}$. We set $m = 10$ for LEN and we find this simple choice performs well in all the datasets we test. We show the results of running time against gradient norm $\|\boldsymbol{F}(\boldsymbol{z})\|$ in Figure 2.

6 Conclusion and Future Works

In this paper, we propose LEN and LEN-restart for C-C and SC-SC min-max problems, respectively. By using lazy Hessian updates, our methods improve the computational complexity of the current best-known second-order methods by a factor of $d^{1/3}$. Numerical experiments on both real and synthetic datasets also verify the efficiency of our method.

For future works, it will be interesting to extend our idea to adaptive second-order methods (Doikov et al., 2024; Carmon et al., 2022; Antonakopoulos et al., 2022; Liu & Luo, 2022b) or stochastic problems with sub-sampled Newton methods (Lin et al., 2022; Chayti et al., 2023; Zhou et al., 2019; Tripuraneni et al., 2018; Wang et al., 2019). Besides, our methods only focus on the convex-concave case, it is also possible to reduce the Hessian oracle for the nonconvex-concave problems (Luo et al., 2022; Lin et al., 2020; Zhang et al., 2022b) or study the structured nonconvex-nonconcave problems (Zheng et al., 2024; Diakonikolas et al., 2021; Lee & Kim, 2021; Chen & Luo, 2022).

REFERENCES

Deeksha Adil, Brian Bullins, Arun Jambulapati, and Sushant Sachdeva. Optimal methods for higher-order smooth monotone variational inequalities. *arXiv preprint arXiv:2205.06167*, 2022.

Ilan Adler, Zhiyue T. Hu, and Tianyi Lin. New proximal newton-type methods for convex optimization. In *CDC*, 2020.

M. Marques Alves and Benar F. Svaiter. A search-free $\mathcal{O}(1/k^{3/2})$ homotopy inexact proximal-newton extragradient algorithm for monotone variational inequalities. *arXiv preprint arXiv:2308.05887*, 2023.

Kimon Antonakopoulos, Ali Kavis, and Volkan Cevher. Extra-newton: A first approach to noise-adaptive accelerated second-order methods. In *NeurIPS*, 2022.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

Brian Bullins and Kevin A. Lai. Higher-order methods for convex-concave min-max optimization and monotone variational inequalities. *SIAM Journal on Optimization*, 32(3):2208–2229, 2022.

Yair Carmon, Danielle Hausler, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Optimal and adaptive monteiro-svaiter acceleration. In *NeurIPS*, 2022.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

El Mahdi Chayti, Nikita Doikov, and Martin Jaggi. Unified convergence theory of stochastic and variance-reduced cubic newton methods. *arXiv preprint arXiv:2302.11962*, 2023.

Lesi Chen and Luo Luo. Near-optimal algorithms for making the gradient small in stochastic minimax optimization. *arXiv preprint arXiv:2208.05925*, 2022.

James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numerische Mathematik*, 108(1):59–91, 2007.

Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *AISTATS*, 2021.

Nikita Doikov, El Mahdi Chayti, and Martin Jaggi. Second-order optimization with lazy hessians. In *ICML*, 2023.

Nikita Doikov, Konstantin Mishchenko, and Yurii Nesterov. Super-universal regularized newton method. *SIAM Journal on Optimization*, 34(1):27–56, 2024.

Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *ICML*, 2017.

Jinyan Fan. A shamanskii-like levenberg-marquardt method for nonlinear equations. *Computational Optimization and Applications*, 56(1):63–80, 2013.

Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *ICML*, 2016.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

Kevin Huang and Shuzhong Zhang. An approximation-based regularized extra-gradient method for monotone variational inequalities. *arXiv preprint arXiv:2210.04440*, 2022.

Kevin Huang, Junyu Zhang, and Shuzhong Zhang. Cubic regularized newton method for the saddle point models: A global and local convergence analysis. *Journal of Scientific Computing*, 91(2): 60, 2022.

Ruichen Jiang and Aryan Mokhtari. Generalized optimistic methods for convex-concave saddle point problems. *arXiv preprint arXiv:2202.09674*, 2022.

Ruichen Jiang, Ali Kavis, Qiujiang Jin, Sujay Sanghavi, and Aryan Mokhtari. Adaptive and optimal second-order optimistic methods for minimax optimization. *arXiv preprint arXiv:2406.02016*, 2024.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Dmitry Kovalev, Adil Salim, and Peter Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. In *NeurIPS*, 2020.

Dmitry Kovalev, Elnur Gasanov, Alexander Gasnikov, and Peter Richtarik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. In *NeurIPS*, 2021.

Francesco Lampariello and Marco Sciandrone. Global convergence technique for the newton method with periodic hessian evaluation. *Journal of optimization theory and applications*, 111: 341–358, 2001.

Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(3):e1452, 2022.

Sucheol Lee and Donghwan Kim. Fast extra gradient methods for smooth structured nonconvex-nonconcave minimax problems. In *NeurIPS*, 2021.

Tianyi Lin and Michael I. Jordan. A control-theoretic perspective on optimal high-order optimization. *Mathematical Programming*, 2022.

Tianyi Lin and Michael I Jordan. Perseus: A simple high-order regularization method for variational inequalities. *Mathematical Programming*, pp. 1–42, 2024.

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

Tianyi Lin, Panayotis Mertikopoulos, and Michael I Jordan. Explicit second-order min-max optimization methods with optimal convergence guarantee. *arXiv preprint arXiv:2210.12860*, 2022.

Chengchang Liu and Luo Luo. Quasi-newton methods for saddle point problems. In *NeurIPS*, 2022a.

Chengchang Liu and Luo Luo. Regularized newton methods for monotone variational inequalities with Holders continuous jacobians. *arXiv preprint arXiv:2212.07824*, 2022b.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

Luo Luo, Yujun Li, and Cheng Chen. Finding second-order stationary points in nonconvex-strongly-concave minimax optimization. In *NeurIPS*, 2022.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, 2015.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *AISTATS*, 2020a.

Aryan Mokhtari, Asuman E. Ozdaglar, and Sarath Pattathil. Convergence rate of $\mathcal{O}(1/k)$ for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020b.

