Journal of Machine Learning Research 20 (2019) 1-47 Submitted 1/19; Revised 7/19; Published 8/19

Stochastic Variance-Reduced Cubic Regularization Methods

Dongruo Zhou DRZHOU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Pan Xu PANXU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Quanquan Gu QGU@CS.UCLA.EDU
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA

Editor: Zhihua Zhang

Abstract

We propose a stochastic variance-reduced cubic regularized Newton method (SVRC) for non-convex optimization. At the core of SVRC is a novel semi-stochastic gradient along with a semi-stochastic Hessian, which are specifically designed for cubic regularization method. For a nonconvex function with n component functions, we show that our algorithm is guaranteed to converge to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum within $\widetilde{O}(n^{4/5}/\epsilon^{3/2})$[1] second-order oracle calls, which outperforms the state-of-the-art cubic regularization algorithms including subsampled cubic regularization. To further reduce the sample complexity of Hessian matrix computation in cubic regularization based methods, we also propose a sample efficient stochastic variance-reduced cubic regularization (Lite-SVRC) algorithm for finding the local minimum more efficiently. Lite-SVRC converges to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum within $\widetilde{O}(n + n^{2/3}/\epsilon^{3/2})$ Hessian sample complexity, which is faster than all existing cubic regularization based methods. Numerical experiments with different nonconvex optimization problems conducted on real datasets validate our theoretical results for both SVRC and Lite-SVRC.

Keywords: Cubic Regularization, Nonconvex Optimization, Variance Reduction, Hessian Sample Complexity, Local Minimum

1. Introduction

We study the following unconstrained finite-sum nonconvex optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}), \tag{1}$$

[1] Here $\widetilde{O}$ hides poly-logarithmic factors.

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a general nonconvex function. Such nonconvex optimization problems are ubiquitous in machine learning, including training deep neural network (LeCun et al., 2015), robust linear regression (Yu and Yao, 2017) and nonconvex regularized logistic regression (Reddi et al., 2016b). In principle, finding the global minimum of (1) is generally a NP-hard problem (Hillar and Lim, 2013) due to the lack of convexity.

Instead of finding the global minimum, various algorithms have been developed in the literature (Nesterov and Polyak, 2006; Cartis et al., 2011a; Carmon and Duchi, 2016; Agarwal et al., 2017; Xu et al., 2018b; Allen-Zhu and Li, 2018) to find an approximate local minimum of (1). In particular, a point $\mathbf{x}$ is said to be an (ϵ_g, ϵ_H)-approximate local minimum of F if

$$\|\nabla F(\mathbf{x})\|_2 \leq \epsilon_g, \quad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \geq -\epsilon_H, \tag{2}$$

where $\epsilon_g, \epsilon_H > 0$ are predefined precision parameters. It has been shown that such approximate local minima can be as good as global minima in some problems. For instance, Ge et al. (2016) proved that any local minimum is actually a global minimum in matrix completion problems. Therefore, to develop an algorithm to find an approximate local minimum is of great interest both in theory and practice.

A very important and popular method to find the approximate local minimum is cubic-regularized (CR) Newton method, which was originally introduced by Nesterov and Polyak (2006). Generally speaking, in the k-th iteration, CR solves a sub-problem which minimizes a cubic-regularized second-order Taylor expansion at current iterate $\mathbf{x}_k$. The update rule can be written as follows:

$$\mathbf{h}_k = \operatorname*{argmin}_{\mathbf{h} \in \mathbb{R}^d} \langle \nabla F(\mathbf{x}_k), \mathbf{h} \rangle + \frac{1}{2}\langle \nabla^2 F(\mathbf{x}_k)\mathbf{h}, \mathbf{h} \rangle + \frac{M}{6}\|\mathbf{h}\|_2^3, \tag{3}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k, \tag{4}$$

where $M > 0$ is a penalty parameter used in CR. Nesterov and Polyak (2006) proved that to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum of a nonconvex function F, CR requires at most $O(\epsilon^{-3/2})$ iterations. However, a major drawback for CR is that it needs to sample n individual gradients $\nabla f_i(\mathbf{x}_k)$ and Hessian matrices $\nabla^2 f_i(\mathbf{x}_k)$ in (3) at each iteration, which leads to a total $O(n\epsilon^{-3/2})$ Hessian sample complexity, i.e., number of queries to the stochastic Hessian $\nabla^2 f_i(\mathbf{x})$ for some i and $\mathbf{x}$. Such computational cost will be extremely expensive when n is large as in many large scale machine learning problems.

To overcome the computational burden of CR based methods, some recent studies have proposed to use sub-sampled Hessian instead of the full Hessian (Kohler and Lucchi, 2017; Xu et al., 2017a) to reduce the Hessian complexity. In detail, Kohler and Lucchi (2017) proposed a sub-sampled cubic-regularized Newton method (SCR), which uses a subsampled Hessian instead of full Hessian to reduce the per iteration sample complexity of Hessian evaluations. Xu et al. (2017a) proposed a refined convergence analysis of SCR, as well as a subsampled Trust Region algorithm (Conn et al., 2000). Nevertheless, SCR bears a much slower convergence rate than the original CR method, and the total Hessian sample complexity for SCR to achieve an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum is $\widetilde{O}(\epsilon^{-5/2})$. This suggests that the computational cost of SCR could be even worse than CR when $\epsilon \lesssim n^{-1}$.

In this paper, we propose a novel cubic regularization algorithm named Stochastic Variance-Reduced Cubic regularization (SVRC), which incorporates the variance reduction techniques (Johnson and Zhang, 2013; Xiao and Zhang, 2014; Allen-Zhu and Hazan, 2016; Reddi et al., 2016a) into the cubic-regularized Newton method. The key component in our algorithm is a novel semi-stochastic gradient, together with a semi-stochastic Hessian, that are specifically designed for cubic regularization. Furthermore, we prove that, for L_2-Hessian Lipschitz functions, to attain an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, our proposed algorithm requires $O(n+n^{4/5}/\epsilon^{3/2})$ Second-order Oracle (SO) calls and $O(1/\epsilon^{3/2})$ Cubic Subproblem Oracle (CSO) calls. Here an SO oracle represents an evaluation of triple $(f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x}))$, and a CSO oracle denotes an evaluation of the exact solution (or inexact solution) of the cubic subproblem (3). Compared with the original cubic regularization algorithm (Nesterov and Polyak, 2006), which requires $O(n/\epsilon^{3/2})$ SO calls and $O(1/\epsilon^{3/2})$ CSO calls, our proposed SVRC algorithm reduces the SO calls by a factor of $\Omega(n^{1/5})$.

The second-order oracle complexity is dominated by the maximum number of queries to one of the elements in the triplet $(f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x}))$, and therefore is not always desirable in reflecting the computational complexity of multifarious applications. Therefore, we need to focus more on the Hessian sample complexity of cubic regularization methods for relatively high dimensional problems. Based on the SVRC algorithm, in order to further reduce the Hessian sample complexity, we also develop a sample efficient stochastic variance-reduced cubic-regularized Newton method called Lite-SVRC, which significantly reduces the sample complexity of Hessian matrix evaluations in stochastic CR methods. Under mild conditions, we prove that Lite-SVRC achieves a lower Hessian sample complexity than existing cubic regularization based methods. We prove that Lite-SVRC converges to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum of a nonconvex function within $\widetilde{O}(n+n^{2/3}\epsilon^{-3/2})$ Hessian sample complexity.

We summarize the major contributions of this paper as follows:

- We present a novel cubic regularization method (SVRC) with improved oracle complexity. To the best of our knowledge, this is the first algorithm that outperforms cubic regularization without any loss in convergence rate. In sharp contrast, existing subsampled cubic regularization methods (Kohler and Lucchi, 2017; Xu et al., 2017a) suffer from worse convergence rates than cubic regularization.

- We also extend SVRC to the case with inexact solution to the cubic regularization subproblem. Similar to previous work (Cartis et al., 2011a; Xu et al., 2017a), we layout a set of sufficient conditions, under which the output of the inexact algorithm is still guaranteed to have the same convergence rate and oracle complexity as the exact algorithm. This further sheds light on the practical implementation of our algorithm.

- As far as we know, our work is the first to rigorously demonstrate the advantage of variance reduction for second-order optimization algorithms. Although there exist a few studies (Lucchi et al., 2015; Moritz et al., 2016; Rodomanov and Kropotov, 2016) using variance reduction to accelerate Newton method, none of them can deliver faster rates of convergence than standard Newton method.

- We also propose a lite version of SVRC, namely, the Lite-SVRC algorithm, which only requires a constant batch size of Hessian evaluations at each iteration. The proposed Lite-SVRC further improves the Hessian sample complexity of SVRC and outperforms the state-of-the-art result by achieving $\widetilde{O}(n+n^{2/3}\epsilon^{-3/2})$ Hessian sample complexity.

- We conduct extensive numerical experiments with different types of nonconvex optimization problems on various real datasets to validate our theoretical results for both SVRC and Lite-SVRC.

When the short version of this paper was submitted to ICML, there was a concurrent work by Wang et al. (2018a), which applies the idea of stochastic variance reduction to cubic regularization as well. Their algorithms have a worse Hessian sample complexity than Lite-SVRC. Since the short version of this paper was published in ICML, there have been two followup works by Wang et al. (2018b) and Zhang et al. (2018), which both proposed similar algorithms to our Lite-SVRC algorithm, and achieved the same Hessian sample complexity. However, Wang et al. (2018b) and Zhang et al. (2018)'s results rely on the adaptive choice of batch size for stochastic Hessian. Furthermore, Zhang et al. (2018)'s result relies on a stronger notion of Hessian Lipschitz condition. We will discuss the key difference between our Lite-SVRC algorithm and the algorithms in Wang et al. (2018a,b); Zhang et al. (2018) in detail in Section 7.

Notation: We use $a(x) = O(b(x))$ if $a(x) \leq Cb(x)$, where C is a constant independent of any parameters in our algorithm. We use $\widetilde{O}(\cdot)$ to hide polynomial logarithm terms. We use $\|\mathbf{v}\|_2$ to denote the 2-norm of vector $\mathbf{v} \in \mathbb{R}^d$. For symmetric matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$, we use $\|\mathbf{H}\|_2$ and $\|\mathbf{H}\|_{S_r}$ to denote the spectral norm and Schatten r- norm of $\mathbf{H}$. We denote the smallest eigenvalue of $\mathbf{H}$ to be $\lambda_{\min}(\mathbf{H})$.

2. Related Work

Cubic Regularization and Trust-region Newton Method Traditional Newton method in convex setting has been widely studied in past decades (Bennett, 1916; Bertsekas, 1999). The most related work to ours is the nonconvex cubic regularized Newton method, which was originally proposed in Nesterov and Polyak (2006). Cartis et al. (2011a) presented an adaptive framework of cubic regularization, which uses an adaptive estimation of the local Lipschitz constant and approximate solution to the cubic subproblem. To connect cubic regularization with traditional trust region method (Conn et al., 2000; Cartis et al., 2009, 2012, 2013), Blanchet et al. (2016); Curtis et al. (2017); Martínez and Raydan (2017) showed that the trust-region Newton method can achieve the same iteration complexity as the cubic regularization method. To overcome the computational burden of gradient and Hessian matrix evaluations, Kohler and Lucchi (2017); Xu et al. (2017a,b) proposed to use subsampled gradient and Hessian in cubic regularization. On the other hand, in order to solve the cubic subproblem (3) more efficiently, Carmon and Duchi (2016) proposed to use gradient descent, while Agarwal et al. (2017) proposed a sophisticated algorithm based on approximate matrix inverse and approximate PCA. Tripuraneni et al. (2018) proposed a refined stochastic cubic regularization algorithm based on above subproblem solver. However, none of the aforementioned variants of cubic regularization outperforms the original cubic regularization method in terms of oracle complexity.

Finding Approximate Local Minima There is another line of work for finding approximate local minima which focuses on escaping from nondegenerated saddle points using the negative curvature. Ge et al. (2015); Jin et al. (2017) showed that simple (stochastic) gradient descent with an injected uniform noise over a small ball is able to converge to approximate local minima. Carmon et al. (2018); Royer and Wright (2018); Allen-Zhu (2018) showed that by calculating the negative curvature using Hessian information or Hessian vector product, one can find approximate local minima faster than first-order methods. Xu et al. (2018b); Allen-Zhu and Li (2018); Jin et al. (2018) further proved that gradient methods with additive noise are also able to find approximate local minima faster than the first-order methods. Yu et al. (2017) proposed the GOSE algorithm to save negative curvature computation and Yu et al. (2018) improved the gradient complexity by exploring the third-order smoothness of objective functions. Raginsky et al. (2017); Zhang et al. (2017); Xu et al. (2018a) proved that a family of algorithms based on discretizations of Langevin dynamics can find a neighborhood of the global minimum of nonconvex objective functions.

Variance Reduction Variance-reduced techniques play an important role in our proposed algorithm, which have been extensively studied for large-scale finite-sum optimization problems. Variance reduction was first proposed in convex finite-sum optimization (Roux et al., 2012; Johnson and Zhang, 2013; Xiao and Zhang, 2014; Defazio et al., 2014), which uses semi-stochastic gradient to reduce the variance of the stochastic gradient and improves the gradient complexity of both stochastic gradient descent (SGD) and gradient descent (GD). Representative algorithms include Stochastic Average Gradient (SAG) (Roux et al., 2012), Stochastic Variance Reduced Gradient (SVRG) (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014), to mention a few. Garber and Hazan (2015); Shalev-Shwartz (2016) studied non-convex finite-sum problems where each individual function may be non-convex, but their sum is still convex. Reddi et al. (2016a) and Allen-Zhu and Hazan (2016) extended SVRG to the general non-convex finite-sum optimization, and proved that SVRG is able to converge to a first-order stationary point with the same convergence rate as gradient descent, yet with an $\Omega(n^{1/3})$ improvement in gradient complexity. Recently Zhou et al. (2018b) and Fang et al. (2018) further improved the gradient complexity of SVRG type of algorithms to converge to a first-order stationary point in nonconvex optimization to an optimal rate. However, to the best of our knowledge, it is still an open problem whether variance reduction can improve the oracle complexity of second-order optimization algorithms.

The remainder of this paper is organized as follows: we present the stochastic variance-reduced cubic regularization (SVRC) algorithm in Section 3. We present our theoretical analysis of the proposed SVRC algorithm in Section 4 and discuss on SVRC with inexact cubic subproblem oracles in Section 5. In Section 6, we propose a modified algorithm, Lite-SVRC, to further reduce Hessian sample complexity and present its theoretical analysis in Section 7. We conduct thorough numerical experiments on different nonconvex optimization problems and on different real world datasets to validate our theory in Section 8. We conclude our work in Section 9.

3. Stochastic Variance-Reduced Cubic Regularization

In this section, we present a novel algorithm, which utilizes stochastic variance reduction techniques to improve cubic regularization method.

As is discussed in the introduction, to reduce the computation burden of gradient and Hessian matrix evaluations in the cubic regularization updates in (3), subsampled gradient and Hessian matrix have been used in subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b) and stochastic cubic regularization (Tripuraneni et al., 2018).

Algorithm 1 Stochastic Variance Reduction Cubic Regularization (**SVRC**)

```
Input: batch size parameters $b_g, b_h$, cubic penalty parameters $\{M_{s,t}\}$, epoch number $S$, epoch length $T$ and starting point $\mathbf{x}_0$.
Initialization $\widehat{\mathbf{x}}^1 = \mathbf{x}_0$
for $s = 1, \ldots, S$ do
    $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$
    $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widehat{\mathbf{x}}^s), \mathbf{H}^s = \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(\widehat{\mathbf{x}}^s)$
    for $t = 0, \ldots, T-1$ do
        Sample index set $I_g, I_h, |I_g| = b_g, |I_h| = b_h$;
        $\mathbf{v}_t^s = \frac{1}{b_g}\sum_{i_t \in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \big(\frac{1}{b_g}\sum_{i_t \in I_g}\nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\big)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)$
        $\mathbf{U}_t^s = \frac{1}{b_h}\sum_{j_t \in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s$
        $\mathbf{h}_t^s = \mathrm{argmin}\langle \mathbf{v}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s \mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3$,
        $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$
    end for
    $\widehat{\mathbf{x}}^{s+1} = \mathbf{x}_T^s$
end for
Output: $\mathbf{x}_{\mathrm{out}} = \mathbf{x}_t^s$, where $s, t$ are uniformly random chosen from $s \in [S]$ and $t \in [T]$.
```

Nevertheless, the stochastic gradient and Hessian matrix have large variances, which undermine the convergence performance. Inspired by SVRG (Johnson and Zhang, 2013), we propose to use a semi-stochastic version of gradient and Hessian matrix, which can control the variances automatically. Specifically, our algorithm has two loops. At the beginning of the s-th iteration of the outer loop, we denote $\widehat{\mathbf{x}}^s = \mathbf{x}_0^s$. We first calculate the full gradient $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s)$ and Hessian matrix $\mathbf{H}^s = \nabla^2 F(\widehat{\mathbf{x}}^s)$, which are stored for further references in the inner loop. At the t-th iteration of the inner loop, we calculate the following semi-stochastic gradient and Hessian matrix:

$$\mathbf{v}_t^s = \frac{1}{b_g}\sum_{i_t \in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \frac{1}{b_g}\sum_{i_t \in I_g}\big(\nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\big)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s), \tag{5}$$

$$\mathbf{U}_t^s = \frac{1}{b_h}\sum_{j_t \in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s, \tag{6}$$

where I_g and I_h are batch index sets, and the batch sizes $b_g = |I_g|, b_h = |I_h|$ will be decided later. In each inner iteration, we solve the following cubic regularization subproblem:

$$\begin{aligned} \mathbf{h}_t^s &= \mathrm{argmin}\, m_t^s(\mathbf{h}), \\ m_t^s(\mathbf{h}) &= \langle \mathbf{v}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s \mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3, \end{aligned} \tag{7}$$

where $\{M_{s,t}\}$ are cubic regularization parameters, which may depend on s and t. Then we perform the update $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$ in the t-th iteration of the inner loop. The proposed algorithm is displayed in Algorithm 1.

There are two notable features of our "estimator" of the full gradient and Hessian in each inner loop, compared with that used in SVRG (Johnson and Zhang, 2013). The first is that our gradient and Hessian estimators consist of mini-batches of stochastic gradient and Hessian. The second one is that we use second-order information when we construct the gradient estimator $\mathbf{v}_t^s$, while classical SVRG only uses first-order information to build it. Intuitively speaking, both features are used to make a more accurate estimation of the true gradient and Hessian with affordable oracle calls. Note that similar approximations of the gradient and Hessian matrix have been staged in recent work by Gower et al. (2018) and Wai et al. (2017), where they used this new kind of estimator for traditional SVRG in the convex setting, which radically differs from our setting.

4. Theoretical Analysis of SVRC

In this section, we prove the convergence rate of SVRC (Algorithm 1) to an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum. We first lay out the following Hessian Lipschitz assumption, which is necessary for our analysis and is widely used in the literature (Nesterov and Polyak, 2006; Xu et al., 2016; Kohler and Lucchi, 2017).

Assumption 1 (Hessian Lipschitz) *There exists a constant $L_2 > 0$, such that for all $\mathbf{x}, \mathbf{y}$ and $i \in [n]$*

$$\big\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\big\|_2 \leq L_2\|\mathbf{x} - \mathbf{y}\|_2.$$

The Hessian Lipschitz assumption plays a central role in controlling the changing speed of second order information. In fact, this is the only assumption we need to prove our theoretical results for SVRC. We then define the following optimal function gap between initial point $\mathbf{x}_0$ and the global minimum of F.

Definition 2 (Optimal Gap) *For function $F(\cdot)$ and the initial point $\mathbf{x}_0$, let Δ_F be*

$$\Delta_F = \inf\{\Delta \in \mathbb{R} : F(\mathbf{x}_0) - F^* \leq \Delta\},$$

where $F^* = \inf_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})$.

W.L.O.G., we assume $\Delta_F < +\infty$ throughout this paper. Before we present nonasymptotic convergence results of Algorithm 1, we define the following useful notation

$$\mu(\mathbf{x}) = \max\bigg\{\|\nabla F(\mathbf{x})\|_2^{3/2}, -\frac{\lambda_{\min}^3\big(\nabla^2 F(\mathbf{x})\big)}{L_2^{3/2}}\bigg\}. \tag{8}$$

A similar definition also appears in Nesterov and Polyak (2006) with a slightly different form, which is used to describe how much a point is similar to a true local minimum. In particular, according to the definition in (8), $\mu(\mathbf{x}) \leq \epsilon^{3/2}$ holds if and only if

$$\|\nabla F(\mathbf{x})\|_2 \leq \epsilon, \quad \lambda_{\min}\big(\nabla^2 F(\mathbf{x})\big) > -\sqrt{L_2\epsilon}. \tag{9}$$

Therefore, in order to find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum of the nonconvex function F, it suffices to find $\mathbf{x}$ which satisfies $\mu(\mathbf{x}) < \epsilon^{3/2}$. Next we formally define our oracles:

Definition 3 (Second-order Oracle) *Given an index i and a point $\mathbf{x}$, one second-order oracle (SO) call returns such a triple:*

$$[f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x})]. \tag{10}$$

Definition 4 (Cubic Subproblem Oracle) *Given a vector $\mathbf{g} \in \mathbb{R}^d$, a Hessian matrix $\mathbf{H}$ and a positive constant θ, one Cubic Subproblem Oracle (CSO) call returns $\mathbf{h}_{sol}$, where $\mathbf{h}_{sol}$ can be solved exactly as follows*

$$\mathbf{h}_{sol} = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \mathbf{g}, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h}\rangle + \frac{\theta}{6}\|\mathbf{h}\|_2^3.$$

Remark 5 *The second-order oracle is a special form of* Information Oracle *firstly introduced by Nemirovsky and Yudin (1983), which returns gradient, Hessian and all high order derivatives of the objective function $F(\mathbf{x})$. Here, our second-order oracle will only returns first and second order information at some point of single objective f_i instead of F. We argue that it is a reasonable adaption because in this paper we focus on finite-sum objective function. The Cubic Subproblem Oracle will return an exact or inexact solution of* (7)*, which plays an important role in both theory and practice.*

Now we are ready to give a general convergence result of Algorithm 1:

Theorem 6 *Under Assumption 1, suppose that the cubic regularization parameter $M_{s,t}$ of Algorithm 1 satisfies that $M_{s,t} = C_M L_2$, where L_2 is the Hessian Lipschitz parameter and $C_M \geq 100$ is a constant. The batch sizes b_g and b_h satisfy that*

$$b_g \geq 5T^4,\ b_h \geq 100T^2 \log d, \tag{11}$$

where $T \geq 2$ is the length of the inner loop of Algorithm 1 and d is the dimension of the problem. Then the output of Algorithm 1 satisfies

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{240 C_M^2 L_2^{1/2} \Delta_F}{ST}. \tag{12}$$

Remark 7 *According to* (8)*, to ensure that $\mathbf{x}_{out}$ is an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, we can set the right hand side of* (12) *to be less than $\epsilon^{3/2}$. This immediately implies that the total iteration complexity of Algorithm 1 is $ST = O(\Delta_F L_2^{1/2}\epsilon^{-3/2})$, which matches the iteration complexity of cubic regularization (Nesterov and Polyak, 2006).*

Remark 8 *Note that there is a $\log d$ term in the expression of the parameter, and it is only related to Hessian batch size b_h. The $\log d$ term comes from matrix concentration inequalities, which is believed to be unavoidable (Tropp et al., 2015). In other words, the batch size of Hessian matrix b_h has an inevitable relation to dimension d, unlike the batch size of gradient b_g.*

The result in Theorem 6 depends on a series of parameters. In the following corollary, we will show how to choose these parameters in practice to achieve a better oracle complexity.

Corollary 9 *Under Assumption 1, let the cubic regularization parameter $M_{s,t} = M = C_M L_2$, where $C_M \geq 100$ is a constant. Let the epoch length $T = n^{1/5}$, batch sizes $b_g = 5n^{4/5}$, $b_h = 100n^{2/5}\log d$, and the number of epochs $S = \max\{1, 240C_M^2 L_2^{1/2}\Delta_F n^{-1/5}\epsilon^{-3/2}\}$. Then Algorithm 1 will find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum $\mathbf{x}_{out}$ within*

$$O\bigg(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\bigg) \text{ SO calls} \tag{13}$$

and

$$O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg) \text{ CSO calls.} \tag{14}$$

Remark 10 *Corollary 9 states that we can reduce the SO calls by setting the batch size b_g, b_h related to n. In contrast, in order to achieve an $(\epsilon, \sqrt{L_2\epsilon})$ local minimum, original cubic regularization method in Nesterov and Polyak (2006) needs $O(n/\epsilon^{3/2})$ second-order oracle calls, which is by a factor of $n^{1/5}$ worse than ours. And subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b) requires $\widetilde{O}(n/\epsilon^{3/2} + 1/\epsilon^{5/2})$ SO calls, which is also worse than our algorithm.*

In Table 1, we summarize the comparison of our SVRC algorithm with the most related algorithms in terms of SO and CSO oracle complexities. It can be seen from Table 1 that our algorithm (SVRC) achieves the lowest (SO and CSO) oracle complexity compared with the original cubic regularization method (Nesterov and Polyak, 2006) which employs full gradient and Hessian evaluations and the subsampled cubic method (Kohler and Lucchi, 2017; Xu et al., 2017b). In particular, our algorithm reduces the SO oracle complexity of cubic regularization by a factor of $n^{1/5}$ for finding an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum.

Algorithm	SO calls	CSO calls	Gradient Lipschitz	Hessian Lipschitz
CR	$O\Big(\frac{n}{\epsilon^{3/2}}\Big)$	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	no	yes
SCR	$\widetilde{O}\Big(\frac{n}{\epsilon^{3/2}} + \frac{1}{\epsilon^{5/2}}\Big)^2$	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	yes	yes
SVRC (Algorithm 1)	$\widetilde{O}\Big(n + \frac{n^{4/5}}{\epsilon^{3/2}}\Big)$	$O\Big(\frac{1}{\epsilon^{3/2}}\Big)$	no	yes

Table 1: Comparisons between different methods to find $(\epsilon, \sqrt{L_2\epsilon})$-local minimum on the second-order oracle (SO) complexity and the cubic sub-problem oracle (CSO) complexity. The compared methods include (1) CR: Cubic regularization (Nesterov and Polyak, 2006) and (2) SCR: Subsampled cubic regularization (Kohler and Lucchi, 2017; Xu et al., 2017b).

[2] It is the refined rate proved by Xu et al. (2017b) for the subsampled cubic regularization algorithm proposed in Kohler and Lucchi (2017).

5. SVRC with Inexact Oracles

In practice, the exact solution to the cubic subproblem (7) cannot be obtained. Instead, one can only get an approximate solution by some inexact solver. Thus we replace the CSO oracle in (4) with the following inexact CSO oracle

$$\widetilde{\mathbf{h}}_{\mathrm{sol}} \approx \mathop{\mathrm{argmin}}_{\mathbf{h}\in\mathbb{R}^d} \langle \mathbf{g}, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{h}, \mathbf{H}\mathbf{h}\rangle + \frac{\theta}{6}\|\mathbf{h}\|_2^3.$$

To analyze the performance of SVRC with inexact cubic subproblem solver, we relax the exact solver $\mathbf{h}_t^s$ in Line 10 of Algorithm 1 with

$$\widetilde{\mathbf{h}}_t^s \approx \mathrm{argmin}\, m_t^s(\mathbf{h}). \tag{15}$$

The ultimate goal of this section is to prove that the theoretical results of SVRC still hold with inexact subproblem solvers. To this end, we present the following sufficient condition, under which inexact solution can ensure the same oracle complexity as the exact solution:

Condition 11 (Inexact Condition) *For each s, t and a given $\delta > 0$, $\widetilde{\mathbf{h}}_t^s$ satisfies δ-inexact condition if $\widetilde{\mathbf{h}}_t^s$ satisfies*

$$\begin{aligned} m_t^s(\widetilde{\mathbf{h}}_t^s) &\le -\frac{M_{s,t}}{12}\|\widetilde{\mathbf{h}}_t^s\|_2^3 + \delta, \\ \|\nabla m_t^s(\widetilde{\mathbf{h}}_t^s)\|_2 &\le M_{s,t}^{1/3}\delta^{2/3}, \\ \big|\|\widetilde{\mathbf{h}}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big| &\le M_{s,t}^{-1/3}\delta^{1/3}. \end{aligned}$$

Remark 12 *Similar inexact conditions have been studied in the literature of cubic regularization. For instance, Nesterov and Polyak (2006) presented a practical way to solve the cubic subproblem without termination condition. Cartis et al. (2011a); Kohler and Lucchi (2017) presented termination criteria for approximate solution to cubic subproblem, which is slightly different from Condition 11.*

Now we present the convergence result of SVRC with inexact CSO oracles:

Theorem 13 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$, which satisfies Condition 11. Under the same conditions of Theorem 6, the output of Algorithm 1 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \le \frac{240 C_M^2 L_2^{1/2}\Delta_F}{ST} + 480 C_M^2 L_2^{1/2}\delta. \tag{16}$$

Remark 14 *By the definition of $\mu(\mathbf{x})$, in order to attain an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, we require $\mathbb{E}[\mu(\mathbf{x}_{out})] \le \epsilon^{3/2}$ and thus $480 C_M^2 L_2^{1/2}\delta < \epsilon^{3/2}$, which implies that δ in Condition 11 should satisfy $\delta < (480 C_M^2 L_2^{1/2})^{-1}\epsilon^{3/2}$. Thus the total iteration complexity of Algorithm 1 with inexact oracle is still $O(\Delta_F L_2^{1/2}\epsilon^{-3/2})$.*

By the same choice of parameters, Algorithm 1 with inexact oracle can achieve a reduction in SO calls.

Corollary 15 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$, which satisfies Condition 11 with $\delta = (960C_M^2)^{-1}L_2^{-1/2}\epsilon^{3/2}$. Under Assumption 1, let the cubic regularization parameter $M_{s,t} = M = C_M L_2$, where $C_M \geq 100$ is a constant. Let the epoch length $T = n^{1/5}$, batch sizes $b_g = 5n^{4/5}$ and $b_h = 100n^{2/5}\log d$, and the number of epochs $S = \max\{1, 480C_M^2 L_2^{1/2}\Delta_F n^{-1/5}\epsilon^{-3/2}\}$. Then Algorithm 1 will find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum within*

$$O\bigg(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\bigg) \text{ SO calls} \tag{17}$$

and

$$O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg) \text{ CSO calls.} \tag{18}$$

Remark 16 *It is worth noting that even with the inexact CSO oracle satisfying Condition 11, the SO and CSO complexities of SVRC remain the same as that of SVRC with exact CSO oracle. Furthermore, this result always holds with any inexact cubic sub-problem solver.*

6. Lite-SVRC for Efficient Hessian Sample Complexity

As we discussed in the introduction section, when the problem dimension d is relatively high, we may want to focus more on the Hessian sample complexity of cubic regularization methods than the second-order oracle complexity. In this section, we present a new algorithm Lite-SVRC based on SVRC, which trades the second-order oracle complexity for a more affordable Hessian sample complexity. As is displayed in Algorithm 2, our Lite-SVRC algorithm has similar structure as Algorithm 1 with S epochs and T iterations within each epoch. At the t-th iteration of the s-th epoch, we also use a semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$ and Hessian $\mathbf{U}_t^s$ to replace the full gradient and full Hessian in CR subproblem (3) as follows

$$\widetilde{\mathbf{v}}_t^s = \frac{1}{B_{g;s,t}}\sum_{i_t\in I_g}\big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s, \tag{19}$$

$$\mathbf{U}_t^s = \frac{1}{B_h}\sum_{j_t\in I_h}\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{H}^s, \tag{20}$$

where $\widehat{\mathbf{x}}^s$ is the reference point at which $\mathbf{g}^s$ and $\mathbf{H}^s$ are computed, I_g and I_h are sampling index sets (with replacement), $B_{g;s,t}$ and B_h are sizes of I_g and I_h.

Compared with SVRC (Algorithm 1), Lite-SVRC uses a lite version of semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$. Note that the additional Hessian information in the semi-stochastic gradient in (5) actually increases the Hessian sample complexity. Therefore, with the goal of reducing the Hessian sample complexity, the standard semi-stochastic gradient (Johnson and Zhang, 2013; Xiao and Zhang, 2014) is used in this section. Note that similar semi-stochastic gradient and Hessian have been proposed in Johnson and Zhang (2013); Xiao and Zhang (2014) and Gower et al. (2018); Wai et al. (2017); Zhou et al. (2018a); Wang et al. (2018a,b); Zhang et al. (2018) respectively. In Algorithm 2, we choose fixed batch size of stochastic Hessian as $B_h = |I_h|$. However, the batch size of stochastic gradient is chosen adaptively at each iteration:

$$B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2, \tag{21}$$

where D_g is a constant only depending on n and d.

Algorithm 2 Sample efficient stochastic variance-reduced cubic regularization method (Lite-SVRC)

```
Input: batch size parameters $D_g, B_h$, cubic penalty parameter $\{M_{s,t}\}$, epoch number $S$, epoch length $T$ and starting point $\mathbf{x}_0$.
Initialization $\widehat{\mathbf{x}}^1 = \mathbf{x}_0$
for $s = 1, \ldots, S$ do
    $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$
    $\mathbf{g}^s = \nabla F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\widehat{\mathbf{x}}^s), \mathbf{H}^s = \nabla^2 F(\widehat{\mathbf{x}}^s) = \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(\widehat{\mathbf{x}}^s)$
    $\mathbf{h}_0^s = \mathrm{argmin}_{\mathbf{h}\in\mathbb{R}^d} m_0^s(\mathbf{h}) = \langle \mathbf{g}^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{H}^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,0}}{6}\|\mathbf{h}\|_2^3$
    $\mathbf{x}_1^s = \mathbf{x}_0^s + \mathbf{h}_0^s$
    for $t = 1, \ldots, T-1$ do
        $B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2, t > 0$
        Sample index set $I_g, I_h \subseteq [n]$, $|I_g| = B_{g;s,t}, |I_h| = B_h$
        $\widetilde{\mathbf{v}}_t^s = \frac{1}{B_{g;s,t}}\big(\sum_{i_t\in I_g} \nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big) + \mathbf{g}^s$
        $\mathbf{U}_t^s = \frac{1}{B_h}\big(\sum_{j_t\in I_h} \nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big) + \mathbf{H}^s$
        $\mathbf{h}_t^s = \mathrm{argmin}_{\mathbf{h}\in\mathbb{R}^d} m_t^s(\mathbf{h}) = \langle \widetilde{\mathbf{v}}_t^s, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}, \mathbf{h}\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}\|_2^3$
        $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$
    end for
    $\widehat{\mathbf{x}}^{s+1} = \mathbf{x}_T^s$
end for
Output: $\mathbf{x}_{\mathrm{out}} = \mathbf{x}_t^s$, where $s, t$ are uniformly random chosen from $s \in [S]$ and $t \in [T]$.
```

In addition, the major difference between our algorithm and the SVRC algorithms proposed in Wang et al. (2018a); Zhang et al. (2018); Wang et al. (2018b) is that our algorithm uses a constant Hessian minibatch size instead of an adaptive one in each iteration, and thus the parameter tuning of our algorithm is much easier. In sharp contrast, the minibatch sizes of the stochastic Hessian in the algorithm proposed by Wang et al. (2018a); Zhang et al. (2018); Wang et al. (2018b) are dependent on both accuracy parameter ϵ and the current update $\mathbf{h}_t^s$, which make the update an implicit one and it is hard to tune such hyperparameters in practice.

7. Theoretical Analysis of Lite-SVRC

In this section, we present our theoretical results on the Hessian sample complexity of Lite-SVRC (Algorithm 2). Different from the analysis of SVRC in Section 4 which only requires the Hessian Lipschitz condition (Assumption 1), we will need additionally the following smoothness assumption for the analysis of Lite-SVRC:

Assumption 17 (Gradient Lipschitz) *There exists a constant $L_1 > 0$, such that for all $\mathbf{x}, \mathbf{y}$ and $i \in \{1, ..., n\}$*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L_1\|\mathbf{x} - \mathbf{y}\|_2.$$

Assumptions 1 and 17 are mild and widely used in the line of research for finding approximate global minima (Carmon and Duchi, 2016; Carmon et al., 2018; Agarwal et al., 2017; Wang et al., 2018a; Yu et al., 2018).