Renato DC Monteiro and Benar F Svaiter. Iteration-complexity of a newton proximal extragradient method for monotone variational inequalities and inclusion problems. *SIAM Journal on Optimization*, 22(3):914–935, 2012.

Renato DC Monteiro and Benar Fux Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

Arkadi Nemirovski. Prox-method with rate of convergence $\mathcal{O}(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

Yurii Nesterov. High-order reduced-gradient methods for composite variational inequalities. *arXiv preprint arXiv:2311.15154*, 2023.

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

Yurii Nesterov and Laura Scrimali. Solving strongly monotone variational and quasi-variational inequalities. 2006.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 68(3):1321–1336, 2022.

Leonid Denisovich Popov. A modification of the arrow-hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28:845–848, 1980.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

VE Shamanskii. A modification of newton's method. *Ukrainian Mathematical Journal*, 19(1): 118–122, 1967.

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

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. In *NeurIPS*, 2018.

Chang-yu Wang, Yuan-yuan Chen, and Shou-qiang Du. Further insight into the shamanskii modification of newton method. *Applied mathematics and computation*, 180(1):46–52, 2006.

Mengdi Wang. Primal-dual π-learning: Sample complexity and sublinear run time for ergodic markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Stochastic variance-reduced cubic regularization for nonconvex optimization. In *AISTATS*, 2019.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *SODA*, 2024.

Stephen J. Wright. Numerical optimization, 2006.

Yangyang Xu. Primal-dual stochastic gradient method for convex programs with many functional constraints. *SIAM Journal on Optimization*, 30(2):1664–1692, 2020.

Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online auc maximization. In *NIPS*, 2016.

Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *ICCV*, 2021.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Junyu Zhang, Mingyi Hong, and Shuzhong Zhang. On lower iteration complexity bounds for the convex concave saddle point problems. *Mathematical Programming*, 194(1-2):901–935, 2022a.

Xuan Zhang, Necdet Serhat Aybat, and Mert Gurbuzbalaban. SAPD+: An accelerated stochastic method for nonconvex-concave minimax problems. In *NeurIPS*, 2022b.

Taoli Zheng, Linglingzhi Zhu, Anthony Man-Cho So, José Blanchet, and Jiajin Li. Universal gradient descent ascent method for nonconvex-nonconcave minimax optimization. In *NeurIPS*, 2024.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularization methods. *JMLR*, 20(134):1–47, 2019.

A Some Useful Lemmas

Lemma A.1 (Proposition 2.8 Lin et al. (2022))**.** *Let*

$$\bar{\boldsymbol{x}}_t = \frac{1}{\sum_{i=0}^{t-1}\eta_i}\sum_{i=0}^{t-1}\eta_i\boldsymbol{x}_i, \quad \bar{\boldsymbol{y}}_t = \frac{1}{\sum_{i=0}^{t-1}\eta_i}\sum_{i=0}^{t-1}\eta_i\boldsymbol{y}_i.$$

Then under Assumption 3.2, for any $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y})$, it holds that

$$f(\bar{\boldsymbol{x}}_t, \boldsymbol{y}) - f(\boldsymbol{x}, \bar{\boldsymbol{y}}_t) \le \frac{1}{\sum_{i=0}^{t-1}\eta_i}\sum_{i=0}^{t-1}\eta_i\langle \boldsymbol{F}(\boldsymbol{z}_i), \boldsymbol{z}_i - \boldsymbol{z}\rangle.$$

B Proof of Lemma 4.2

Proof. We prove the result by induction.

Apparently, it is true for $m = 2$, which is the induction base.

Assume that it holds for $m \ge 2$. Then

$$\begin{aligned}
&\sum_{t=1}^{m}\left(\sum_{i=0}^{t-1} r_i\right)^2 \\
&= \sum_{t=1}^{m-1}\left(\sum_{i=0}^{t-1} r_i\right)^2 + \left(\sum_{i=0}^{m-1} r_i\right)^2 \\
&\le \frac{m^2}{2}\sum_{t=0}^{m-1} r_t^2 + m\sum_{t=0}^{m-1} r_t^2 \\
&\le \left(\frac{m^2 + 2m}{2}\right)\sum_{t=0}^{m-1} r_t^2 \\
&\le \frac{(m+1)^2}{2}\sum_{t=0}^{m-1} r_t^2.
\end{aligned}$$

□

C Proof of Lemma 4.1

Proof. Instead of directly providing a proof for Algorithm 1, we give the proof for the more general inexact algorithm (Algorithm 4), which recovers Algorithm 1 if $\alpha = 1$.

For convenience, we denote $\eta_t = 1/\gamma_t$.

For ant $\boldsymbol{z} \in \mathbb{R}^d$, we have

$$\begin{aligned}
&\eta_t\langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z}\rangle \\
&= \langle \boldsymbol{z}_t - \boldsymbol{z}_{t+1}, \boldsymbol{z}_{t+1/2} - \boldsymbol{z}\rangle \\
&= \langle \boldsymbol{z}_t - \boldsymbol{z}_{t+1}, \boldsymbol{z}_{t+1} - \boldsymbol{z}\rangle + \langle \boldsymbol{z}_t - \boldsymbol{z}_{t+1}, \boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\rangle \\
&= \langle \boldsymbol{z}_t - \boldsymbol{z}_{t+1}, \boldsymbol{z}_{t+1} - \boldsymbol{z}\rangle + \langle \boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}, \boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\rangle + \langle \boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}, \boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\rangle \\
&= \frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_{t+1} - \boldsymbol{z}\|^2 - \cancel{\frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1}\|^2} \\
&\quad + \cancel{\frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1}\|^2} - \frac{1}{2}\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\|^2 - \frac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2 + \|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_{t+1}\|^2.
\end{aligned} \tag{13}$$

Note that by the updates of the algorithm, we have that

$$\gamma_t(z_t - z_{t+1/2}) = F(z_t) + \nabla F(z_{\pi(t)})(z_{t+1/2} - z_t),$$
$$\gamma_t(z_t - z_{t+1}) = F(z_{t+1/2}).$$

It implies that

$$\begin{aligned} & z_{t+1/2} - z_{t+1} \\ &= \eta_t(F(z_{t+1/2}) - F(z_t) - \nabla F(z_{\pi(t)})(z_{t+1/2} - z_t))) \\ &= \eta_t(F(z_{t+1/2}) - F(z_t) - \nabla F(z_t)(z_{t+1/2} - z_t)) + \eta_t(\nabla F(z_{\pi(t)})) - \nabla F(z_t))(z_{t+1/2} - z_t) \end{aligned} \tag{14}$$

Note that ∇F is ρ-Lipschitz continuous. Taking norm on both sides of (14), we have that

$$\begin{aligned} \|z_{t+1/2} - z_{t+1}\| &\le \frac{\rho\eta_t}{2}\|z_{t+1/2} - z_t\|^2 + \rho\eta_t\|z_{\pi(t)} - z_t\|\|z_{t+1/2} - z_t\| \\ &\le \frac{\rho}{2M}\|z_{t+1/2} - z_t\| + \frac{\rho}{M}\|z_{\pi(t)} - z_t\|, \end{aligned}$$

where we use the condition $M\|z_t - z_{t+1/2}\| \le \gamma_t$ in the last step.

By Young's inequality, this further means

$$\|z_{t+1/2} - z_{t+1}\|^2 \le \frac{\rho^2}{2M^2}\|z_{t+1/2} - z_t\|^2 + \frac{2\rho^2}{M^2}\|z_{\pi(t)} - z_t\|^2.$$

Plug the above inequality into the last term in (13).

$$\begin{aligned} & \eta_t\langle F(z_{t+1/2}), z_{t+1/2} - z\rangle \\ &\le \frac{1}{2}\|z_t - z\|^2 - \frac{1}{2}\|z_{t+1} - z\|^2 - \frac{1}{2}\|z_{t+1/2} - z_{t+1}\|^2 \\ &\quad - \frac{1}{2}\|z_t - z_{t+1/2}\|^2 + \frac{\rho^2}{2M^2}\|z_t - z_{t+1/2}\|^2 + \frac{2\rho^2}{M^2}\|z_{\pi(t)} - z_t\|^2. \end{aligned}$$

□

D PROOF OF THEOREM 4.1

Proof. When $m = 1$, the algorithm reduces to the EG-2 algorithm (Huang & Zhang, 2022; Lin et al., 2022; Adil et al., 2022). When $m \ge 2$, we use Lemma 4.2 to bound the error that arises from lazy Hessian updates.

Instead of directly providing a proof for Algorithm 1, we give the proof for the more general inexact algorithm (Algorithm 4), which recovers Algorithm 1 if $\alpha = 1$.

Define $r_t = \|z_{t+1} - z_t\|$. By triangle inequality and Young's inequality, we have

$$\begin{aligned} & \eta_t\langle F(z_{t+1/2}), z_{t+1/2} - z\rangle \\ &\le \frac{1}{2}\|z_t - z\|^2 - \frac{1}{2}\|z_{t+1} - z\|^2 - \left(\frac{1}{4} - \frac{\rho^2}{2M^2}\right)\|z_t - z_{t+1/2}\|^2 \\ &\quad - \left(\frac{1}{8}r_t^2 - \frac{2\rho^2}{M^2}\left(\sum_{i=\pi(t)}^{t-1} r_i\right)^2\right). \end{aligned}$$

For any $1 \le s \le m$. Telescoping over $t = \pi(t), \cdots, \pi(t) + s - 1$, we have

$$\begin{aligned} & \sum_{t=\pi(t)}^{\pi(t)+s-1} \eta_t\langle F(z_{t+1/2}), z_{t+1/2} - z\rangle \\ &\le \frac{1}{2}\|z_{\pi(t)} - z\|^2 - \frac{1}{2}\|z_{\pi(t)+s} - z\|^2 - \left(\frac{1}{4} - \frac{\rho^2}{2M^2}\right)\sum_{t=\pi(t)}^{\pi(t)+s-1}\|z_t - z_{t+1/2}\|^2 \\ &\quad - \left(\frac{1}{8}\sum_{t=\pi(t)}^{\pi(t)+s-1} r_t^2 - \frac{2\rho^2}{M^2}\sum_{t=\pi(t)+1}^{\pi(t)+s-1}\left(\sum_{i=\pi(t)}^{t-1} r_i\right)^2\right). \end{aligned}$$

Applying Lemma 4.2, we further have

$$\sum_{t=\pi(t)}^{\pi(t)+s-1} \eta_t \langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z} \rangle$$

$$\leq \frac{1}{2}\|\boldsymbol{z}_{\pi(t)} - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_{\pi(t)+s} - \boldsymbol{z}\|^2 - \left(\frac{1}{4} - \frac{\rho^2}{2M^2}\right) \sum_{t=\pi(t)}^{\pi(t)+s-1} \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2$$

$$- \left(\frac{1}{8} - \frac{\rho^2 s^2}{M^2}\right) \sum_{t=\pi(t)}^{\pi(t)+s-1} r_t^2.$$

Note that $s \leq m$. Let $M \geq 3\rho m$. Then

$$\sum_{t=\pi(t)}^{\pi(t)+s-1} \eta_t \langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z} \rangle$$

$$\leq \frac{1}{2}\|\boldsymbol{z}_{\pi(t)} - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_{\pi(t)+s} - \boldsymbol{z}\|^2 - \frac{1}{8} \sum_{t=\pi(t)}^{\pi(t)+s-1} \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2.$$

Let $s = m$ and further telescope over $t = 0, \cdots, T-1$. Then

$$\sum_{t=0}^{T} \eta_t \langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z} \rangle \leq \frac{1}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}\|^2 - \frac{1}{2}\|\boldsymbol{z}_T - \boldsymbol{z}\|^2 - \frac{1}{8} \sum_{t=0}^{T} \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|^2. \quad (15)$$

This inequality is the key to the convergence. It implies the following results. First, letting $\boldsymbol{z} = \boldsymbol{z}^*$ and using the fact that $\langle \boldsymbol{F}(\boldsymbol{z}_{t+1/2}), \boldsymbol{z}_{t+1/2} - \boldsymbol{z}^* \rangle \geq 0$ according to monotonicity of $\boldsymbol{F}$, we can prove the iterate is bounded

$$\|\boldsymbol{z}_t - \boldsymbol{z}^*\| \leq \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|, \quad \text{and} \quad \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\| \leq 2\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|, \quad t = 0, \cdots, T-1. \quad (16)$$