Recall the definition in (8), we need to upper bound $\mu(\mathbf{x}_{\text{out}})$ in order to find the approximate local minimum. The following theorem spells out the upper bound of $\mu(\mathbf{x}_{\text{out}})$.

Theorem 18 *Under Assumptions 1 and 17, suppose that* $n > 10, M_{s,t} = C_M L_2$, $D_g \geq C_1 L_1^2/L_2^2 \cdot n^{4/3}/C_M$ *and* $B_h > 144(C_1 C_h)^{2/3} n^{2/3}/C_M^2$*, where* $C_h = 1200(\log d)$ *and* C_M, C_1 *are absolute constants. Then the output* $\mathbf{x}_{out}$ *of Algorithm 2 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{216 C_M^2 L_2^{1/2} \Delta_F}{ST}. \tag{22}$$

Remark 19 *Theorem 18 suggests that with a fixed number of inner loops* T*, if we run Algorithm 2 for sufficiently large* S *epochs, then we have a point sequence* $\mathbf{x}_i$ *where* $\mathbb{E}[\mu(\mathbf{x}_i)] \to 0$*. That being said,* $\mathbf{x}_i$ *will converge to a local minimum, which is consistent with the convergence analysis in existing related work (Nesterov and Polyak, 2006; Kohler and Lucchi, 2017; Wang et al., 2018a).*

Now we provide a specific choice of parameters used in Theorem 18 to derive the total Hessian sample complexity of Algorithm 2.

Corollary 20 *Under the same assumptions as in Theorem 18, let batch size parameters satisfy* $D_g = 4L_1^2/L_2^2 \cdot n^{4/3}$ *and* $B_h = \log d \cdot (C_h \cdot n)^{2/3}$*. Set the inner loop parameter* $T = n^{1/3}$ *and cubic regularization parameter* $M_{s,t} = C_M L_2$*, where* C_M *is an absolute constant. Set the epoch number* $S = O(\max\{L_2^{1/2}\Delta_F/(\epsilon^{3/2} n^{1/3}), 1\})$*. Then the output* $\mathbf{x}_{out}$ *from Algorithm 2 is an* $(\epsilon, \sqrt{L_2 \epsilon})$*-approximate local minimum after*

$$\widetilde{O}\bigg(n + \frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}} \cdot n^{2/3}\bigg) \textit{ stochastic Hessian evaluations.} \tag{23}$$

Moreover, the total number of CSO calls of Algorithm 2 is

$$O\bigg(\frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}}\bigg).$$

Remark 21 *Note that the CSO oracle complexity of Lite-SVRC is the same as SVRC. In what follows, we present a comprehensive comparison on Hessian sample complexity between our Lite-SVRC and other related algorithms in Table 2. The algorithm proposed in Wang et al. (2018a) has two versions: sample with replacement and sample without replacement. For the completeness, we present both versions in Wang et al. (2018a). From Table 2 we can see that Lite-SVRC strictly outperforms CR by a factor of* $n^{1/3}$ *and outperforms SVRC by a factor of* $n^{2/15}$ *in terms of Hessian sample complexity. Lite-SVRC also outperforms SCR when* $\epsilon = O(n^{-2/3})$*, which suggests that the variance reduction scheme makes Lite-SVRC perform better in the high accuracy regime. More importantly, our proposed Lite-SVRC does not rely on the assumption that the function* F *is Lipschitz continuous, which is required by* *the algorithm proposed in Wang et al. (2018a). In terms of Hessian sample complexity, our algorithm directly improves that of Wang et al. (2018a) by a factor of $n^{2/33}$. The Hessian sample complexity of Lite-SVRC is the same as that of the algorithms recently proposed in Wang et al. (2018b) and Zhang et al. (2018). Nevertheless, Lite-SVRC uses a constant Hessian sample batch size in contrast to the adaptive batch size as used in Wang et al. (2018b); Zhang et al. (2018), which makes the use of Lite-SVRC algorithm much simpler and more practical.*

Algorithm	**Per-iteration**	**Total**	**Function Lipschitz**	**Gradient Lipschitz**	**Hessian Lipschitz**
CR	$O(n)$	$O\Big(\frac{n}{\epsilon^{3/2}}\Big)$	No	No	Yes
SCR	$\widetilde{O}\Big(\frac{1}{\epsilon}\Big)$	$\widetilde{O}\Big(\frac{1}{\epsilon^{5/2}}\Big)$	No[3]	Yes	Yes
SVRC$_{\text{with}}$[4]	$\widetilde{O}\Big(\frac{\lVert\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\rVert_2^2}{\lVert\mathbf{h}_t^s\rVert_2^2}\Big)$	$\widetilde{O}\Big(n+\frac{n^{3/4}}{\epsilon^{3/2}}\Big)$	Yes	Yes	Yes
SVRC$_{\text{without}}$[4]	$\widetilde{O}\Big(\Big(\frac{1}{n}+\frac{\lVert\mathbf{h}_t^s\rVert_2^2}{\lVert\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\rVert_2^2}\Big)^{-1}\Big)$	$\widetilde{O}\Big(n+\frac{n^{8/11}}{\epsilon^{3/2}}\Big)$	Yes	Yes	Yes
SVRC$_{\text{Wang}}$	$\widetilde{O}\Big(\frac{\lVert\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\rVert_2^2}{\max\{\epsilon,\lVert\mathbf{h}_t^s\rVert_2^2\}}\Big)$	$\widetilde{O}\Big(n+\frac{n^{2/3}}{\epsilon^{3/2}}\Big)$	No	Yes	Yes
SVRC$_{\text{Zhang}}$	$\widetilde{O}\Big(\frac{\lVert\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\rVert_2^2}{\epsilon}\Big)$	$\widetilde{O}\Big(n+\frac{n^{2/3}}{\epsilon^{3/2}}\Big)$	No	Yes	Yes
SVRC[5] (Algorithm 1)	$\widetilde{O}(n^{4/5})$	$\widetilde{O}\Big(n+\frac{n^{4/5}}{\epsilon^{3/2}}\Big)$	No	No	Yes
Lite-SVRC (Algorithm 2)	$\widetilde{O}(n^{2/3})$	$\widetilde{O}\Big(n+\frac{n^{2/3}}{\epsilon^{3/2}}\Big)$	No	Yes	Yes

Table 2: Comparisons of per-iteration and total sample complexities of Hessian evaluations for different algorithms: CR (Nesterov and Polyak, 2006), SCR (Kohler and Lucchi, 2017; Xu et al., 2017a), SVRC$_{\text{with}}$ (Wang et al., 2018a), SVRC$_{\text{without}}$ (Wang et al., 2018a), SVRC$_{\text{Wang}}$ (Wang et al., 2018b), SVRC$_{\text{Zhang}}$ (Zhang et al., 2018), SVRC (Algorithm 1) and Lite-SVRC (Algorithm 2). Similar to Table 1, the CSO oracle complexities of all the methods being compared are the same, i.e. $O(1/\epsilon^{3/2})$. Therefore, we omit it for simplicity.

Recall the inexact cubic subproblem solver defined in Section 5. The same inexact CSO oracles can also be used in Algorithm 2. In what follows, we present the convergence result of Lite-SVRC with inexact CSO oracles.

[3] Although the refined SCR in Xu et al. (2017b) does not need function Lipschitz, the original SCR in Kohler and Lucchi (2017) needs it.

[4] In Wang et al. (2018a), both algorithms need to calculate $\lambda_{\min}(\nabla^2 F(\mathbf{x}_t^s))$ at each iteration to decide whether the algorithm should continue, which adds additional $O(n)$ Hessian sample complexity. We choose not to include this into the results in the table.

[5] For SVRC (Algorithm 1), we present its second-order oracle calls derived in Section 4 as the Hessian sample complexity.

Theorem 22 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$ satisfying Condition 11. Then under the same conditions of Theorem 18, the output of Algorithm 2 satisfies*

$$\mathbb{E}[\mu(\mathbf{x}_{out})] \leq \frac{216 C_M^2 L_2^{1/2} \Delta_F}{ST} + 432 C_M^2 L_2^{1/2} \delta. \tag{24}$$

In addition, Algorithm 2 with inexact oracle can also reduce the Hessian sample complexity, which is summarized in the following corollary.

Corollary 23 *Suppose that for each s, t, $\widetilde{\mathbf{h}}_t^s$ is an inexact solver of cubic subproblem $m_t^s(\mathbf{h})$ satisfying Condition 11 with $\delta = (864 C_M^2 L_2^{1/2})^{-1} \epsilon^{3/2}$. Then with the same choice of parameters in Corollary 20, Algorithm 2 will find an $(\epsilon, \sqrt{L_2 \epsilon})$-approximate local minimum within*

$$\widetilde{O}\bigg(n + \frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}} \cdot n^{2/3}\bigg) \text{ stochastic Hessian evaluations,}$$

and

$$O\bigg(\frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}}\bigg) \text{ CSO calls.}$$

8. Experiments

In this section, we conduct experiments on real world datasets to support our theoretical analysis of the proposed SVRC and Lite-SVRC algorithms. We investigate two nonconvex problems on three different datasets, *a9a*, *ijcnn1* and *covtype*, which are all common datasets used in machine learning and the sizes are summarized in Table 3.

8.1. Baseline Algorithms

Dataset	sample size n	dimension d
a9a	32,561	123
covtype	581,012	54
ijcnn1	35,000	22

Table 3: Datasets used in experiments.

To validate the superior performance of the proposed SVRC (Algorithm 1) in terms of second-order oracles, we compare it with the following baseline algorithms: (1) trust-region Newton methods (TR) (Conn et al., 2000); (2) Adaptive Cubic regularization (Cartis et al., 2011a,b); (3) Subsampled Cubic regularization (Kohler and Lucchi, 2017); (4) Gradient Cubic regularization (Carmon and Duchi, 2016) and (5) Stochastic Cubic regularization (Tripuraneni et al., 2018). To demonstrate the improvement of Lite-SVRC (Algorithm 2) on Hessian sample complexity, we further conduct experiments to compare Lite-SVRC with all the baselines above including SVRC. In addition, we also compare Lite-SVRC with (6) SVRC-without (Wang et al., 2018a), which focuses on reducing the Hessian sample complexity as well. In addition, there are two versions of SVRC in Wang et al. (2018a), but the one based on sampling without replacement performs better in both theory and experiments. We therefore only compare with this one. Note that the SVRC algorithms in Wang et al. (2018b); Zhang et al. (2018) are essentially the same as our Lite-SVRC algorithm, except in the choice of batch size for stochastic Hessian. Thus we do not compare our Lite-SVRC with these algorithms (Wang et al., 2018b; Zhang et al., 2018).

8.2. Implementation Details

For Subsampled Cubic and SVRC-without, the sample size B_k is dependent on $\|\mathbf{h}_k\|_2$ (Kohler and Lucchi, 2017) and B_h is dependent on $\|\mathbf{h}_t^s\|_2$ (Wang et al., 2018a), which make these two algorithms implicit algorithms. To address this issue, we follow the suggestion in Kohler and Lucchi (2017); Wang et al. (2018a) and use $\|\mathbf{h}_{k-1}\|_2$ and $\|\mathbf{h}_{t-1}^s\|_2$ instead of $\|\mathbf{h}_k\|_2$ and $\|\mathbf{h}_t^s\|_2$. Furthermore, we choose the penalty parameter $M_{s,t}$ for SVRC, SVRC-without and Lite-SVRC as constants which are suggested by the original papers of these algorithms. Finally, to solve the CR sub-problem in each iteration, we choose to solve the sub-problem approximately in the Krylov subspace spanned by Hessian related vectors, as used by Kohler and Lucchi (2017).

8.3. Nonconvex Optimization Problems

In this subsection, we formulate the nonconvex optimization problems that will be studied in our experiments. In particular, we choose two nonconvex regression problem as our objectives with the following nonconvex regularizer

$$g(\lambda, \gamma, \mathbf{x}) = \lambda \cdot \sum_{i=1}^{d} \frac{(\gamma x_i)^2}{1 + (\gamma x_i)^2}, \tag{25}$$

where λ, γ are the control parameters and x_i is the i-th coordinate of $\mathbf{x}$. λ and γ are set differently for each dataset. This regularizer has been widely used in nonconvex regression problem, which can be regarded as a special example of robust nonlinear regression (Reddi et al., 2016b; Kohler and Lucchi, 2017; Wang et al., 2018a).

8.3.1. Logistic Regression with Nonconvex Regularizer

The first problem is a binary logistic regression problem with a nonconvex regularizer g (Reddi et al., 2016b). Given training data $\mathbf{x}_i \in \mathbb{R}^d$ and label $y_i \in \{0, 1\}$, $1 \le i \le n$, our goal is to solve the following optimization problem:

$$\min_{\mathbf{s} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \big[y_i \cdot \log \phi(\mathbf{s}^\top \mathbf{x}_i) + (1 - y_i) \cdot \log[1 - \phi(\mathbf{s}^\top \mathbf{x}_i)]\big] + g(\lambda, \gamma, \mathbf{s}), \tag{26}$$

where $\phi(x) = 1/(1 + \exp(-x))$ is the sigmoid function and g is defined in (25).

8.3.2. Nonlinear Least Square with Nonconvex Regularizer

Another problem is the nonlinear least square problem with a nonconvex regularizer $g(\lambda, \gamma, \mathbf{x})$ defined in (25). The nonlinear least square problem is also studied in Xu et al. (2017b).

Given training data $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{0,1\}$, $1 \leq i \leq n$, our goal is to minimize the following problem

$$\min_{\mathbf{s}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\big[y_i - \phi(\mathbf{s}^\top \mathbf{x}_i)\big]^2 + g(\lambda, \gamma, \mathbf{s}). \tag{27}$$

Here $\phi(x) = 1/(1+\exp(-x))$ is again the sigmoid function and g is defined in (25).

8.4. Experimental Results for SVRC

In this subsection, we present the experimental results for SVRC compared with baseline algorithms (1)-(5) listed in Section 8.1. Here, we fix $\lambda = 10$ and $\gamma = 1$ of the nonconvex regularizer g in (25) for both the logistic regression and the nonlinear least square problems.

Calculation for SO calls: For *Subsampled Cubic*, each loop takes $(B_g + B_h)$ SO calls, where B_g and B_h are the subsampling sizes of gradient and Hessian. For *Stochastic Cubic*, each loop costs $(n_g + n_h)$ SO calls, where n_g and n_h denote the subsampling sizes of gradient and Hessian-vector operator. *Gradient Cubic*, *Adaptive Cubic* and *TR* cost n SO calls in each loop. We define the amount of epochs to be the amount of SO calls divided by n.

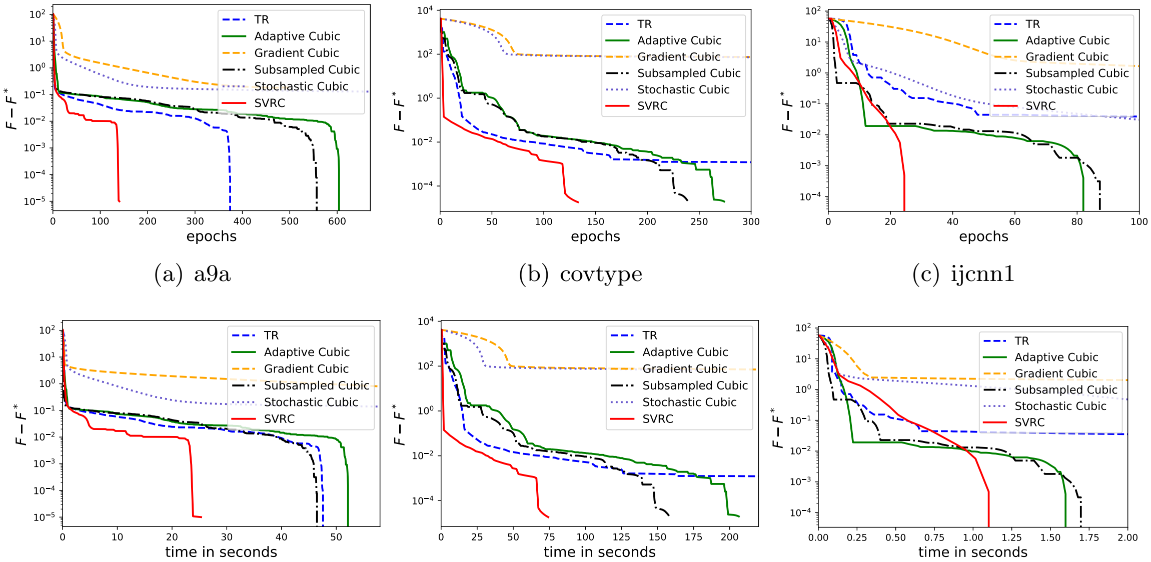
Parameters: For each algorithm and each dataset, we choose different b_g, b_h, T for the best performance. Meanwhile, we choose the cubic regularization parameter as $M_{s,t} = \alpha/(1+\beta)^{(s+t/T)}$, α, $\beta > 0$ for each iteration. When $\beta = 0$, it has been proved to enjoy good convergence performance. This choice of parameter is similar to the choice of penalty parameter in *Subsampled Cubic* and *Adaptive Cubic*, which sometimes makes some algorithms behave better in our experiments.

Subproblem Solver: With regard to the cubic subproblem solver for solving (7), we choose the Lanczos-type method used in Cartis et al. (2011a), which finds the global minimizer $\mathbf{h}_t^s$ of $m_t^s(\mathbf{h})$ in a Krylov subspace $\mathcal{K}_l = \mathrm{span}\{\mathbf{v}_t^s, \mathbf{U}_t^s\mathbf{v}_t^s, (\mathbf{U}_t^s)^2\mathbf{v}_t^s, \ldots, (\mathbf{U}_t^s)^{l-1}\mathbf{v}_t^s\}$, where $l \ll d$ is the dimension of $\mathcal{K}_l$ and can be selected manually or adaptively (Cartis et al., 2011a; Kohler and Lucchi, 2017). The computational complexity of Lanczos-type method consists of two parts according to Carmon and Duchi (2018). First, $(l-1)$ matrix-vector products are performed to calculate the basis of $\mathcal{K}_l$, whose computational complexity is $O(d^2 l)$. Second, the minimizer of $m_t^s(\mathbf{h})$ is computed in subspace $\mathcal{K}_l$, whose computational complexity is $O(l \log l)$. Thus, the total computational complexity of Lanczos-type method is $O(d^2 l)$.