Then using triangle inequality, we obtain

$$\|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}^*\| \leq 3\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|, \quad \forall t = 0, \cdots, T-1.$$

Second, as (15) holds for all $\boldsymbol{z} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*)$, Lemma A.1 indicates

$$\operatorname{Gap}(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T; 3\beta) \leq \frac{\max_{\boldsymbol{z} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*)} \|\boldsymbol{z}_0 - \boldsymbol{z}\|^2}{2\sum_{t=0}^{T-1} \eta_t} \leq \frac{16\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2}{2\sum_{t=0}^{T-1} \eta_t}, \quad (17)$$

where the last step uses $\|\boldsymbol{z}_0 - \boldsymbol{z}\| \leq \|\boldsymbol{z}_0 - \boldsymbol{z}^*\| + \|\boldsymbol{z} - \boldsymbol{z}^*\| \leq 4\beta$ for any $\boldsymbol{z} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*)$.

Third, we can also use (15) to lower bound $\sum_{t=0}^{T-1} \eta_t$. (15) with $\boldsymbol{z} = \boldsymbol{z}^*$ implies

$$\sum_{t=0}^{T} \gamma_t^2 \leq 4\alpha^2 M^2 \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2,$$

where we use the condition $\gamma_t \leq \alpha M \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|$ in the last step. Then by Holder's inequality,

$$T = \sum_{t=0}^{T-1} (\eta_t)^{2/3} \left(\gamma_t^2\right)^{1/3} \leq \left(\sum_{t=0}^{T-1} \eta_t\right)^{2/3} \left(\sum_{t=0}^{T-1} \gamma_t^2\right)^{1/3}.$$

Therefore,

$$\sum_{t=0}^{T-1} \eta_t \geq \frac{T^{3/2}}{2\alpha M \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|}. \quad (18)$$

We plug in (18) to (17) and obtain that

$$\operatorname{Gap}(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T; \beta) \leq \frac{16\alpha M \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^3}{T^{3/2}}.$$

The desired theorem is the case $\alpha = 1$. □

E PROOF OF THEOREM 4.2

Proof. Using the strongly monotonicity of operator $\boldsymbol{F}$ in (15), we obtain that

$$\sum_{t=0}^{T} \mu\eta_t \|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}^*\|^2 \leq \frac{1}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2 - \frac{1}{2}\|\boldsymbol{z}_T - \boldsymbol{z}^*\|^2.$$

Using Jensen's inequality, for each epoch, we have

$$\|\bar{\boldsymbol{z}}_T - \boldsymbol{z}^*\|^2 \leq \frac{\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2}{2\mu\sum_{t=0}^{T-1}\eta_t} \leq \frac{M\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^3}{\mu T^{3/2}} := c\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2.$$

Next, we consider the iterate $\{\boldsymbol{z}^{(s)}\}_{s=0}^{S-1}$. For the first epoch, the setting of T ensures $c \leq 1/2$:

$$\|\boldsymbol{z}^{(1)} - \boldsymbol{z}^*\|^2 \leq \frac{1}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2.$$

Then for the second one, it is improved by

$$\|\boldsymbol{z}^{(2)} - \boldsymbol{z}^*\|^2 \leq \frac{\|\boldsymbol{z}^{(1)} - \boldsymbol{z}^*\|^3}{2\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|} \leq \left(\frac{1}{2}\right)^{1+3/2} \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2.$$

Keep repeating this process. We can get

$$\|\boldsymbol{z}^{(s)} - \boldsymbol{z}^*\|^2 \leq \left(\frac{1}{2}\right)^{q_s} \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2,$$

where q_s satisfies the recursion

$$q_s = \begin{cases} 1, & s = 1; \\ \frac{3}{2}q_{s-1} + 1, & s \geq 2. \end{cases}$$

This implies

$$\|\boldsymbol{z}^{(s)} - \boldsymbol{z}^*\|^2 \leq \left(\frac{1}{2}\right)^{\left(\frac{3}{2}\right)^{s-1}+1} \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2.$$

Set $m = \Theta(d)$, LEAN-restart takes $\mathcal{O}(d^{2/3}\kappa^{2/3}\log\log(1/\epsilon))$ oracle to $\boldsymbol{F}(\cdot)$ and $\mathcal{O}((1 + d^{-1/3}\kappa^{2/3})\log\log(1/\epsilon))$ oracle to $\nabla\boldsymbol{F}(\cdot)$. Under Assumption 3.4, the computational complexities of the oracles is

$$\begin{aligned} &\mathcal{O}\left(N \cdot d^{2/3}\kappa^{2/3}\log\log(1/\epsilon) + Nd \cdot (1 + d^{-1/3}\kappa^{2/3})\log\log(1/\epsilon)\right) \\ &= \mathcal{O}\left((Nd + Nd^{2/3}\kappa^{2/3})\log\log(1/\epsilon)\right). \end{aligned}$$

□

F PROOF OF COROLLARY 4.1

Proof. The computational complexity of inner loop can be directly obtained by replacing ϵ^{-1} by κ in Theorem 4.3 such that

$$\text{Inner Computational Complexity} = \tilde{\mathcal{O}}\left((N + d^2) \cdot (d + d^{2/3}\kappa^{2/3})\right).$$

The iterations of outer loop is $S = \log\log(1/\epsilon)$, thus, the total computational complexity of LEAN-restart is

$$S \cdot \text{ Inner Computational Complexity} = \tilde{\mathcal{O}}\left((N + d^2) \cdot (d + d^{2/3}\kappa^{2/3})\right).$$

□

Algorithm 4 Inexact LEN$(\boldsymbol{z}_0, T, m, M, \alpha)$

1: **for** $t = 0, \cdots, T-1$ **do**
2: Use Algorithm 5 to find $(\boldsymbol{z}_{t+1/2}, \gamma_t)$ that satisfies

$$\boldsymbol{z}_{t+1/2} = \boldsymbol{z}_t - (\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + \gamma_t \boldsymbol{I}_d)^{-1} \boldsymbol{F}(\boldsymbol{z}_t)$$

and $M\|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\| \leq \gamma_t \leq \alpha M \|\boldsymbol{z}_t - \boldsymbol{z}_{t+1/2}\|$ for given $\alpha \geq 1$.
3: Compute extra-gradient step $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \gamma_t^{-1} \boldsymbol{F}(\boldsymbol{z}_{t+1/2})$.
4: **end for**
5: **return** $\bar{\boldsymbol{z}}_T = \frac{1}{\sum_{t=0}^{T-1} \gamma_t^{-1}} \sum_{t=0}^{T-1} \gamma_t^{-1} \boldsymbol{z}_{t+1/2}$.