At each iteration, SVRC needs to compute the semi-stochastic gradient $\mathbf{v}_t^s$ and Hessian $\mathbf{U}_t^s$, which costs $O(db_g + d^2 b_h)$ computational complexity for both nonconvex regularized logistic regression and nonlinear least square problems, where b_g and b_h are the mini-batch sizes of stochastic gradient and Hessian respectively. Putting these pieces together, the per-iteration complexity of SVRC is $O(db_g + d^2 b_h + d^2 l)$, and the total computational complexity of SVRC is $O(ST(db_g + d^2 b_h + d^3))$, where S is the number of epochs and T is the length of epoch.

For the binary logistic regression problem in (26), the parameters of $M_{s,t} = \alpha/(1+\beta)^{(s+t/T)}, \alpha, \beta > 0$ are set as follows: $\alpha = 0.05, \beta = 0$ for *a9a* and *ijcnn1* datasets and $\alpha = 5e3, \beta = 0.15$ for *covtype*. The experimental results are shown in Figure 1. For the non-linear least squares problem in (27), we set $\alpha = 0.05, 1e8, 0.003$ and $\beta = 0, 1, 0.5$ for *a9a*, *covtype* and *ijcnn1* datasets respectively. The experimental results are shown in Figure

2. From both Figures 1 and 2, we can see that SVRC outperforms all the other baseline algorithms on all the datasets. The only exception happens in the non-linear least square problem on the *covtype* dataset, where our algorithm behaves a little worse than Adaptive Cubic at the high accuracy regime in terms of epoch counts. However, under this setting, our algorithm still outperforms the other baselines in terms of the CPU time.



(a) a9a

(b) covtype

(c) ijcnn1

(d) a9a

(e) covtype

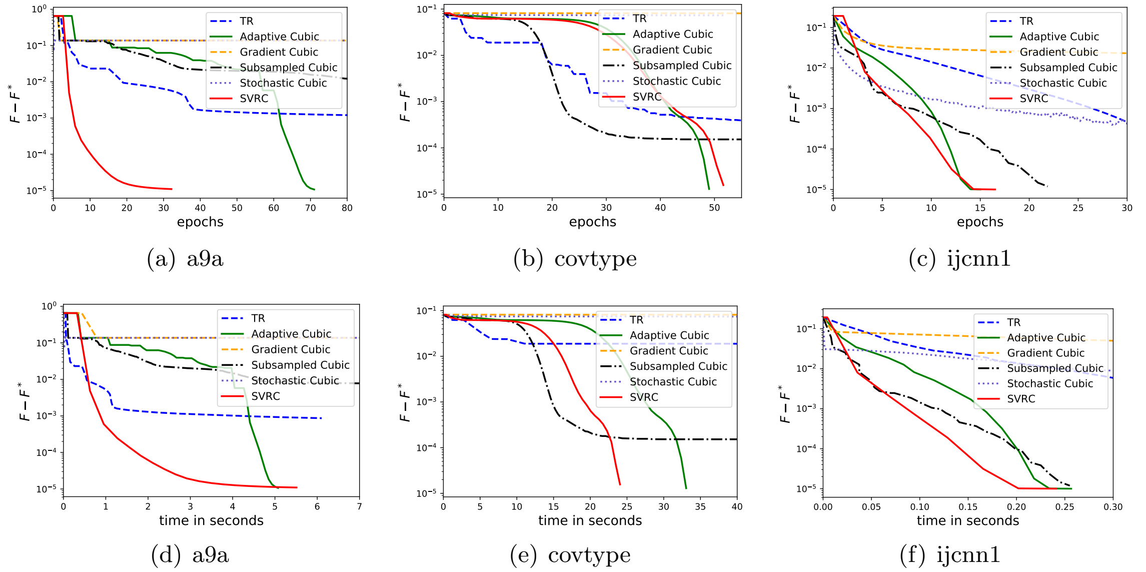
(f) ijcnn1

Figure 1: Logarithmic function value gap for nonconvex regularized logistic regression on different datasets. (a), (b) and (c) present the oracle complexity comparison; (d), (e) and (f) present the runtime comparison.

8.5. Experimental Results for Lite-SVRC

In this subsection, we present the experimental results for Lite-SVRC compared with all the baselines listed in Section 8.1. For Lite-SVRC, we use the same cubic subproblem solver used for SVRC in the previous subsection.

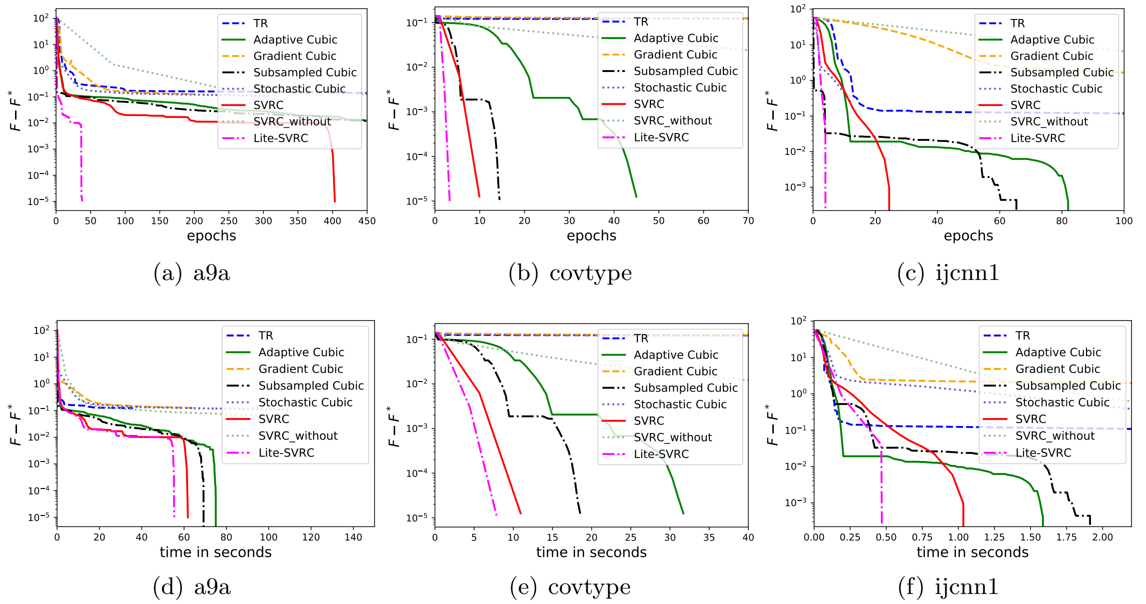
In the binary logistic regression problem in (26), for the nonconvex regularizer g in (25), we set $\lambda = 10^{-3}$ for all three datasets, and set $\gamma = 10, 50, 100$ for *a9a*, *ijcnn1* and *covtype* datasets respectively. The experimental results are displayed in Figure 3. The first row of the figure shows the plots of function value gap v.s. Hessian sample complexity of all the compared algorithms, and the second row presents the plots of function value gap v.s. CPU runtime (in seconds) of all the algorithms. It can be seen from Figure 3 that Lite-SVRC performs the best among all algorithms regarding both sample complexity of Hessian and runtime on all three datasets, which is consistent with our theoretical analysis. We remark that SVRC performs the second best in most settings in terms of both Hessian sample complexity and runtime. It should also be noted that although SVRC-without is also a variance-reduced method similar to Lite-SVRC and SVRC, it indeed performs much worse



(a) a9a (b) covtype (c) ijcnn1

(d) a9a (e) covtype (f) ijcnn1

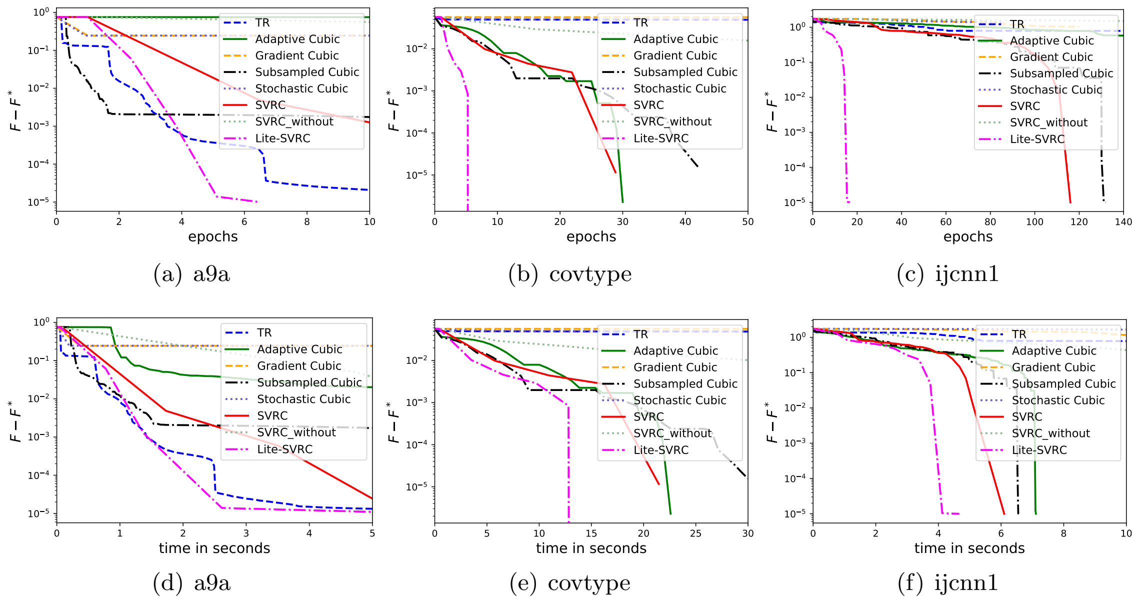
Figure 2: Logarithmic function value gap for nonlinear least square on different datasets. (a), (b) and (c) present the oracle complexity comparison; (d), (e) and (f) present the runtime comparison.



(a) a9a (b) covtype (c) ijcnn1

(d) a9a (e) covtype (f) ijcnn1

Figure 3: Function value gap of different algorithms for nonconvex regularized logistic regression problems on different datasets. (a)-(c) are plotted w.r.t. Hessian sample complexity. (d)-(e) are plotted w.r.t. CPU runtime.



(a) a9a

(b) covtype

(c) ijcnn1

(d) a9a

(e) covtype

(f) ijcnn1

Figure 4: Function value gap of different algorithms for nonlinear least square problems on different datasets. (a)-(c) are plotted w.r.t. Hessian sample complexity. (d)-(e) are plotted w.r.t. CPU runtime.

than other methods, because as we pointed out in the introduction, it needs to compute the minimum eigenvalue of the Hessian in each iteration, which actually makes the Hessian sample complexity even worse than Subsampled Cubic, let alone the runtime complexity.

For the least square problem in (27), the parameters λ and γ in the nonconvex regularizer for different datasets are set as follows: $\lambda = 5 \times 10^{-3}$ for all three datasets, and $\gamma = 10, 20, 50$ for *a9a*, *ijcnn1* and *covtype* datasets respectively. The experimental results are summarized in Figure 4, where the first row shows the plots of function value gap v.s. Hessian sample complexity and the second row presents the plots of function value gap v.s. CPU runtime (in seconds). It can be seen that Lite-SVRC again achieves the best performance among all the algorithms regarding to both sample complexity of Hessian and runtime when the required precision is high, which supports our theoretical analysis.

9. Conclusions

In this paper, we propose two novel second-order algorithms for non-convex optimization: SVRC and Lite-SVRC. Our proposed algorithm SVRC is the first algorithm which improves the oracle complexity of cubic regularization and its subsampled variants under certain regime using variance reduction techniques. We also show that similar oracle complexity also holds with inexact oracles. Under both exact and inexact oracle settings our algorithm outperforms the state-of-the-art methods. Furthermore, our proposed algorithm Lite-SVRC achieves a lower sample complexity of Hessian compared with SVRC and existing variance reduction based cubic regularization algorithms. Extensive experiments on various nonconvex optimization problems and datasets validate our theory.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1904183 and IIS-1906169. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

Appendix A. Proof of Main Theoretical Results for SVRC

In this section, we present the proofs of our main theoretical results for SVRC. Let us first recall the notations used in Algorithm 1. $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$ are the semi-stochastic gradient and Hessian defined in (5) and (6) respectively. $\mathbf{x}_t^s$'s are the iterates and $\widehat{\mathbf{x}}^s$'s are the reference points used in Algorithm 1. b_g and b_h are the batch sizes of semi-stochastic gradient and Hessian. S and T are the number of epochs and epoch length of Algorithm 1. We set $M_{s,t} := M = C_M L_2$ as suggested by Theorems 6 and 13, where $C_M > 0$ is a constant. $\mathbf{h}_t^s$ is the exact minimizer of $m_t^s(\mathbf{h})$, where $m_t^s(\mathbf{h})$ is defined in (7). $\widetilde{\mathbf{h}}_t^s$ is the inexact minimizer defined in (15).

In order to prove Theorems 6 and 13, we lay down the following useful technical lemmas. The first lemma is standard in the analysis cubic regularization methods.

Lemma 24 *Suppose F is L_2-Hessian Lipschitz for some constant $L_2 > 0$. For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have the following results:*

$$\mathbf{v}_t^s + \mathbf{U}_t^s \mathbf{h}_t^s + \frac{M}{2}\|\mathbf{h}_t^s\|_2 \mathbf{h}_t^s = 0, \tag{28}$$

$$\mathbf{U}_t^s + \frac{M}{2}\|\mathbf{h}_t^s\|_2 \mathbf{I} \succeq 0, \tag{29}$$

$$\langle \mathbf{v}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle \mathbf{U}_t^s \mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M}{6}\|\mathbf{h}_t^s\|_2^3 \le -\frac{M}{12}\|\mathbf{h}_t^s\|_2^3. \tag{30}$$

The next two important lemmas control the variances of $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$.

Lemma 25 *For the semi-stochastic gradient $\mathbf{v}_t^s$ defined in* (5)*, we have*

$$\mathbb{E}_{i_t}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{L_2^{3/2}}{b_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $\mathbb{E}_{i_t}$ is the expectation over all $i_t \in I_g$.

Lemma 26 *Let $\mathbf{U}_t^s$ be the semi-stochastic Hessian defined in* (6)*. If the batch size satisfy $b_h \ge 400 \log d$, then we have*

$$\mathbb{E}_{j_t}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \le 1200 L_2^3\bigg(\frac{\log d}{b_h}\bigg)^{3/2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $\mathbb{E}_{j_t}$ is the expectation over all $j_t \in I_h$.

Lemma 27 *For the semi-stochastic gradient and Hessian defined in* (5) *and* (6) *and* $\mathbf{h} \in \mathbb{R}^d$*, we have*

$$\begin{aligned} \langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}\rangle &\le \frac{M}{27}\|\mathbf{h}\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}}, \\ \big\langle \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}, \mathbf{h}\big\rangle &\le \frac{2M}{27}\|\mathbf{h}\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3. \end{aligned}$$

Lemma 28 *Let* $\mathbf{h} \in \mathbb{R}^d$ *and* $C_M \ge 100$*. For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have*

$$\begin{aligned} \mu(\mathbf{x}_t^s + \mathbf{h}) \le 9C_M^{3/2}\Big[&M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \\ &+ \|\nabla m_t^s(\mathbf{h})\|_2^{3/2} + M^{3/2}\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|^3\Big]. \end{aligned}$$

Lemma 29 *Let* $\mathbf{h} \in \mathbb{R}^d$ *and* $C \ge 3/2$*, For the semi-stochastic gradient and Hessian defined in* (5) *and* (6)*, we have*

$$\|\mathbf{x}_t^s + \mathbf{h} - \widehat{\mathbf{x}}^s\|_2^3 \le 2C^2\|\mathbf{h}\|_2^3 + (1 + 3/C)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{31}$$

Lemma 30 *We define constant series* c_t *for* $0 \le t \le T$ *as follows:* $c_T = 0$ *and* $c_t = c_{t+1}(1 + 3/T) + M(500T^3)^{-1}$ *for* $0 \le t \le T - 1$*. Then we have for any* $1 \le t \le T$*,*

$$M/24 - 2c_tT^2 \ge 0. \tag{32}$$

A.1. Proof of Theorem 6

Proof [Proof of Theorem 6] We first upper bound $F(\mathbf{x}_{t+1}^s)$ as follows

$$\begin{aligned} F(\mathbf{x}_{t+1}^s) &\le F(\mathbf{x}_t^s) + \langle \nabla F(\mathbf{x}_t^s), \mathbf{h}_t^s\rangle + \frac{1}{2}\big\langle \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}_t^s, \mathbf{h}_t^s\big\rangle + \frac{L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\ &= F(\mathbf{x}_t^s) + \langle \mathbf{v}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle \mathbf{U}_t^s\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M}{6}\|\mathbf{h}_t^s\|_2^3 + \langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}_t^s\rangle \\ &\quad + \frac{1}{2}\big\langle \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}_t^s, \mathbf{h}_t^s\big\rangle - \frac{M - L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\ &\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\mathbf{h}_t^s\|_2^3 + \bigg(\frac{M}{27}\|\mathbf{h}_t^s\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}}\bigg) \\ &\quad + \frac{1}{2}\bigg(\frac{2M}{27}\|\mathbf{h}_t^s\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\bigg) - \frac{M - L_2}{6}\|\mathbf{h}_t^s\|_2^3 \\ &\le F(\mathbf{x}_t^s) - \frac{M}{12}\|\mathbf{h}_t^s\|_2^3 + \frac{2}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3, \end{aligned} \tag{33}$$

where the first inequality follows from Lemma 40 and the second inequality holds due to Lemmas 24 and 27. We define

$$R_t^s = \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big], \tag{34}$$

where c_t is defined in Lemma 30. Then by Lemma 29, for $T \geq 3/2$ we have

$$c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 = c_{t+1}\|\mathbf{h}_t^s + \mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq 2c_{t+1}T^2\|\mathbf{h}_t^s\|_2^3 + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{35}$$