G Computational Complexity Using Fast Matrix Operations

Theoretically, one may use fast matrix operations for Schur decomposition and matrix inversion (Demmel et al., 2007), with a computational complexity of d^ω, where $\omega \approx 2.371552$ is the matrix multiplication constant. In this case, the total computational complexity of Algorithm 3 is

$$\tilde{\mathcal{O}}\left(\left(\frac{Nd + d^\omega}{m} + d^2 + N\right) m^{2/3} \epsilon^{-2/3}\right)$$

Setting the optimal m, we obtain the following complexity of Algorithm 3:

$$\begin{cases} \tilde{\mathcal{O}}(d^{\frac{2}{3}(\omega+1)} \epsilon^{-2/3}) \ \text{(with } m = d^{\omega-2}), & N \lesssim d^{\omega-1} \\ \tilde{\mathcal{O}}(N^{2/3} d^{4/3} \epsilon^{-2/3}) \ \text{(with } m = N/d), & d^{\omega-1} \lesssim N \lesssim d^2 \\ \tilde{\mathcal{O}}(N d^{2/3} \epsilon^{-2/3}) \ \text{(with } m = d), & d^2 \lesssim N. \end{cases}$$

Our result is always better than the $\mathcal{O}((Nd + d^\omega)\epsilon^{-2/3})$ of existing optimal second-order methods.

H The Inexact Algorithm

Algorithm 1 requires a cubic regularized Newton (CRN) oracle (Implicit Step, (4)). We provide implementation details for the CRN oracle in Section 4.3. One missing detail is that we can not obtain the exact solution to the CRN oracle in practice. To make our result more rigorous, we analyze the inexact LEN (Algorithm 1), which allows inexact sub-problem solving with a parameter $\alpha \geq 1$. Note that this algorithm reduces to the exact version (Algorithm 1) when $\alpha = 1$.

Below, we present the following theorem as the inexact version of Theorem 4.1.

Theorem H.1. *Suppose that Assumption 3.1 and 3.2 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point and $\beta = \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$. Set $M \geq 3\rho m$. The sequence of iterates generated by Algorithm 4 is bounded $\boldsymbol{z}_t \in \mathbb{B}_\beta(\boldsymbol{z}^*), \ \boldsymbol{z}_{t+1/2} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*), \quad \forall t = 0, \cdots, T-1$, and satisfies the following ergodic convergence:*

$$\mathrm{Gap}(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T; 3\beta) \leq \frac{16\alpha M \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^3}{T^{3/2}}.$$

Let $M = 3\rho m$ and $\alpha = 2$. Algorithm 1 finds an ϵ-saddle point within $\mathcal{O}(m^{2/3}\epsilon^{-2/3})$ iterations.

Proof. See Section D. □

The only remaining thing is to show how to compute γ_t in the auxiliary problem (Line 2 in Algorithm 4). Below, we present an efficient sub-procedure to achieve the desired goal using the standard Newton step. We define the monotone operator $\boldsymbol{A}_t : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\boldsymbol{A}_t(\boldsymbol{z}) = \boldsymbol{F}(\boldsymbol{z}_t) + \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z} - \boldsymbol{z}_t). \tag{19}$$

Then we can write down the (regularized) Newton step as

$$\begin{aligned} \boldsymbol{z}_{t+1/2}(\eta; \boldsymbol{z}_t) &:= \boldsymbol{z}_t - (\nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}) + \eta^{-1}\boldsymbol{I}_d)^{-1})\boldsymbol{F}(\boldsymbol{z}_t) \\ &= (\boldsymbol{I}_d + \eta \boldsymbol{A}_t)^{-1}(\boldsymbol{z}_t). \end{aligned} \tag{20}$$

Algorithm 5 Bracketing/Bisection Procedure$(\boldsymbol{A}_t, \boldsymbol{z}_t, M, \alpha, \eta_t^0)$

1: **(Bracketing Stage)** Compute $\boldsymbol{z}_{t+1/2}^0 = (\boldsymbol{I}_d + \eta_t^0 \boldsymbol{A}_t)^{-1}(\boldsymbol{z}_t)$ with one Newton step.

- (1a) **if** $\eta_t^0 \|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| \in (\frac{1}{\alpha M}, \frac{1}{M})$, **then** let $\boldsymbol{z}_{t+1/2} = \boldsymbol{z}_{t+1/2}^0, \eta_t = \eta_t^0$ and **go to** Line 3.
- (1b) **if** $\eta_t^0 \|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| < \frac{1}{\alpha M}$, **then** set $c_t^- = \eta_t^0$ and $c_t^+ = \frac{1}{M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|}$;
- (1c) **if** $\eta_0^t \|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| > \frac{1}{M}$, **then** set $c_t^- = \frac{1}{\alpha M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|}$ and $c_t^+ = \eta_t^0$;

2: **(Bisection Stage)**

- (2a) set $\eta_t = \sqrt{c_t^- c_t^+}$ and compute $\boldsymbol{z}_{t+1/2} = (\boldsymbol{I}_d + \eta_t \boldsymbol{A}_t)^{-1}(\boldsymbol{z}_t)$ with one Newton step;
- (2b) **if** $\eta_t \|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t\| \in (\frac{1}{\alpha M}, \frac{1}{M})$, **then go to** Line 3;
- (2c) **if** $\eta_t \|\boldsymbol{z}_{t+1/2} - \boldsymbol{z}_t\| > \frac{1}{M}$, **then** set $c_t^+ = \eta_t$; **else** set $c_t^- = \eta_t$;
- (2d) **go to** step (2a).