Applying Lemma 28 with $\mathbf{h} = \mathbf{h}_t^s$, we have

$$\begin{aligned}\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) &\leq \frac{M}{24}\|\mathbf{h}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2} \\ &\quad + \frac{\|\nabla m_t^s(\mathbf{h}_t^s)\|_2^{3/2}}{24M^{1/2}} + \frac{M}{24}\big|\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}_t^s\|_2\big|^3 \\ &= \frac{M}{24}\|\mathbf{h}_t^s\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2},\end{aligned} \tag{36}$$

where the equality is due to Lemma 24. Adding (33) with (35) and (36) and taking total expectation, we have

$$\begin{aligned}&R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \\ &= \mathbb{E}\Big[F(\mathbf{x}_{t+1}^s) + c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s)\Big] \\ &\leq \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 - \|\mathbf{h}_t^s\|_2^3\big(M/24 - 2c_{t+1}T^2\big)\big] \\ &\quad + \mathbb{E}\Big[3M^{-1/2}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + 28M^{-2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\Big] \\ &\leq \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] + \mathbb{E}\Big[3M^{-1/2}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \\ &\quad + 28M^{-2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\Big],\end{aligned} \tag{37}$$

where the third inequality holds due to Lemma 30. To further bound (37), we have

$$\frac{3}{M^{1/2}}\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \leq \frac{3L_2^{3/2}}{M^{1/2}b_g^{3/4}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \tag{38}$$

where the first inequality holds due to Lemma 25, the second inequality holds due to $M \geq 100L_2$ and $b_g \geq 5T^4$ from the condition of Theorem 6. We also have

$$\frac{28}{M^2}\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 \leq \frac{28 \times 15000L_2^3}{M^2(b_h/\log d)^{3/2}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \tag{39}$$

where the first inequality holds due to Lemma 26, where we have $b_h \geq 100T^2\log d \geq 400\log d$, and the second inequality holds due to $M \geq 100L_2$ and $b_h \geq 100T^2\log d$ from the assumption of Theorem 6. Thus, submitting (38) and (39) into (37), we have

$$\begin{aligned}R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\leq \mathbb{E}\bigg[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\bigg(c_{t+1}(1+3/T) + \frac{M}{500T^3}\bigg)\bigg] \\ &= \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] \\ &= R_t^s,\end{aligned} \tag{40}$$

where the first equality holds due to the definition of c_t in Lemma 30. Telescoping (40) from $t = 0$ to $T-1$, we have

$$R_0^s - R_T^s \geq \sum_{t=1}^{T} \big(240 C_M^2 L_2^{1/2}\big)^{-1} \mathbb{E}[\mu(\mathbf{x}_t^s)].$$

By the definition of c_T in Lemma 30, we have $c_T = 0$, then $R_T^s = \mathbb{E}\big[F(\mathbf{x}_T^s) + c_T\|\mathbf{x}_T^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}F(\widehat{\mathbf{x}}^{s+1})$; meanwhile by the definition of $\mathbf{x}_0^s$, we have $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$. Thus we have $R_0^s = \mathbb{E}\big[F(\mathbf{x}_0^s) + c_0\|\mathbf{x}_0^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}F(\widehat{\mathbf{x}}^s)$, which implies

$$\mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}) = R_0^s - R_T^s \geq \big(240 C_M^2 L_2^{1/2}\big)^{-1} \sum_{t=1}^{T} \mathbb{E}[\mu(\mathbf{x}_t^s)]. \tag{41}$$

Finally, telescoping (41) from $s = 1$ to S yields

$$\Delta_F \geq \sum_{s=1}^{S} \mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}) \geq \big(240 C_M^2 L_2^{1/2}\big)^{-1} \sum_{s=1}^{S}\sum_{t=1}^{T} \mathbb{E}[\mu(\mathbf{x}_t^s)].$$

By the definition about choice of $\mathbf{x}_{\text{out}}$, we complete the proof. ■

A.2. Proof of Corollary 9

Proof We can verify that the parameter setting in Corollary 9 satisfies the requirement of Theorem 6. Thus, submitting the choice of parameters into Theorem 6, the output of Algorithm 1 $\mathbf{x}_{\text{out}}$ satisfies that

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})] \leq \frac{240 C_M^2 L_2^{1/2} \Delta_F}{ST} \leq \epsilon^{3/2}, \tag{42}$$

which indeed implies that $\mathbf{x}_{\text{out}}$ is an $(\epsilon, \sqrt{L_2 \epsilon})$-approximate local minimum. Next we calculate how many SO calls and CSO calls are needed. Algorithm 1 needs to calculate full gradient $\mathbf{g}_s$ and full Hessian $\mathbf{H}_s$ at the beginning of each epoch, with n SO calls. In each epoch, Algorithm 1 needs to calculate $\mathbf{v}_t^s$ and $\mathbf{U}_t^s$ with $b_g + b_h$ SO calls at each iteration. Thus, the total amount of SO calls is

$$\begin{aligned} Sn + (ST)(b_g + b_h) &\leq n + C_1 \Delta_F L_2^{1/2} n^{4/5} \epsilon^{-3/2} + C_1 \Delta_F L_2^{1/2} \epsilon^{-3/2}(5n^{4/5} + 1000 n^{2/5} \log d) \\ &= \widetilde{O}\bigg(n + \frac{\Delta_F \sqrt{L_2} n^{4/5}}{\epsilon^{3/2}}\bigg), \end{aligned}$$

where $C_1 = 240 C_M^2$. For the CSO calls, Algorithm 1 needs to solve cubic subproblem at each single iteration. Thus, the total amount of CSO calls is

$$ST \leq C_1 \Delta_F L_2^{1/2} \epsilon^{-3/2} = O\bigg(\frac{\Delta_F \sqrt{L_2}}{\epsilon^{3/2}}\bigg).$$

■

A.3. Proof of Theorem 13

Proof [Proof of Theorem 13] Similar to (33) in the proof of Theorem 6, we have

$$\begin{aligned}
F(\mathbf{x}_{t+1}^s) &\le F(\mathbf{x}_t^s) + \big\langle \nabla F(\mathbf{x}_t^s), \widetilde{\mathbf{h}}_t^s \big\rangle + \frac{1}{2}\big\langle \nabla^2 F(\mathbf{x}_t^s)\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s \big\rangle + \frac{L_2}{6}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 \\
&= F(\mathbf{x}_t^s) + \big\langle \mathbf{v}_t^s, \widetilde{\mathbf{h}}_t^s \big\rangle + \frac{1}{2}\big\langle \mathbf{U}_t^s\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s \big\rangle + \frac{M}{6}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \big\langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \widetilde{\mathbf{h}}_t^s \big\rangle \\
&\quad + \frac{1}{2}\big\langle (\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s)\widetilde{\mathbf{h}}_t^s, \widetilde{\mathbf{h}}_t^s \big\rangle - \frac{M - L_2}{6}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \delta + \bigg(\frac{M}{27}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}}\bigg) \\
&\quad + \frac{1}{2}\bigg(\frac{2M}{27}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\bigg) - \frac{M - L_2}{6}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 \\
&\le F(\mathbf{x}_t^s) - \frac{M}{12}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \frac{2}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 + \delta,
\end{aligned} \tag{43}$$

where the second inequality holds because $\widetilde{\mathbf{h}}_t^s$ is an inexact solver satisfying Condition 11. By Lemma 29 with $\mathbf{h} = \widetilde{\mathbf{h}}_t^s$, we have

$$c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 = c_{t+1}\big\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s + \widetilde{\mathbf{h}}_t^s\big\|_2^3 \le 2c_{t+1}T^2\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + c_{t+1}(1 + 3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \tag{44}$$

By Lemma 28, we also have

$$\begin{aligned}
&\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) \\
&= \big(240C_M^2L_2^{1/2}\big)^{-1}\mu\big(\mathbf{x}_t^s + \widetilde{\mathbf{h}}_t^s\big) \\
&\le \frac{M}{24}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2} + \frac{\big\|\nabla m_t^s\big(\widetilde{\mathbf{h}}_t^s\big)\big\|_2^{3/2}}{24M^{1/2}} + \frac{M\big|\big\|\widetilde{\mathbf{h}}_t^s\big\|_2 - \|\mathbf{h}_t^s\|_2\big|^3}{24},
\end{aligned} \tag{45}$$

Since $\widetilde{\mathbf{h}}_t^s$ is an inexact solver satisfying Condition 11, we have

$$\frac{\big\|\nabla m_t^s\big(\widetilde{\mathbf{h}}_t^s\big)\big\|_2^{3/2}}{24M^{1/2}} + \frac{M\big|\big\|\widetilde{\mathbf{h}}_t^s\big\|_2 - \|\mathbf{h}_t^s\|_2\big|^3}{24} \le \frac{\delta}{24} + \frac{\delta}{24} < \delta. \tag{46}$$

Submitting (46) into (45), we have

$$\big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s) \le \frac{M}{24}\big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3 + \frac{\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{24M^{1/2}} + \frac{\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3}{24M^2} + \delta. \tag{47}$$

Then adding (43), (44) and (47) up, we have

$$\begin{aligned}
&R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \\
&= \mathbb{E}\Big[F(\mathbf{x}_{t+1}^s) + c_{t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(240C_M^2L_2^{1/2}\big)^{-1}\mu(\mathbf{x}_{t+1}^s)\Big]
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}\Big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 - \big\|\widetilde{\mathbf{h}}_t^s\big\|_2^3\big(M/24 - 2c_{t+1}T^2\big)\Big] \\
&\quad + \mathbb{E}\bigg[\frac{3}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{28}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\bigg] + 2\delta \\
&\leq \mathbb{E}\big[F(\mathbf{x}_t^s) + c_{t+1}(1+3/T)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] \\
&\quad + \mathbb{E}\bigg[\frac{3}{M^{1/2}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} + \frac{28}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3\bigg] + 2\delta. \qquad (48)
\end{aligned}
$$

Since the parameter setting is the same as Theorem 6, by (38) and (39), we have

$$
\begin{aligned}
\frac{3}{M^{1/2}}\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} &\leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \qquad (49)\\
\frac{28}{M^2}\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 &\leq \frac{M}{1000T^3}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \qquad (50)
\end{aligned}
$$

Submitting (49) and (50) into (48) yields

$$
\begin{aligned}
R_{t+1}^s + \big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\leq \mathbb{E}\bigg[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\bigg(c_{t+1}(1+3/T) + \frac{M}{500T^3}\bigg)\bigg] + 2\delta \\
&= \mathbb{E}\big[F(\mathbf{x}_t^s) + c_t\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\big] + 2\delta \\
&= R_t^s + 2\delta, \qquad (51)
\end{aligned}
$$

where the first equality holds due to the definition of c_t in Lemma 30. Telescoping (40) from $t = 0$ to $T-1$, we have

$$
R_0^s - R_T^s \geq \sum_{t=1}^{T}\Big(\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big).
$$

By the definition of c_T in Lemma 30, we have $c_T = 0$, then $R_T^s = \mathbb{E}\big[F(\mathbf{x}_T^s) + c_T\|\mathbf{x}_T^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})]$; meanwhile by the definition of $\mathbf{x}_0^s$, we have $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s$. Thus we have $R_0^s = \mathbb{E}\big[F(\mathbf{x}_0^s) + c_0\|\mathbf{x}_0^s - \widehat{\mathbf{x}}^s\|_2^3\big] = \mathbb{E}[F(\widehat{\mathbf{x}}^s)]$, which further implies

$$
\mathbb{E}[F(\widehat{\mathbf{x}}^s)] - \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})] = R_0^s - R_T^s \geq \sum_{t=1}^{T}\Big(\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big). \qquad (52)
$$

Finally, telescoping (52) from $s = 1$ to S, we obtain

$$
\Delta_F \geq \sum_{s=1}^{S}\mathbb{E}[F(\widehat{\mathbf{x}}^s)] - \mathbb{E}[F(\widehat{\mathbf{x}}^{s+1})] \geq \sum_{s=1}^{S}\sum_{t=1}^{T}\Big[\big(240C_M^2L_2^{1/2}\big)^{-1}\mathbb{E}[\mu(\mathbf{x}_t^s)] - 2\delta\Big].
$$

By the definition about choice of $\mathbf{x}_{\mathrm{out}}$, we finish the proof. ■

A.4. Proof of Corollary 15

Proof [Proof of Corollary 15] Under the parameter choice in Corollary 15, it holds that

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})] \leq \frac{240C_M^2 L_2^{1/2}\Delta_F}{ST} + 480C_M^2 L_2^{1/2}\delta \leq \epsilon^{3/2}/2 + \epsilon^{3/2}/2 = \epsilon^{3/2}. \tag{53}$$

Thus, $\mathbf{x}_{\text{out}}$ is an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum. By the proof of Corollary 9, the total amount of SO calls is

$$\begin{aligned} Sn + (ST)(b_g + b_h) &\leq n + C_1\Delta_F L_2^{1/2} n^{4/5}\epsilon^{-3/2} + C_1\Delta_F L_2^{1/2}\epsilon^{-3/2}(5n^{4/5} + 1000n^{2/5}\log d) \\ &= \widetilde{O}\bigg(n + \frac{\Delta_F\sqrt{L_2}n^{4/5}}{\epsilon^{3/2}}\bigg), \end{aligned}$$

where $C_1 = 480C_M^2$. For the CSO calls, Algorithm 1 needs to solve cubic subproblem at each single iteration. Thus, the total amount of CSO calls is

$$ST \leq C_1\Delta_F L_2^{1/2}\epsilon^{-3/2} = O\bigg(\frac{\Delta_F\sqrt{L_2}}{\epsilon^{3/2}}\bigg).$$

■

Appendix B. Proof of Technical Lemmas in Appendix A

In this section, we prove the technical lemmas used in Appendix A.

B.1. Proof of Lemma 24

The result of Lemma 24 is typical in the literature of cubic regularization (Nesterov and Polyak, 2006; Cartis et al., 2011a,b), but no exactly the same result has been shown in any formal way. Thus we present the proof here for self-containedness.

Proof [Proof of Lemma 24] For simplicity, we let $\mathbf{g} = \mathbf{v}_t^s, \mathbf{H} = \mathbf{U}_t^s, \theta = M_t$ and $\mathbf{h}_{\text{opt}} = \mathbf{h}_t^s$. Then we need to prove

$$\mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{h}_{\text{opt}} = \mathbf{0}, \tag{54}$$

$$\mathbf{H} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{I} \succeq \mathbf{0}, \tag{55}$$

$$\langle\mathbf{g}, \mathbf{h}_{\text{opt}}\rangle + \frac{1}{2}\langle\mathbf{H}\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}}\rangle + \frac{\theta}{6}\|\mathbf{h}_{\text{opt}}\|_2^3 \leq -\frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3. \tag{56}$$

Let $\lambda = \theta\|\mathbf{h}_{\text{opt}}\|_2/2$. Note that $\mathbf{h}_{\text{opt}} = \operatorname{argmin} m(\mathbf{h})$, then the necessary condition $\nabla m(\mathbf{h}_{\text{opt}}) = \mathbf{0}$ and $\nabla^2 m(\mathbf{h}_{\text{opt}}) \succeq \mathbf{0}$ can be written as

$$\nabla m(\mathbf{h}_{\text{opt}}) = \mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \lambda\mathbf{h}_{\text{opt}} = \mathbf{0}, \tag{57}$$

$$\mathbf{w}^\top\nabla^2 m(\mathbf{h}_{\text{opt}})\mathbf{w} = \mathbf{w}^\top\bigg(\mathbf{H} + \lambda\mathbf{I} + \lambda\bigg(\frac{\mathbf{h}_{\text{opt}}}{\|\mathbf{h}_{\text{opt}}\|_2}\bigg)\bigg(\frac{\mathbf{h}_{\text{opt}}}{\|\mathbf{h}_{\text{opt}}\|_2}\bigg)^\top\bigg)\mathbf{w} \geq 0, \forall\mathbf{w} \in \mathbb{R}^d. \tag{58}$$

Apparently, (57) directly implies (54). To prove (55), we adapt the proof of Lemma 5.1 in Agarwal et al. (2017). Note that if $\langle \mathbf{w}, \mathbf{h}_{\text{opt}} \rangle = 0$, then (58) directly implies (55). So we only need to focus on the case that $\langle \mathbf{w}, \mathbf{h}_{\text{opt}} \rangle \neq 0$.

Since $\langle \mathbf{w}, \mathbf{h}_{\text{opt}} \rangle \neq 0$, there exists $\eta \neq 0$ such that $\|\mathbf{h}_{\text{opt}} + \eta\mathbf{w}\|_2 = \|\mathbf{h}_{\text{opt}}\|_2$. (In fact, we can find $\eta = -2\langle \mathbf{w}, \mathbf{h}_{\text{opt}} \rangle / \|\mathbf{w}\|_2^2$ satisfies the requirement). Next we will take a close look at the difference $m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}})$. On one hand, we have

$$\begin{aligned}
& m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}}) \\
&= \mathbf{g}^\top[(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - \mathbf{h}_{\text{opt}}] + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top \mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top \mathbf{H}\mathbf{h}_{\text{opt}}}{2} \\
&= -[(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - \mathbf{h}_{\text{opt}}]^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top \mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top \mathbf{H}\mathbf{h}_{\text{opt}}}{2} \quad (59) \\
&= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + [\mathbf{h}_{\text{opt}} - (\mathbf{h}_{\text{opt}} + \eta\mathbf{w})]^\top \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top \mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} - \frac{\mathbf{h}_{\text{opt}}^\top \mathbf{H}\mathbf{h}_{\text{opt}}}{2} \\
&\qquad (60) \\
&= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + \frac{\mathbf{h}_{\text{opt}}^\top \mathbf{H}\mathbf{h}_{\text{opt}}}{2} - (\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})^\top \mathbf{H}(\mathbf{h}_{\text{opt}} + \eta\mathbf{w})}{2} \\
&= \frac{\lambda\eta^2}{2}\|\mathbf{w}\|_2^2 + \frac{\eta^2}{2}\mathbf{w}^\top \mathbf{H}\mathbf{w} = \frac{\eta^2}{2}\mathbf{w}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{w},
\end{aligned}$$

where (59) holds due to (57) and (60) holds due to the definition of η. On the other hand, by the definition of $\mathbf{h}_{\text{opt}}$, $m(\mathbf{h}_{\text{opt}} + \eta\mathbf{w}) - m(\mathbf{h}_{\text{opt}}) \geq 0$. Thus, we have proved (55). Finally, we prove (56) by showing that

$$\begin{aligned}
& \langle \mathbf{g}, \mathbf{h}_{\text{opt}} \rangle + \frac{1}{2}\langle \mathbf{H}\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}} \rangle + \frac{\theta}{6}\|\mathbf{h}_{\text{opt}}\|_2^3 \\
&= \left\langle \mathbf{g} + \mathbf{H}\mathbf{h}_{\text{opt}} + \frac{\theta}{2}\|\mathbf{h}_{\text{opt}}\|_2\mathbf{h}_{\text{opt}}, \mathbf{h}_{\text{opt}} \right\rangle - \frac{1}{2}\mathbf{h}_{\text{opt}}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} - \frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3 \\
&= -\frac{1}{2}\mathbf{h}_{\text{opt}}^\top(\mathbf{H} + \lambda\mathbf{I})\mathbf{h}_{\text{opt}} - \frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3 \qquad (61) \\
&\leq -\frac{\theta}{12}\|\mathbf{h}_{\text{opt}}\|_2^3, \qquad (62)
\end{aligned}$$

where (61) holds due to (54) and (62) holds due to (55). ■

B.2. Proof of Lemma 25

In order to prove Lemma 25, we need the following useful lemma.