3: **return** $(\boldsymbol{z}_{t+1/2}, \gamma_t)$ that meets the requirement of Line 2 in Algorithm 4, where $\gamma_t = 1/\eta_t$.

And the inexact condition (Line 2 in Algorithm 4) is

$$\frac{1}{\alpha M} \le \phi_t(\eta; \boldsymbol{z}_t) \le \frac{1}{M}, \tag{21}$$

where $\phi_t(\eta; \boldsymbol{z}_t)$ is defined as $\phi_t(\eta; \boldsymbol{z}_t) := \eta \|\boldsymbol{z}_{t+1/2}(\eta; \boldsymbol{z}_t) - \boldsymbol{z}_t\|$.

Note that a stepsize η that satisfies (21) directly implies $\gamma_t = 1/\eta$ satisfies the requirement of Line 2 in Algorithm 4. Therefore, the main goal of this section is to design a sub-procedure that can determine the stepsize η that satisfies (21).

A similar sub-procedure without using lazy Hessian updates has been proposed in (Monteiro & Svaiter, 2012). Below, we show that we can use a similar sub-procedure for our algorithm. We recall some useful lemmas in (Monteiro & Svaiter, 2012), which holds for any monotone operators $\boldsymbol{A}$. Below, we state their results when $\boldsymbol{A} = \boldsymbol{A}_t$.

Lemma H.1 (Lemma 4.3 and Lemma 4.4 (Monteiro & Svaiter, 2012))**.** *Recall the definition of ϕ_t right after (21). For any $\boldsymbol{z} \in \mathbb{R}^d$, the following statements hold:*

1. *For any $\eta > 0$, we have $\phi_t(\eta; \boldsymbol{z}) > 0$.*
2. *For any $0 < \eta' \le \eta$, we have that*

$$\frac{\eta}{\eta'} \phi_t(\eta'; \boldsymbol{z}) \le \phi_t(\eta; \boldsymbol{z}) \le \left(\frac{\eta}{\eta'}\right)^2 \phi_t(\eta'; \boldsymbol{z}).$$

   As a corollary, $\phi_t(\eta; \boldsymbol{z})$ is a continuous and strictly increasing function, which converges to 0 or $+\infty$ as η tends to 0 or $+\infty$, respectively.

3. *For any $0 < \beta^- < \beta^+$, the set of all scalars $\eta > 0$ satisfying $\beta^- \le \phi_t(\eta; \boldsymbol{z}) \le \beta^+$ is a closed interval $[\eta^-, \eta^+]$ such that $\eta^+/\eta^- \ge \sqrt{\beta^+/\beta^-}$.*

Algorithm 5 presents our sub-procedure to output the tuple $(\boldsymbol{z}_{t+1/2}, \gamma_t)$ satisfying (21). Similar to (Monteiro & Svaiter, 2012), the procedure consists of two stages. The first one is a bracketing stage, which either outputs an acceptable solution or an initial interval $[c_t^-, c_t^+]$ that contains all the η satisfying (21). The second one is a bisection stage, which uses binary search in the logarithmic scale to find a stepsize η satisfying (21). Note that the log-scale binary search would finally lead to a $\mathcal{O}(\log\log(1/\epsilon))$ iteration complexity, which improves the $\mathcal{O}(\log(1/\epsilon))$ iteration complexity using naive binary search in (Adil et al., 2022; Bullins & Lai, 2022).

Our first result of Algorithm 5 is the correctness of the bracketing stage, stated as follows.

Lemma H.2. *Let $[\eta_t^-, \eta_t^+]$ be the interval that contains all the stepsizes satisfying (21). Compute $\boldsymbol{z}_{t+1/2}^0 = (\boldsymbol{I}_d + \eta_t^0 \boldsymbol{A}_t)^{-1}(\boldsymbol{z}_t)$ with one Newton step as Algorithm 5. The following statements hold:*

1. *if $\eta_t^0 \|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| < \frac{1}{\alpha M}$, then $\eta_t^0 < \eta_t^-$ and $\eta_t^+ \le \frac{1}{M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|}$;*
2. *if $\eta_t^0 \|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| > \frac{1}{M}$, then $\eta_t^+ < \eta_t^0$ and $\frac{1}{\alpha M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|} \le \eta_t^-$.*

Proof. We only prove the first claim since the proof of the second claim follows in a similar manner.

Recall the definition of ϕ_t right after (21). The condition $\eta_t^0\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| < \frac{1}{\alpha M}$ is equivalent to $\phi_t(\eta_t^0; \boldsymbol{z}_t) < \phi_t(\eta_t^-; \boldsymbol{z}_t)$. Firstly. the fact that $\phi_t(\eta_t^-; \boldsymbol{z}_t)$ is a strictly increasing function according to the second statement in Lemma H.1, we know that $\eta_t^0 < \eta_t^-$.

Secondly, using the inequality in the second statement of Lemma H.1, we know that

$$\eta_t^+\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| = \frac{\eta_t^+}{\eta_t^0}\phi_t(\eta_t^0; \boldsymbol{z}_t) \le \phi_t(\eta_t^+; \boldsymbol{z}_t) = \frac{1}{M},$$

which implies $\eta_t^+ \le \frac{1}{M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|}$ by rearranging.

□

Therefore, the bracketing stage can always output an interval that contains the acceptable stepsizes η satisfying (21). Given such a valid initial interval, the bisection stage always find an acceptable stepsize, stated as follows.

Lemma H.3. *Consider Algorithm 5. If the bracketing stage outputs an interval $[c_t^-, c_t^+]$ containing all the stepsizes η satisfying (21), which is then input to the bisection stage, then the number of Newton step during the bisection stage is bounded by $1 + \log(\log(h_t)/\log\alpha)$, where*

$$h_t = \max\left\{\frac{1}{\eta_t^0 M\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|}, \alpha M\eta_t^0\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|\right\} \tag{22}$$

is the maximal ratio of c_t^+/c_t^-.