Lemma 31 *Suppose* $\mathbf{a}_1, \ldots, \mathbf{a}_N$ *are i.i.d. and* $\mathbb{E}\mathbf{a}_i = 0$*, then*

$$\mathbb{E}\bigg\|\frac{1}{N}\sum_{i=1}^N \mathbf{a}_i\bigg\|_2^{3/2} \leq \frac{1}{N^{3/4}}\big(\mathbb{E}\|\mathbf{a}_i\|_2^2\big)^{3/4}.$$

Proof [Proof of Lemma 25] For simplification, we use $\mathbb{E}$ to replace $\mathbb{E}_{\mathbf{v}_{i_t}}$. We have

$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}$

$$= \mathbb{E}\bigg\| \frac{1}{b_g}\sum \big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\big] + \mathbf{g}^s - \bigg[\frac{1}{b_g}\sum \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s) - \mathbf{H}^s\bigg](\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)\bigg\|_2^{3/2}$$
$$= \mathbb{E}\bigg\| \frac{1}{b_g}\sum \big[\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)$$
$$+ \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\big]\bigg\|_2^{3/2}.$$

Now we set the parameters in Lemma 31 as $N = b_g$ and

$$\mathbf{a}_{i_t} = \nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^{s+1}) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s).$$

We can check that $\mathbf{a}_{i_t}$ satisfy the assumption of Lemma 31. Thus, by Lemma 31, we have

$$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{1}{b_g^{3/4}}\Big(\mathbb{E}\big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)$$
$$- \nabla F(\mathbf{x}_t^s) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2^2\Big)^{3/4}. \tag{63}$$

By Assumption 1 and Lemma 40, we have

$$\big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s) + \nabla F(\widehat{\mathbf{x}}^s) + \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2$$
$$\le \big\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) - \nabla^2 f_{i_t}(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2 + \big\|\nabla F(\mathbf{x}_t^s) - \nabla F(\widehat{\mathbf{x}}^s) - \nabla^2 F(\widehat{\mathbf{x}}^s)(\mathbf{x}_t^s - \widehat{\mathbf{x}}^s)\big\|_2$$
$$\le \frac{L_2}{2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 + \frac{L_2}{2}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2$$
$$= L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2. \tag{64}$$

Plugging (64) into (63) yields

$$\mathbb{E}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2} \le \frac{1}{b_g^{3/4}}\big(L_2^2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^4\big)^{3/4} = \frac{L_2^{3/2}}{b_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3.$$

■

B.3. Proof of Lemma 26

In order to prove Lemma 26, we need the following supporting lemma.

Lemma 32 *Suppose that $q \ge 2, p \ge 2$, and fix $r \ge \max\{q, 2\log p\}$. Consider i.i.d. random self-adjoint matrices $\mathbf{Y}_1, ..., \mathbf{Y}_N$ with dimension $p \times p$, $\mathbb{E}\mathbf{Y}_i = \mathbf{0}$. It holds that*

$$\bigg[\mathbb{E}\bigg\|\sum_{i=1}^N \mathbf{Y}_i\bigg\|_2^q\bigg]^{1/q} \le 2\sqrt{er}\bigg\|\bigg(\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\bigg)^{1/2}\bigg\|_2 + 4er\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}.$$

Proof [Proof of Lemma 26] For simplicity, we use $\mathbb{E}$ to denote $\mathbb{E}_{j_t}$. We have

$$\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 = \mathbb{E}\bigg\|\nabla^2 F(\mathbf{x}_t^s) - \frac{1}{b_h}\Big(\sum \big(\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s\big)\Big)\bigg\|_2^3$$

$$= \mathbb{E}\bigg\|\frac{1}{b_h}\bigg[\sum\big[\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big]\bigg]\bigg\|_2^3. \qquad (65)$$

We apply Lemma 32 with parameters

$$q = 3, p = d, r = 2\log p, \mathbf{Y}_{j_t} = \nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s), N = b_h.$$

It can be easily checked that these parameters satisfy the assumption of Lemma 32. Meanwhile, by Assumption 1, we have the following upper bound for $\mathbf{Y}_{j_t}$:

$$\begin{aligned}\big\|\mathbf{Y}_{j_t}\big\|_2 &= \big\|\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big\|_2 \\ &\le \big\|\nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s)\big\|_2 + \big\|\mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)\big\|_2 \\ &\le L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \\ &= 2L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2. \qquad (66)\end{aligned}$$

By Lemma 32, we have

$$\bigg[\mathbb{E}\Big\|\sum \mathbf{Y}_{j_t}\Big\|_2^3\bigg]^{1/3} \le 2\sqrt{er}\Big\|\Big(\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big)^{1/2}\Big\|_2 + 4er\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^3\big)^{1/3}. \qquad (67)$$

The first term in RHS of (67) can be bounded as

$$\begin{aligned}2\sqrt{er}\Big\|\Big(\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big)^{1/2}\Big\|_2 &= 2\sqrt{er}\Big\|\sum \mathbb{E}\mathbf{Y}_{j_t}^2\Big\|_2^{1/2} \\ &= 2\sqrt{Ner}\Big\|\mathbb{E}\mathbf{Y}_{j_t}^2\Big\|_2^{1/2} \\ &\le 2\sqrt{Ner}\Big(\mathbb{E}\Big\|\mathbf{Y}_{j_t}^2\Big\|_2\Big)^{1/2} \\ &= 2\sqrt{Ner}\Big(\mathbb{E}\|\mathbf{Y}_{j_t}\|_2^2\Big)^{1/2} \\ &\le 4L_2\sqrt{Ner}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2, \qquad (68)\end{aligned}$$

where the first inequality holds due to Jensen's inequality, the third equality holds because $\big\|\mathbf{Y}_{j_t}^2\big\|_2 = \|\mathbf{Y}_{j_t}\|_2^2$ and the last inequality holds due to (66). The second term in RHS of (67) can be bounded as

$$4er\big(\mathbb{E}\max_i \big\|\mathbf{Y}_i\big\|_2^3\big)^{1/3} \le 4er[(2L_2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2)^3]^{1/3} = 8L_2 er\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2. \qquad (69)$$

Submitting (68), (69) into (67), we have

$$\bigg[\mathbb{E}\Big\|\sum \mathbf{Y}_{j_t}\Big\|_2^3\bigg]^{1/3} \le 4L_2\sqrt{Ner}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + 8L_2 er\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2,$$

which immediately implies

$$\mathbb{E}\Big\|\frac{1}{N}\sum \mathbf{Y}_{j_t}\Big\|_2^3 \le 64L_2^3\bigg(\sqrt{\frac{er}{N}} + \frac{2er}{N}\bigg)^3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \qquad (70)$$

Submitting (70) into (65) with $\mathbf{Y}_{j_t} = \nabla^2 f_{j_t}(\mathbf{x}_t^s) - \nabla^2 f_{j_t}(\widehat{\mathbf{x}}^s) + \mathbf{H}^s - \nabla^2 F(\mathbf{x}_t^s)$, $r = 2\log d$, $N = b_h$, we have

$$\begin{aligned}\mathbb{E}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3 &\leq 64L_2^3\bigg(\sqrt{\frac{2e\log d}{b_h}} + \frac{4e\log d}{b_h}\bigg)^3 \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 \\ &\leq 1200L_2^3\bigg(\frac{\log d}{b_h}\bigg)^{3/2} \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,\end{aligned}$$

where the last inequality holds due to $b_h \geq 400\log d$. ■

B.4. Proof of Lemma 27

Proof we have

$$\begin{aligned}\langle \nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s, \mathbf{h}\rangle &\leq \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 \cdot \|\mathbf{h}\|_2 \\ &= \bigg(\frac{M^{1/3}}{9^{1/3}}\|\mathbf{h}\|_2\bigg) \cdot \bigg(\frac{9^{1/3}}{M^{1/3}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2\bigg) \\ &\leq \frac{1}{3}\bigg(\frac{M^{1/3}}{9^{1/3}}\|\mathbf{h}\|_2\bigg)^3 + \frac{2}{3}\bigg(\frac{9^{1/3}}{M^{1/3}}\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2\bigg)^{3/2} \\ &= \frac{M}{27}\|\mathbf{h}\|_2^3 + \frac{2\|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2^{3/2}}{M^{1/2}},\end{aligned}$$

where the second inequality holds due to Young's inequality. Meanwhile, we have

$$\begin{aligned}\big\langle\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big)\mathbf{h}, \mathbf{h}\big\rangle &\leq \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2 \cdot \|\mathbf{h}\|_2^2 \\ &= \bigg(\frac{M^{2/3}}{9^{2/3}}\|\mathbf{h}\|_2^2\bigg) \cdot \bigg(\frac{9^{2/3}}{M^{2/3}}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2\bigg) \\ &\leq \frac{2}{3}\bigg(\frac{M^{2/3}}{9^{2/3}}\|\mathbf{h}\|_2^2\bigg)^{3/2} + \frac{1}{3}\bigg(\frac{9^{2/3}}{M^{2/3}}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{H}^s\big\|_2\bigg)^3 \\ &= \frac{2M}{27}\|\mathbf{h}\|_2^3 + \frac{27}{M^2}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^3,\end{aligned}$$

where the second inequality holds due to Young's inequality. ■

B.5. Proof of Lemma 28

In order to prove Lemma 28, we need the following two useful lemmas.

Lemma 33 *Under Assumption 1, if $M \geq 2L_2$, then we have*

$$\big\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\big\|_2 \leq M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2.$$

Lemma 34 *Under Assumption 1, if $M \geq 2L_2$, then we have*

$$-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big) \leq M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}\|_2 - \|\mathbf{h}_t^s\|_2\big|.$$

Proof [Proof of Lemma 28] By the definition of μ, we can bound $\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\|_2^{3/2}$ and $0 \vee -L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big)\big]^3$ separately. To bound $\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\|_2^{3/2}$, applying Lemma 33 we have

$$\begin{aligned}
&\big\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\big\|_2^{3/2}\\
&\le \Big[M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s)-\mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s)-\mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2\Big]^{3/2}\\
&\le 2\Big[M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s)-\mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s)-\mathbf{U}_t^s\big\|_2^3 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2^{3/2}\Big],
\end{aligned}$$

where the second inequality holds due to the following basic inequality $(a+b+c+d)^{3/2} \le 2(a^{3/2}+b^{3/2}+c^{3/2}+d^{3/2})$. To bound $-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big)$, applying Lemma 26, we have

$$\begin{aligned}
&-L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big)\big]^3\\
&= -C_M^{3/2}M^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big)\big]^3\\
&\le C_M^{3/2}M^{-3/2}\Big[M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s)-\mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}\|_2-\|\mathbf{h}_t^s\|_2\big|\Big]^3\\
&\le 9C_M^{3/2}\Big[M^{3/2}\|\mathbf{h}\|_2^3 + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s)-\mathbf{U}_t^s\big\|_2^3 + M^{3/2}\big|\|\mathbf{h}\|_2-\|\mathbf{h}_t^s\|_2\big|^3\Big],
\end{aligned}$$

where the second inequality is due to $(a+b+c)^3 \le 9(a^3+b^3+c^3)$. Since $9C_M^{3/2} > 2$, we have

$$\begin{aligned}
\mu(\mathbf{x}_t^s+\mathbf{h}) &= \max\Big\{\|\nabla F(\mathbf{x}_t^s+\mathbf{h})\|_2^{3/2}, -L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s+\mathbf{h})\big)\big]^3\Big\}\\
&\le 9C_M^{3/2}\Big[M^{3/2}\|\mathbf{h}\|_2^3 + \|\nabla F(\mathbf{x}_t^s)-\mathbf{v}_t^s\|_2^{3/2} + M^{-3/2}\big\|\nabla^2 F(\mathbf{x}_t^s)-\mathbf{U}_t^s\big\|_2^3\\
&\qquad + \big\|\nabla m_t^s(\mathbf{h})\big\|_2^{3/2} + M^{3/2}\big|\|\mathbf{h}\|_2-\|\mathbf{h}_t^s\|_2\big|^3\Big],
\end{aligned}$$

which completes the proof. ■

B.6. Proof of Lemma 29

Proof For any $\mathbf{x}_t^s, \mathbf{h}, \widehat{\mathbf{x}}^s$ and a constant $C>0$, we have

$$\begin{aligned}
&\|\mathbf{x}_t^s+\mathbf{h}-\widehat{\mathbf{x}}^s\|_2^3\\
&\le \big(\|\mathbf{h}\|_2 + \|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2\big)^3\\
&= \|\mathbf{h}\|_2^3 + 3\|\mathbf{h}\|_2^2\cdot\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2 + 3\|\mathbf{h}\|_2\cdot\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^2 + \|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^3\\
&= \|\mathbf{h}\|_2^3 + 3\big(C^{1/3}\|\mathbf{h}\|_2^2\big)\cdot\frac{\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2}{C^{1/3}} + 3\big(C^{2/3}\|\mathbf{h}\|_2\big)\cdot\frac{\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^2}{C^{2/3}} + \|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^3\\
&\le \|\mathbf{h}\|_2^3 + 3\bigg(\frac{2}{3}\big(C^{1/3}\|\mathbf{h}\|_2^2\big)^{3/2} + \frac{1}{3}\bigg(\frac{\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2}{C^{1/3}}\bigg)^3\bigg)\\
&\qquad + 3\bigg(\frac{1}{3}\big(C^{2/3}\|\mathbf{h}\|_2\big)^3 + \frac{2}{3}\bigg(\frac{\|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^2}{C^{2/3}}\bigg)^{3/2}\bigg) + \|\mathbf{x}_t^s-\widehat{\mathbf{x}}^s\|_2^3
\end{aligned}$$

$$= \|\mathbf{h}\|_2^3 + \bigg(2C^{1/2}\|\mathbf{h}\|_2^3 + \frac{1}{C}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\bigg) + \bigg(C^2\|\mathbf{h}\|_2^3 + \frac{2}{C}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\bigg) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$$
$$\le 2C^2\|\mathbf{h}\|_2^3 + \bigg(1 + \frac{3}{C}\bigg)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|^3, \tag{71}$$

where the second inequality holds due to Young's inequality, the last inequality holds because $1 + 2\sqrt{C} \le C^2$ when $C \ge 3/2$. ■

B.7. Proof of Lemma 30

Proof By induction, we have for any $0 \le t \le T$,

$$c_t = M\frac{(1+3/T)^{T-t} - 1}{1500T^2}.$$

Then for any $0 \le t \le T$,

$$2c_tT^2 \le M\frac{2(1+3/T)^T}{1500} \le M\frac{2\cdot 27}{1500} < \frac{M}{24}.$$

■

Appendix C. Proof of Auxiliary Lemmas

In this section, we prove auxiliary lemmas used in Appendix B.

C.1. Proof of Lemma 31

Proof We have

$$\mathbb{E}\bigg\|\frac{1}{N}\sum_{i=1}^N \mathbf{a}_i\bigg\|_2^{3/2} = \frac{\mathbb{E}\|\sum_{i=1}^N \mathbf{a}_i\|_2^{3/2}}{N^{3/2}} \le \frac{(\mathbb{E}\|\sum_{i=1}^N \mathbf{a}_i\|_2^2)^{3/4}}{N^{3/2}} = \frac{(\sum_{i=1}^N \mathbb{E}\|\mathbf{a}_i\|_2^2)^{3/4}}{N^{3/2}} = \frac{\big(\mathbb{E}\|\mathbf{a}_i\|_2^2\big)^{3/4}}{N^{3/4}}.$$

The first inequality holds due to Lemma 41 with $s = 3/2, t = 2$. The second equality holds due to $\mathbb{E}\mathbf{a}_i = 0$ and that $\mathbf{a}_i$ are independently identically distributed. ■

C.2. Proof of Lemma 32

Proof This proof is mainly adapted from Chen et al. (2012); Tropp (2016). First, Let $\{\mathbf{Y}_i{}' : i = 1, \dots, N\}$ be an independent copy of the sequence $\{\mathbf{Y}_i : i = 1, \dots, N\}$. We denote $\mathbb{E}_{\mathbf{Y}'}$ to be the expectation over the independent copy $\mathbf{Y}'$. Then $\mathbb{E}_{\mathbf{Y}'}\mathbf{Y}_i{}' = 0$ and

$$\mathbb{E}\bigg\|\sum_{i=1}^N \mathbf{Y}_i\bigg\|_2^q = \mathbb{E}\bigg\|\sum_{i=1}^N \mathbb{E}_{\mathbf{Y}'}(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q \le \mathbb{E}\bigg[\mathbb{E}_{\mathbf{Y}'}\bigg\|\sum_{i=1}^N(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q\bigg] = \mathbb{E}\bigg\|\sum_{i=1}^N(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q. \tag{72}$$

The first equality holds due to $\mathbb{E}_{\mathbf{Y}'}\mathbf{Y}_i{}' = 0$, the first inequality holds because $\|\cdot\|_2^q$ is a convex function, and the second equality holds because we combine the iterated expectation into a single expectation.