Proof. After j steps of bisection iterations, we have that $\log\frac{c_t^+}{c_t^-} = \frac{1}{2^j}\log h_t$. In view of the third statement in Lemma H.1, we know that $c_t^+/c_t^- \ge \sqrt{\alpha}$. These two inequalities immediately imply that the bisection stage would terminates in $j \le 1 + \log(\log(h_t)/\log\alpha)$ iterations. □

Our goal from now on would be giving a uniform upper bound of h_t all for t, which can imply the total complexity of our algorithm. From the definition of h_t in (22), we need to give both lower and upper bounds of $\eta_t^0\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|$. We recall some technical lemmas in (Monteiro & Svaiter, 2012).

Lemma H.4 (Proposition 4.5 Monteiro & Svaiter (2012))**.** *Let $\boldsymbol{A} : \mathbb{R}^d \to \mathbb{R}^d$ be a monotone operator. For a point $\boldsymbol{z}^* \in \mathbb{R}^d$ such that $\boldsymbol{A}(\boldsymbol{z}^*) = 0$, for any $\eta > 0$ and $\boldsymbol{z} \in \mathbb{R}^d$ it holds that*

$$\max\left\{\|(\boldsymbol{I}_d + \eta\boldsymbol{A})^{-1}\boldsymbol{z} - \boldsymbol{z}^*\|, \|(\boldsymbol{I}_d + \eta\boldsymbol{A})^{-1}\boldsymbol{z} - \boldsymbol{z}\|\right\} \le \|\boldsymbol{z} - \boldsymbol{z}^*\|.$$

From now on, we will fix all the η_t^0 in all the iterations such that $\eta_t^0 = \bar{\eta}$ and analyze Algorithm 4. The following lemma shows a uniform upper bound of $\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|$.

Lemma H.5 (Upper bound of $\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|$)**.** *Suppose that Assumption 3.1 and 3.2 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point. Set $M = 3\rho m$ as in Theorem 4.1. For all the iterations of Algorithm 4, it holds that*

$$\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| \le \|\boldsymbol{z}_0 - \boldsymbol{z}^*\| + \frac{5\bar{\eta}\rho}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2. \tag{23}$$

Proof. Let $r_t := \boldsymbol{F}(\boldsymbol{z}^*) - \boldsymbol{A}_t(\boldsymbol{z}^*)$ and define the operator $\tilde{\boldsymbol{A}}_t$ as $\tilde{\boldsymbol{A}}_t(\boldsymbol{z}) = \boldsymbol{A}_t(\boldsymbol{z}) + r_t$. From the definition of $\tilde{\boldsymbol{A}}_t$ we know that all any $\eta > 0$ and $\boldsymbol{z} \in \mathbb{R}^d$ we have that

$$(\boldsymbol{I}_d + \eta\tilde{\boldsymbol{A}}_t)^{-1}(\boldsymbol{z} + \eta r_t) = (\boldsymbol{I}_d + \eta\boldsymbol{A}_t)^{-1}(\boldsymbol{z}) \tag{24}$$

Now we upper bound $\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\|$ as follows.

$$\begin{aligned}
&\|\boldsymbol{z}_{t+1/2}^0 - \boldsymbol{z}_t\| \\
&= \|(\boldsymbol{I}_d + \bar{\eta}\boldsymbol{A}_t)^{-1}(\boldsymbol{z}_t) - \boldsymbol{z}_t\| \\
&= \|(\boldsymbol{I}_d + \bar{\eta}\tilde{\boldsymbol{A}}_t)^{-1}(\boldsymbol{z}_t + \bar{\eta}r_t) - \boldsymbol{z}_t\| \\
&\le \|(\boldsymbol{I}_d + \bar{\eta}\tilde{\boldsymbol{A}}_t)^{-1}(\boldsymbol{z}_t) - \boldsymbol{z}_t\| + \|(\boldsymbol{I}_d + \bar{\eta}\tilde{\boldsymbol{A}}_t)^{-1}(\boldsymbol{z}_t) - (\boldsymbol{I}_d + \bar{\eta}\tilde{\boldsymbol{A}}_t)^{-1}(\boldsymbol{z}_t + \bar{\eta}r_t)\| \\
&\le \|\boldsymbol{z}_t - \boldsymbol{z}^*\| + \bar{\eta}\|r_t\|,
\end{aligned} \tag{25}$$

where in the last step we use Lemma H.4 to upper bound the first term and use the non-expansiveness of resolvent (see *i.e.* (Rockafellar, 1976)) to upper bound the second term.

We continue to upper bound $\|r_t\|$. Recall the definition of $\boldsymbol{A}_t$ in (19), we know that

$$\begin{aligned} r_t &= \boldsymbol{F}(\boldsymbol{z}^*) - \boldsymbol{F}(\boldsymbol{z}_t) - \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}^* - \boldsymbol{z}_t) \\ &= \boldsymbol{F}(\boldsymbol{z}^*) - \boldsymbol{F}(\boldsymbol{z}_t) - \nabla \boldsymbol{F}(\boldsymbol{z}_t)(\boldsymbol{z}^* - \boldsymbol{z}_t) + (\nabla \boldsymbol{F}(\boldsymbol{z}_t) - \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}^* - \boldsymbol{z}_t) \end{aligned}$$

Note that $\nabla \boldsymbol{F}$ is ρ-Lipschitz continuous. Taking norm on both sides of the above identity, we have

$$\|r_t\| \leq \frac{\rho}{2}\|\boldsymbol{z}^* - \boldsymbol{z}_t\|^2 + \rho\|\boldsymbol{z}_t - \boldsymbol{z}_{\pi(t)}\|\|\boldsymbol{z}^* - \boldsymbol{z}_t\|$$

Recalling (16) that we have $\|\boldsymbol{z}_t - \boldsymbol{z}^*\| \leq \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$ for all t, by the triangle inequality we also have $\|\boldsymbol{z}_t - \boldsymbol{z}_{\pi(t)}\| \leq 2\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$. Therefore, we have that $\|r_t\| \leq \frac{5}{2}\|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^2$. Finally, we plug into (25) to obtain the desired upper bound in (23).

□

Next, we give a uniform lower bound of $\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\|$.

Lemma H.6 (Lower bound of $\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\|$)**.** *Suppose that Assumption 3.1 and 3.2 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point and $\beta = \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$. Set $M = 3\rho m$ as in Theorem 4.1. If in all the iterations of Algorithm 5 the point $\boldsymbol{z}^0_{t+1/2}$ is not an ϵ-solution, it holds that*

$$\bar{\eta}\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\| \geq \xi_t, \tag{26}$$

where $\xi_t = \min\left\{2\beta, \frac{\bar{\eta}\epsilon}{6\beta(3\bar{\eta}\beta\rho+1)}\right\}$.

Proof. We show a contradiction if (26) does not hold. Firstly, if $\boldsymbol{z}^0_{t+1/2} = (\boldsymbol{x}^0_{t+1/2}, \boldsymbol{y}^0_{t+1/2})$ is not an ϵ-solution to the problem, then by Lemma A.1 we know that $\|\boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\|$ must be large due to

$$\epsilon \leq \mathrm{Gap}(\boldsymbol{x}^0_{t+1/2}, \boldsymbol{y}^0_{t+1/2}; 3\beta) \leq \max_{\boldsymbol{z}\in\mathbb{B}_{3\beta}(\boldsymbol{z}^*)} \langle \boldsymbol{F}(\boldsymbol{z}^0_{t+1/2}), \boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}\rangle \leq 6\beta\|\boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\|,$$

where the last step uses that $\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\| \leq 2\beta$ if (26) does not hold, $\|\boldsymbol{z}_t - \boldsymbol{z}^*\| \leq \beta$ and the triangle inequality. Therefore, we can conclude that

$$\|\boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\| \geq \frac{\epsilon}{6\beta}. \tag{27}$$

Secondly, from the update of the algorithm, we have that

$$\boldsymbol{z}_t - \boldsymbol{z}^0_{t+1/2} = \bar{\eta}(\boldsymbol{F}(\boldsymbol{z}_t) + \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t))$$

Then we further know that

$$\begin{aligned} &\boldsymbol{z}_t - \boldsymbol{z}^0_{t+1/2} - \boldsymbol{F}(\boldsymbol{z}^0_{t+1/2}) \\ &= \bar{\eta}(\boldsymbol{F}(\boldsymbol{z}_t) + \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)})(\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t) - \boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})) \\ &= \bar{\eta}(\boldsymbol{F}(\boldsymbol{z}_t) + \nabla \boldsymbol{F}(\boldsymbol{z}_t)(\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t) - \boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})) \\ &\quad + \bar{\eta}(\nabla \boldsymbol{F}(\boldsymbol{z}_t) - \nabla \boldsymbol{F}(\boldsymbol{z}_{\pi(t)}))(\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t). \end{aligned}$$

Note that $\nabla \boldsymbol{F}$ is ρ-Lipschitz continuous. Taking norm on both sides of the above identity, we have

$$\begin{aligned} &\|\boldsymbol{z}_t - \boldsymbol{z}^0_{t+1/2} - \boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\| \\ &\leq \frac{\bar{\eta}\rho}{2}\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\|^2 + \bar{\eta}\rho\|\boldsymbol{z}_t - \boldsymbol{z}_{\pi(t)}\|\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\| \\ &\leq 3\bar{\eta}\beta\rho\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\|. \end{aligned}$$

where the last step uses the triangle inequality, that $\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\| \leq 2\beta$ if (26) does not hold, and that $\|\boldsymbol{z}_t - \boldsymbol{z}^*\| \leq \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$ by (16). Then we can know that

$$\begin{aligned} \bar{\eta}\|\boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\| &\leq \|\boldsymbol{z}_t - \boldsymbol{z}^0_{t+1/2} - \bar{\eta}\boldsymbol{F}(\boldsymbol{z}^0_{t+1/2})\| + \|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\| \\ &\leq (3\bar{\eta}\beta\rho + 1)\|\boldsymbol{z}^0_{t+1/2} - \boldsymbol{z}_t\|. \end{aligned}$$

Recalling (27), we know that this would contradict the hypothesis that (26) does not hold. □

Lemma H.5 and Lemma H.6 tell us that the h_t defined in (22) is uniformly bounded for all t. Finally, we obtain the following theorem by combining Theorem H.1 and Theorem H.3.

Theorem H.2. *Suppose that Assumption 3.1 and 3.2 hold. Let $\boldsymbol{z}^* = (\boldsymbol{x}^*, \boldsymbol{y}^*)$ be a saddle point and $\beta = \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|$. Set $M \geq 3\rho m$. The sequence of iterates generated by Algorithm 4 is bounded $\boldsymbol{z}_t \in \mathbb{B}_\beta(\boldsymbol{z}^*),\ \boldsymbol{z}_{t+1/2} \in \mathbb{B}_{3\beta}(\boldsymbol{z}^*), \quad \forall t = 0, \cdots, T-1$, and satisfies the following ergodic convergence:*

$$\mathrm{Gap}(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T; 3\beta) \leq \frac{16\alpha M \|\boldsymbol{z}_0 - \boldsymbol{z}^*\|^3}{T^{3/2}}.$$

Let $M = 3\rho m$ and $\alpha = 2$. Algorithm 1 finds an ϵ-saddle point within $\mathcal{O}(m^{2/3}\epsilon^{-2/3})$ iterations.

If we call the sub-procedure (Algorithm 5) wit fixed $\eta_t^0 = \bar{\eta}$, every call of this sub-procedure makes at most $\mathcal{O}(\log\log(\mathrm{poly}(m, \beta, \rho, \bar{\eta}, 1/\epsilon))$ Newton steps.

The above theorem shows that the CRN sub-problem can be solved to guarantee the desired precision for target problem in $\mathcal{O}(\log\log(1/\epsilon))$ iterations, which tightens the $\mathcal{O}(\log(1/\epsilon))$ iteration complexity in (Bullins & Lai, 2022; Adil et al., 2022). Additionally, (Bullins & Lai, 2022; Adil et al., 2022) requires additionally assume $\sigma_{\min}(\nabla \boldsymbol{F}(\boldsymbol{z})) \geq \mu$ for some positive constant μ, which makes the problem similar to strongly-convex(-strongly-concave) problems, while our analysis does not require such an assumption.