Note that $\mathbf{Y}_i - \mathbf{Y}_i{}'$ has the same distribution as $\mathbf{Y}_i{}' - \mathbf{Y}_i$, thus the independent sequence $\{\xi_i(\mathbf{Y}_i - \mathbf{Y}_i{}') : 1 \le i \le n\}$ has the same distribution as $\{\mathbf{Y}_i - \mathbf{Y}_i{}' : 1 \le i \le N\}$, where ξ_i are independent Rademacher random variables, also independent with $\mathbf{Y}_i, \mathbf{Y}_i{}'$. Therefore,

$$\mathbb{E}\bigg\|\sum_{i=1}^{N}(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q = \mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q. \tag{73}$$

Furthermore, we have

$$\mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i(\mathbf{Y}_i - \mathbf{Y}_i{}')\bigg\|_2^q \le \mathbb{E}\bigg[2^{q-1}\bigg(\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_2^q + \bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i{}'\bigg\|_2^q\bigg)\bigg] = 2^q \cdot \mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_2^q. \tag{74}$$

The first inequality holds due to $\|\mathbf{A} - \mathbf{B}\|_2^q \le (\|\mathbf{A}\|_2 + \|\mathbf{B}\|_2)^q \le 2^{q-1}(\|\mathbf{A}\|_2^q + \|\mathbf{B}\|_2^q)$, where we let $\mathbf{A} = \sum_{i=1}^N \xi_i\mathbf{Y}_i, \mathbf{B} = \sum_{i=1}^N \xi_i\mathbf{Y}_i{}'$; the equality holds due to the identical distribution of $\{\xi\mathbf{Y}_i\}$ and $\{\xi\mathbf{Y}_i{}'\}$. Submitting (73), (74) into (72) yields

$$\mathbb{E}\bigg\|\sum_{i=1}^{N}\mathbf{Y}_i\bigg\|_2^q \le 2^q \cdot \mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_2^q \tag{75}$$

Taking q-th root for both sides, we have

$$\bigg[\mathbb{E}\bigg\|\sum_{i=1}^{N}\mathbf{Y}_i\bigg\|_2^q\bigg]^{1/q} \le 2\bigg[\mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_2^q\bigg]^{1/q}. \tag{76}$$

Next, we have the inequality chain:

$$\begin{aligned} 2\bigg[\mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_2^q\bigg]^{1/q} &\le 2\bigg[\mathbb{E}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_{S_r}^q\bigg]^{1/q} \\ &= 2\bigg[\mathbb{E}_{\mathbf{Y}_i}\bigg(\mathbb{E}_{\xi_i}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_{S_r}^q\bigg)\bigg]^{1/q} \\ &\le 2\bigg[\mathbb{E}_{\mathbf{Y}_i}\bigg(\mathbb{E}_{\xi_i}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_{S_r}^r\bigg)^{q/r}\bigg]^{1/q}, \end{aligned} \tag{77}$$

where the first inequality holds due to $\|\cdot\|_2 \le \|\cdot\|_{S_r}$, the second inequality holds due to Lyapunov's inequality (Lemma 41), where we set $s = q, t = r$. Since $q < r$, then the second inequality holds. Note we have

$$2\bigg[\mathbb{E}_{\mathbf{Y}_i}\bigg(\mathbb{E}_{\xi_i}\bigg\|\sum_{i=1}^{N}\xi_i\mathbf{Y}_i\bigg\|_{S_r}^r\bigg)^{q/r}\bigg]^{1/q} \le 2\sqrt{r}\bigg[\mathbb{E}\bigg\|\bigg(\sum_{i=1}^{N}\mathbf{Y}_i^2\bigg)^{1/2}\bigg\|_{S_r}^q\bigg]^{1/q}$$

$$
\begin{aligned}
&\le 2\sqrt{r}\bigg[\mathbb{E}\bigg(p^{1/r}\bigg\|\bigg(\sum_{i=1}^N \mathbf{Y}_i^2\bigg)^{1/2}\bigg\|_2\bigg)^q\bigg]^{1/q} \\
&\le 2\sqrt{er}\bigg[\mathbb{E}\bigg\|\bigg(\sum_{i=1}^N \mathbf{Y}_i^2\bigg)^{1/2}\bigg\|_2^q\bigg]^{1/q} \\
&= 2\sqrt{er}\bigg[\mathbb{E}\bigg\|\sum_{i=1}^N \mathbf{Y}_i^2\bigg\|_2^{q/2}\bigg]^{1/q},
\end{aligned} \tag{78}
$$

where the first inequality holds due to Proposition 42; the second inequality holds because $\|\mathbf{A}\|_{S_r} \le p^{1/r}\|\mathbf{A}\|_2$, where we set $\mathbf{A} = \big(\sum_{i=1}^N \mathbf{Y}_i^2\big)^{1/2}$ and p is the dimension of $\mathbf{A}$; the third inequality holds because $p^{1/r} \le p^{1/(2\log p)} = \sqrt{e}$.

Finally, we use Proposition 43 to bound (78). Since $\mathbf{Y}_i^2$ are positive-semidefinite and independent random matrices, we can set $\mathbf{W}_i$ in Proposition 43 as $\mathbf{W}_i = \mathbf{Y}_i^2$. Meanwhile, $q/2 \ge 1$, so we have

$$
\bigg[\mathbb{E}\bigg\|\sum_{i=1}^N \mathbf{Y}_i^2\bigg\|_2^{q/2}\bigg]^{2/q} \le \bigg[\bigg\|\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\bigg\|_2^{1/2} + 2\sqrt{er}\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}\bigg]^2,
$$

which immediately implies

$$
\bigg[\mathbb{E}\bigg\|\sum_{i=1}^N \mathbf{Y}_i^2\bigg\|_2^{q/2}\bigg]^{1/q} \le \bigg\|\sum_{i=1}^N \mathbb{E}\mathbf{Y}_i^2\bigg\|_2^{1/2} + 2\sqrt{er}\big(\mathbb{E}\max_i \|\mathbf{Y}_i\|_2^q\big)^{1/q}. \tag{79}
$$

Submitting (77), (78),(79) into (76), we have the proof completed. ■

C.3. Proof of Lemma 33

Proof We have

$$
\begin{aligned}
\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\|_2 &= \big\|\nabla F(\mathbf{x}_t^s + \mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h} + \mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h} \\
&\qquad + \big(\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\big) + \big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2 \\
&\le \big\|\nabla F(\mathbf{x}_t^s + \mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}\big\|_2 + \bigg\|\mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h} + \frac{M}{2}\|\mathbf{h}\|_2\mathbf{h}\bigg\|_2 \\
&\qquad + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \big\|\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2 + \frac{M}{2}\|\mathbf{h}\|_2^2,
\end{aligned} \tag{80}
$$

where the inequality holds due to triangle inequality. In the following, we are going to bound the right-hand side of (80). For the first term in the right-hand side of (80), it can be bounded as

$$
\big\|\nabla F(\mathbf{x}_t^s + \mathbf{h}) - \nabla F(\mathbf{x}_t^s) - \nabla^2 F(\mathbf{x}_t^s)\mathbf{h}\big\|_2 \le \frac{L_2}{2}\|\mathbf{h}\|_2^2 \le \frac{M}{4}\|\mathbf{h}\|_2^2,
$$

where the first inequality holds due to Assumption 1 and the second inequality holds due to $2L_2 \le M$. For the second term in the the right hand side of (80), it equals to

$$
\bigg\|\mathbf{v}_t^s + \mathbf{U}_t^s\mathbf{h} + \frac{M}{2}\|\mathbf{h}\|_2\mathbf{h}\bigg\|_2 = \big\|\nabla m_t^s(\mathbf{h})\big\|_2.
$$

And the final term can be bounded as

$$\big\|\big(\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big)\mathbf{h}\big\|_2 \le \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 \cdot \|\mathbf{h}\|_2 \le \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \frac{M}{4}\|\mathbf{h}\|_2^2,$$

where the last inequality is due to Young's inequality. Putting all these bounds together and submit them into (80), we have

$$\big\|\nabla F(\mathbf{x}_t^s + \mathbf{h})\big\|_2 \le M\|\mathbf{h}\|_2^2 + \|\nabla F(\mathbf{x}_t^s) - \mathbf{v}_t^s\|_2 + \frac{1}{M}\big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2^2 + \big\|\nabla m_t^s(\mathbf{h})\big\|_2.$$

■

C.4. Proof of Lemma 34

Proof We have

$$\begin{aligned}\nabla^2 F(\mathbf{x}_t^s + \mathbf{h}) &\succeq \nabla^2 F(\mathbf{x}_t^s) - L_2\|\mathbf{h}\|_2\mathbf{I}\\ &\succeq \mathbf{U}_t^s - \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2\mathbf{I} - L_2\|\mathbf{h}\|_2\mathbf{I}\\ &\succeq -\frac{M}{2}\|\mathbf{h}_t^s\|_2\mathbf{I} - \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2\mathbf{I} - L_2\|\mathbf{h}\|_2\mathbf{I},\end{aligned}$$

where the first inequality holds because $\nabla^2 F$ is L_2-Hessian Lipschitz, the last inequality holds due to (29) in Lemma 24. Thus we have

$$\begin{aligned}-\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_t^s + \mathbf{h})\big) &\le \frac{M}{2}\|\mathbf{h}_t^s\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + L_2\|\mathbf{h}\|_2\\ &= \frac{M}{2}(\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}\|_2) + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + (L_2 + M/2)\|\mathbf{h}\|_2\\ &\le M\|\mathbf{h}\|_2 + \big\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\big\|_2 + M\big|\|\mathbf{h}_t^s\|_2 - \|\mathbf{h}\|_2\big|,\end{aligned}$$

where the last inequality holds because $L_2 \le M/2$. ■

Appendix D. Proof of Main Theoretical Results for Lite-SVRC

In this section, we provide the proofs of theoretical results of Lite-SVRC (Algorithm 2).

D.1. Proof of Theorem 18

For the simplification of notation, we define $\mathbf{e}_{\mathbf{v}}, \mathbf{e}_{\mathbf{U}}$ as follows

$$\mathbf{e}_{\mathbf{v}} = \nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s, \quad \mathbf{e}_{\mathbf{U}} = \nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s, \tag{81}$$

where $\mathbf{v}_0^s = \mathbf{g}^s, \mathbf{U}_0^s = \mathbf{H}^s$. Before we state the proof, we present some technical lemmas that are useful in our analysis. Firstly, we give a sharp bound of $\mu(\mathbf{x}_{t+1}^s)$. A very crucial observation is that we can bound the norm of gradient $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2$ and the smallest eigenvalue of Hessian $\lambda_{\min}(\nabla^2 F(\mathbf{x}_{t+1}^s))$ with $\|\mathbf{h}_t^s\|_2$, $\|\mathbf{e}_{\mathbf{v}}\|_2$ and $\|\mathbf{e}_{\mathbf{U}}\|_2$ defined in (81).

Lemma 35 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have*

$$\mu(\mathbf{x}_{t+1}^s) \leq 9C_M^{3/2}\big(M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + \|\mathbf{e}_\mathbf{v}\|_2^{3/2} + M_{s,t}^{-3/2}\|\mathbf{e}_\mathbf{U}\|_2^3\big), \tag{82}$$

where C_M is the parameter in $M_{s,t} = C_M L_2$.

Lemma 35 suggests that to bound our target $\mathbb{E}\mu(\mathbf{x}_{t+1}^s)$, we only need to focus on $\mathbb{E}\|\mathbf{h}_t^s\|_2^3$, $\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2}$ and $\mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3$. The next lemma upper bound $F(\mathbf{x}_t^s) - F(\mathbf{x}_{t+1}^s)$ with $\mathbf{e}_\mathbf{v}, \mathbf{e}_\mathbf{U}$ and $\mathbf{h}_t^s$, which can be straightforwardly derived from the Hessian Lipschitz condition.

Lemma 36 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have the following result:*

$$F(\mathbf{x}_{t+1}^s) \leq F(\mathbf{x}_t^s) - M_{s,t}/12 \cdot \|\mathbf{h}_t^s\|_2^3 + C_1\big(\|\mathbf{e}_\mathbf{v}\|_2^{3/2}/M_{s,t}^{1/2} + \|\mathbf{e}_\mathbf{U}\|_2^3/M_{s,t}^2\big), \tag{83}$$

where $C_1 = 200$.

We can also bound $\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3$ with $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$ as follows.

Lemma 37 *Under the same assumption as in Theorem 18, let $\mathbf{h}_t^s, \mathbf{x}_t^s, \mathbf{x}_{t+1}^s, M_{s,t}$ be variables defined by Algorithm 2. Then we have the following result:*

$$\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 \leq (1 + 3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + 2n^{2/3}\|\mathbf{h}_t^s\|_2^3. \tag{84}$$

Finally, we bound the variance of $\widetilde{\mathbf{v}}_t^s$ and $\mathbf{U}_t^s$ as follows.

Lemma 38 *Under the same assumption as in Theorem 18, let $\mathbf{x}_t^s, \widetilde{\mathbf{v}}_t^s$ and $\widehat{\mathbf{x}}^s$ be the iterates defined in Algorithm 2. Then we have*

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\mathbf{e}_\mathbf{v}\|_2^{3/2} \leq \frac{3L_1^{3/2}}{D_g^{3/4}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where D_g is the batch size parameter defined in (21) *and $\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}$ takes expectation over $\widetilde{\mathbf{v}}_t^s$.*

Lemma 39 *Under the same assumption as in Theorem 18, let $\mathbf{x}_t^s, \mathbf{U}_t^s$ and $\widehat{\mathbf{x}}^s$ be iterates defined in Algorithm 2. If the batch size of Hessian satisfies $B_h > 400\log d$, we have*

$$\mathbb{E}_{\mathbf{U}_t^s}\|\mathbf{e}_\mathbf{U}\|_2^{3/2} \leq \frac{C_h L_2^3}{B_h^{3/2}}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where $C_h = 1200(\log d)^{3/2}$ and $\mathbb{E}_{\mathbf{U}_t^s}$ only takes expectation over $\mathbf{U}_t^s$.

Lemmas 38 and 39 suggest that with carefully selection of batch size, both $\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2}$ and $\mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3$ can be bounded by $\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3$, which play similar roles to Lemmas 25 and 26 in the analysis of SVRC.

Proof [Proof of Theorem 18] We first define $R_t^s = \mathbb{E}[F(\mathbf{x}_t^s) + c_{s,t}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3]$, where $c_{s,t}$ is a number series defined as follows: $c_{s,T} = 0$ and for $s = 1, \ldots, S$, $t = 0, \ldots, T-1$,

$$c_{s,t} = c_{s,t+1}\big(1 + 3/n^{1/3}\big) + M_{s,t}/(500n). \tag{85}$$

Combining Lemmas 36 and 37, we can upper bound $F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3$. Specifically, (83) + $c_{s,t+1}\times$ (84) yields

$$\begin{aligned}
&F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3\\
&\le F(\mathbf{x}_t^s) - \|\mathbf{h}_t^s\|_2^3\big(M_{s,t}/12 - 2n^{2/3}c_{s,t+1}\big) + C_1\big(\|\mathbf{e}_\mathbf{v}\|_2^{3/2}/M_{s,t}^{1/2} + \|\mathbf{e}_\mathbf{U}\|_2^3/M_{s,t}^2\big)\\
&\quad + c_{s,t+1}(1+3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3. \qquad (86)
\end{aligned}$$

By Lemma 35, multiplying (82) with $\big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}$ and adding it into (86) yields

$$\begin{aligned}
&F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mu(\mathbf{x}_{t+1}^s)\\
&\le F(\mathbf{x}_t^s) + c_{s,t+1}\big(1+3/n^{1/3}\big)\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\\
&\quad + \frac{(24C_1+1)}{24M_{s,t}^{1/2}}\cdot\|\mathbf{e}_\mathbf{v}\|_2^{3/2} + \frac{(24C_1+1)}{24M_{s,t}^2}\cdot\|\mathbf{e}_\mathbf{U}\|_2^3, \qquad (87)
\end{aligned}$$

where we use the fact that $M_{s,t}/24 - 2n^{2/3}c_{s,t+1} > 0$ which can be easily verified by the definition in (85) and a similar argument in Appendix B.7. By Lemmas 38 and 39 we have

$$\mathbb{E}\|\mathbf{e}_\mathbf{v}\|_2^{3/2} \le \frac{3L_1^{3/2}}{D_g^{3/4}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3, \qquad \mathbb{E}\|\mathbf{e}_\mathbf{U}\|_2^3 \le \frac{C_hL_2^3}{B_h^{3/2}}\mathbb{E}\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3,$$

where D_g, B_h are batch size parameters and $C_h = 1200(\log d)^{3/2}$. Taking expectation on (87), we obtain the following result

$$\begin{aligned}
&\mathbb{E}\big[F(\mathbf{x}_{t+1}^s) + c_{s,t+1}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 + \big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mu(\mathbf{x}_{t+1}^s)\big]\\
&\le \mathbb{E}\bigg[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\cdot\bigg(c_{s,t+1}\big(1+3/n^{1/3}\big) + \frac{6L_1^{3/2}C_1}{D_g^{3/4}M_{s,t}^{1/2}} + \frac{2L_2^3C_hC_1}{B_h^{3/2}M_{s,t}^2}\bigg)\bigg]\\
&\le \mathbb{E}\big[F(\mathbf{x}_t^s) + \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3\cdot\big(c_{s,t+1}\big(1+3/n^{1/3}\big) + M_{s,t}/(500n)\big)\big], \qquad (88)
\end{aligned}$$

where in the last inequality we use the fact that $M_{s,t} = C_ML_2$, $D_g \ge C_1n^{4/3}L_1^2/(L_2^2C_M^{4/3})$ and $B_h \ge 144(C_1C_h)^{2/3}n^{2/3}/C_M^2$. By the definition of $c_{s,t}$ in (85), (88) is equivalent to

$$\big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mathbb{E}(\mu(\mathbf{x}_{t+1}^s)) \le R_t^s - R_{t+1}^s. \qquad (89)$$

Then we sum up (89) from $t=0$ to $T-1$, while yields

$$\sum_{t=0}^{T-1}\big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \le R_0^s - R_T^s.$$

Substituting $\mathbf{x}_0^s = \widehat{\mathbf{x}}^s, \mathbf{x}_T^s = \widehat{\mathbf{x}}^{s+1}$ and $c_{s,T} = 0$ into the inequality above, we get

$$\sum_{t=0}^{T-1}\big(24\times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \le \mathbb{E}F(\widehat{\mathbf{x}}^s) - \mathbb{E}F(\widehat{\mathbf{x}}^{s+1}).$$

Then we take summation from $s = 1$ to S, we have

$$\begin{aligned}\sum_{s=1}^{S}\sum_{t=0}^{T-1}\big(24 \times 9C_M^{3/2}M_{s,t}^{1/2}\big)^{-1}\cdot\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] &\le \mathbb{E}F(\widehat{\mathbf{x}}^0) - \mathbb{E}F(\widehat{\mathbf{x}}^{S+1})\\ &\le F(\widehat{\mathbf{x}}^0) - \inf_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})\\ &= \Delta_F.\end{aligned}$$

Plugging $M_{s,t} = C_M L_2$ into the above inequality, we have

$$\sum_{s=1}^{S}\sum_{t=0}^{T-1}\mathbb{E}[\mu(\mathbf{x}_{t+1}^s)] \le 216C_M^2 L_2^{1/2}\Delta_F. \tag{90}$$

In Algorithm 2, we choose $\mathbf{x}_{\text{out}}$ randomly over s and t, thus we have our result from (90):

$$\mathbb{E}[\mu(\mathbf{x}_{\text{out}})] \le \frac{216C_M^2 L_2^{1/2}\Delta_F}{ST}.$$

This competes the proof. ■

D.2. Proof of Corollary 20

Now we provide the proof of our corollary for the sample complexity of Lite-SVRC.

Proof [Proof of Corollary 20] By the definition in (8) and the result in Theorem 18, to find an $(\epsilon, \sqrt{L_2\epsilon})$-approximate local minimum, we only need to make sure $216C_M^2 L_2^{1/2}\Delta_F/(ST) \le \epsilon^{3/2}$. Setting $T = n^{1/3}$, it suffices to let $S = O(\max\{L_2^{1/2}\Delta_F/(\epsilon^{3/2}n^{1/3}), 1\})$. We need to sample n Hessian at the beginning of each inner loop, and in each inner loop, we need to sample $B_h = \widetilde{O}(n^{2/3})$ Hessian matrices. Therefore, the total sample complexity of Hessian for Algorithm 2 is $S \cdot n + S \cdot T \cdot B_h = \widetilde{O}(n + n^{2/3}\cdot(\Delta_F\sqrt{L_2})/\epsilon^{3/2})$. The total amount of CSO calls is $O(ST) = O(\Delta_F\sqrt{L_2})/\epsilon^{3/2})$. ■

D.3. Proofs of Theorem 22 and Corollary 23

The proofs of the convergence of Lite-SVRC with an inexact cubic subproblem solver defined in Section 5 are almost the same as that of the convergence of SVRC. More specifically, the proof of Theorem 22 is the same as that of Theorem 13, and the proof of Corollary 23 is the same as that of Corollary 15. Therefore, we omit the proofs here for simplicity.

Appendix E. Proof of Technical Lemmas

In this section, we prove the technical lemmas used in the proof of Theorem 18.

E.1. Proof of Lemma 35

Proof [Proof of Lemma 35] Recall the definition of $\mu(\cdot)$ in (8). We need to upper bound $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}$ and $-L_2^{3/2}\lambda_{\min}^3(\nabla^2 F(\mathbf{x}_{t+1}^s))$, which can be achieved by applying Lemmas 33 and 34 since we have $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$. To bound $\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}$, we apply Lemma 33:

$$\begin{aligned}
&\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}\\
&\le \bigg[M_{s,t}\|\mathbf{h}_t^s\|_2^2 + \|\nabla m_t^s(\mathbf{h}_t^s)\|_2 + \|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2 + \frac{1}{M_{s,t}}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^2\bigg]^{3/2}\\
&\le 2\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} + \|\nabla m_t^s(\mathbf{h}_t^s)\|_2^{3/2} + M_{s,t}^{-3/2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\Big]\\
&\le 2\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} + M_{s,t}^{-3/2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\Big],
\end{aligned}$$

where the second inequality holds due to the basic inequality $(a+b+c+d)^{3/2} \le 2(a^{3/2} + b^{3/2} + c^{3/2} + d^{3/2})$ and in the last inequality we use the fact that $\nabla m_t^s(\mathbf{h}_t^s) = 0$. Next we bound $-M_{s,t}^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_{t+1}^s)\big)\big]^3$. Applying Lemma 34 with $\mathbf{h} = \mathbf{h}_t^s$, we have

$$\begin{aligned}
-L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_{t+1}^s)\big)\big]^3 &\le C_M^{3/2}M_{s,t}^{-3/2}\big[M_{s,t}\|\mathbf{h}_t^s\|_2 + \|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2\big]^3\\
&\le 9C_M^{3/2}M_{s,t}^{-3/2}\cdot\bigg[\frac{M_{s,t}^3}{4}\|\mathbf{h}_t^s\|_2^3 + \|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\bigg]\\
&\le 9C_M^{3/2}\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + M_{s,t}^{-3/2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\Big],
\end{aligned}$$

where the second inequality holds due to $(a+b+c)^3 \le 9(a^3+b^3+c^3)$. Thus, we have

$$\begin{aligned}
\mu(\mathbf{x}_{t+1}^s) &= \max\Big\{\|\nabla F(\mathbf{x}_{t+1}^s)\|_2^{3/2}, -L_2^{-3/2}\big[\lambda_{\min}\big(\nabla^2 F(\mathbf{x}_{t+1}^s)\big)\big]^3\Big\}\\
&\le 9C_M^{3/2}\Big[M_{s,t}^{3/2}\|\mathbf{h}_t^s\|_2^3 + \|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} + M_{s,t}^{-3/2}\|\nabla^2 F(\mathbf{x}_t^s) - \mathbf{U}_t^s\|_2^3\Big],
\end{aligned}$$

which completes the proof. ■

E.2. Proof of Lemma 36

Proof [Proof of Lemma 36] Note that $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$. Apply Lemma 40 and we have

$$\begin{aligned}
F(\mathbf{x}_{t+1}^s) = F(\mathbf{x}_t^s + \mathbf{h}_t^s) &\le F(\mathbf{x}_t^s) + \langle\nabla F(\mathbf{x}_t^s), \mathbf{h}_t^s\rangle + \frac{1}{2}\langle\nabla^2 F(\mathbf{x}_t^s)\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{L_2}{6}\|\mathbf{h}_t^s\|_2^3\\
&= F(\mathbf{x}_t^s) + \langle\widetilde{\mathbf{v}}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle\mathbf{U}_t^s\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3 + \langle\mathbf{e}_{\mathbf{v}}, \mathbf{h}_t^s\rangle\\
&\quad + \frac{1}{2}\langle\mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{L_2 - M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3. \qquad (91)
\end{aligned}$$

Based on Lemma 24 for the sub-problem in cubic regularization, we have

$$\langle\widetilde{\mathbf{v}}_t^s, \mathbf{h}_t^s\rangle + \frac{1}{2}\langle\mathbf{U}_t^s\mathbf{h}_t^s, \mathbf{h}_t^s\rangle + \frac{M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3 \le \frac{-M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3. \qquad (92)$$

Meanwhile, we have following two bounds on $\langle \mathbf{e}_{\mathbf{v}}, \mathbf{h}_t^s\rangle$ and $\langle \mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s, \mathbf{h}_t^s\rangle$ by Young's inequality:

$$\langle \mathbf{e}_{\mathbf{v}}, \mathbf{h}_t^s\rangle \le C_4\|\mathbf{e}_{\mathbf{v}}\|_2 \cdot \frac{1}{C_4}\|\mathbf{h}_t^s\|_2 \le C_4^{3/2}\|\mathbf{e}_{\mathbf{v}}\|_2^{3/2} + \frac{1}{C_4^3}\|\mathbf{h}_t^s\|_2^3, \tag{93}$$

$$\langle \mathbf{e}_{\mathbf{U}}\mathbf{h}_t^s, \mathbf{h}_t^s\rangle \le C_5^2\|\mathbf{e}_{\mathbf{U}}\|_2 \cdot \bigg(\frac{\|\mathbf{h}_t^s\|_2}{C_5}\bigg)^2 \le C_5^6\|\mathbf{e}_{\mathbf{U}}\|_2^3 + \frac{\|\mathbf{h}_t^s\|_2^3}{C_5^3}. \tag{94}$$

We set $C_4 = C_5 = (18/M_{s,t})^{1/3}$. Finally, because $L_2 \le M_{s,t}/2$, we have

$$\frac{L_2 - M_{s,t}}{6}\|\mathbf{h}_t^s\|_2^3 \le \frac{-M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3. \tag{95}$$

Substituting (92), (93), (94) and (95) into (91), we have the final result:

$$F(\mathbf{x}_{t+1}^s) \le F(\mathbf{x}_t^s) - \frac{M_{s,t}}{12}\|\mathbf{h}_t^s\|_2^3 + C_1\bigg(\frac{\|\mathbf{e}_{\mathbf{v}}\|_2^{3/2}}{M_{s,t}^{1/2}} + \frac{\|\mathbf{e}_{\mathbf{U}}\|_2^3}{M_{s,t}^2}\bigg), \tag{96}$$

where $C_1 = 200$. ■

E.3. Proof of Lemma 37

Proof [Proof of Lemma 37] Note that $\mathbf{x}_{t+1}^s = \mathbf{x}_t^s + \mathbf{h}_t^s$, then we have

$$\begin{aligned}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 &\le \big(\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 + \|\mathbf{h}_t^s\|_2\big)^3 \\ &= \|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + \|\mathbf{h}_t^s\|_2^3 + 3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2 + 3\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2. \end{aligned}\tag{97}$$

The inequality holds due to triangle inequality. Next, we bound $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2$ and $\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2$ by Young's inequality:

$$\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2 \cdot \|\mathbf{h}_t^s\|_2 = \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2}{n^{2/9}} \cdot n^{2/9}\|\mathbf{h}_t^s\|_2 \le \frac{2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3}{3n^{1/3}} + \frac{n^{2/3}\|\mathbf{h}_t^s\|_2^3}{3}, \tag{98}$$

$$\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2 \cdot \|\mathbf{h}_t^s\|_2^2 = \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2}{n^{1/9}} \cdot n^{1/9}\|\mathbf{h}_t^s\|_2^2 \le \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3}{3n^{1/3}} + \frac{2n^{1/6}\|\mathbf{h}_t^s\|_2^3}{3}. \tag{99}$$

Substituting (98), (99) into (97), we have the result:

$$\begin{aligned}\|\mathbf{x}_{t+1}^s - \widehat{\mathbf{x}}^s\|_2^3 &\le (1 + 1/n^{1/3} + 2/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + (1 + 2n^{1/6} + n^{2/3})\|\mathbf{h}_t^s\|_2^3 \\ &\le (1 + 3/n^{1/3})\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^3 + 2n^{2/3}\|\mathbf{h}_t^s\|_2^3,\end{aligned}$$

which completes the proof. ■

E.4. Proof of Lemma 38

Proof [Proof of Lemma 38] This proof is essentially the same as that of Lemma 25 in Section B.2. However, we replace the semi-stochastic gradient $\widetilde{\mathbf{v}}_t^s$ defined in (19) with $\mathbf{v}_t^s$ used in Lemma 25, which leads to the following inequality that is similar to (63):

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} \le 1/B_{g;s,t}^{3/4}\big(\mathbb{E}\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2^2\big)^{3/4}. \tag{100}$$

Next we bound $\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2$. By Assumption 17, we have

$$\begin{aligned}\|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s) + \mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2 &\le \|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\|_2 + \|\mathbf{g}^s - \nabla F(\mathbf{x}_t^s)\|_2\\ &= \|\nabla f_{i_t}(\mathbf{x}_t^s) - \nabla f_{i_t}(\widehat{\mathbf{x}}^s)\|_2 + \|\nabla F(\widehat{\mathbf{x}}^s) - \nabla F(\mathbf{x}_t^s)\|_2\\ &\le 2L_1\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2. \end{aligned} \tag{101}$$

Finally, substituting (101) and $B_{g;s,t} = D_g/\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2$ into (100), we have

$$\mathbb{E}_{\widetilde{\mathbf{v}}_t^s}\|\nabla F(\mathbf{x}_t^s) - \widetilde{\mathbf{v}}_t^s\|_2^{3/2} \le \frac{\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^{3/2}}{D_g^{3/4}} \cdot \big(4L_1^2\|\mathbf{x}_t^s - \widehat{\mathbf{x}}^s\|_2^2\big)^{3/4} \le \frac{2^{3/2}L_1^{3/2}}{D_g^{3/4}}\|\mathbf{x}_t^{s+1} - \widehat{\mathbf{x}}^s\|_2^3.$$

This completes the proof. ■

E.5. Proof of Lemma 39

Proof [Proof of Lemma 39] The proof of Lemma 39 is the same as that of Lemma 26 in Section B.3. Thus we omit it for simplicity. ■

Appendix F. Additional Lemmas and Propositions

It is obvious that Assumption 1 implies the Hessian Lipschitz assumption of F, which is also equivalent to the following lemma:

Lemma 40 *(Nesterov and Polyak, 2006) Suppose F is L_2-Hessian Lipschitz for some constant $L_2 > 0$, then we have*

$$\begin{aligned}\|\nabla^2 F(\mathbf{x}) - \nabla^2 F(\mathbf{y})\| &\le L_2\|\mathbf{x} - \mathbf{y}\|_2,\\ F(\mathbf{x}+\mathbf{h}) &\le F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2}\langle \nabla^2 F(\mathbf{x})\mathbf{h}, \mathbf{h}\rangle + \frac{L_2}{6}\|\mathbf{h}\|_2^3,\\ \|\nabla F(\mathbf{x}+\mathbf{h}) - \nabla F(\mathbf{x}) - \nabla^2 F(\mathbf{x})\mathbf{h}\|_2 &\le \frac{L_2}{2}\|\mathbf{h}\|_2^2.\end{aligned}$$

Lemma 41 (Lyapunov's Inequality) *(Durrett, 2010) For a random variable X, when $0 < s < t$, it holds that*

$$(\mathbb{E}|X|^s)^{1/s} \le (\mathbb{E}|X|^t)^{1/t}.$$

The following two lemmas are matrix concentration inequalities.

Proposition 42 (Matrix Khintchine Inequality) *(Mackey et al., 2014) Suppose $r > 2$. Consider a finite sequence $\{\mathbf{A}_i, 1 \le i \le N\}$ of deterministic, self-adjoint matrices. Then*

$$\left[\mathbb{E}\left\|\sum_{i=1}^{N}\xi_i\mathbf{A}_i\right\|_{S_r}^{r}\right]^{1/r} \le \sqrt{r}\left\|\left[\sum_{i=1}^{N}\mathbf{A}_i^2\right]^{1/2}\right\|_{S_r},$$

where sequence ξ_i consists of independent Rademacher random variables.

Proposition 43 *(Chen et al., 2012) Let $q \ge 1$, and fix $r \ge \max\{q, 2\log p\}$. Consider $\mathbf{W}_1, ..., \mathbf{W}_N$ of independent, random, positive-definite matrices with dimension $p \times p$. Then*

$$\left[\mathbb{E}\left\|\sum_{i=1}^{N}\mathbf{W}_i\right\|_2^{q}\right]^{1/q} \le \left[\left\|\sum_{i=1}^{N}\mathbb{E}\mathbf{W}_i\right\|_2^{1/2} + 2\sqrt{er}\big(\mathbb{E}\max_i\|\mathbf{W}_i\|_2^q\big)^{1/(2q)}\right]^2.$$

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199. ACM, 2017.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems*, pages 2676–2687, 2018.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707, 2016.

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, pages 3720–3730, 2018.

Albert A Bennett. Newton's method in general analysis. *Proceedings of the National Academy of Sciences*, 2(10):592–598, 1916.

Dimitri P Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999. ISBN 9781886529144.

Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust region method for nonconvex optimization. *arXiv preprint arXiv:1609.07428*, 2016.

Yair Carmon and John C Duchi. Gradient descent efficiently finds the cubic-regularized non-convex Newton step. *arXiv preprint arXiv:1612.00547*, 2016.

Yair Carmon and John C Duchi. Analysis of Krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 10705–10715, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Trust-region and other regularisations of linear least-squares problems. *BIT Numerical Mathematics*, 49(1):21–53, 2009.

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

Coralia Cartis, Nicholas I. M Gould, and Philippe L Toint. *Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity.* Springer-Verlag New York, Inc., 2011b.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012.

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM Journal on Optimization*, 23(3):1553–1574, 2013.

Richard Y Chen, Alex Gittens, and Joel A Tropp. The masked sample covariance estimator: an analysis using matrix concentration inequalities. *Information and Inference: A Journal of the IMA*, 1(1):2–20, 2012.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region Methods.* Society for Industrial and Applied Mathematics, 2000. ISBN 0-89871-460-5.

Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of $O(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162(1-2):1–32, 2017.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Rick Durrett. *Probability: theory and examples.* Cambridge university press, 2010.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal nonconvex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 686–696, 2018.

Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Robert Gower, Nicolas Le Roux, and Francis Bach. Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods. In *International Conference on Artificial Intelligence and Statistics*, pages 707–715, 2018.

Christopher J Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

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

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 1042–1085. PMLR, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1895–1904. PMLR, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic Newton method. *arXiv preprint arXiv:1503.08316*, 2015.

Lester Mackey, Michael I Jordan, Richard Y Chen, Brendan Farrell, Joel A Tropp, et al. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014.

José Mario Martínez and Marcos Raydan. Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *Journal of Global Optimization*, 68(2):367–385, 2017.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

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

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703, 2017.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pages 314–323, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for smooth nonconvex optimization. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1971–1977. IEEE, 2016b.

Anton Rodomanov and Dmitry Kropotov. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *International Conference on Machine Learning*, pages 2597–2605, 2016.

Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

Clément W Royer and Stephen J Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2): 1448–1477, 2018. doi: 10.1137/17M1134329.

Shai Shalev-Shwartz. SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754, 2016.

Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2904–2913, 2018.

Joel A Tropp. The expected norm of a sum of independent random matrices: An elementary approach. In *High Dimensional Probability VII*, pages 173–202. Springer, 2016.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

H. Wai, W. Shi, A. Nedic, and A. Scaglione. Curvature-aided incremental aggregated gradient method. In *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 526–532, 2017. doi: 10.1109/ALLERTON.2017.8262782.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Sample complexity of stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372v1*, 2018a.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. Stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372v2*, 2018b.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of Langevin dynamics based algorithms for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3125–3136, 2018a.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher R, and Michael W Mahoney. Sub-sampled Newton methods with non-uniform sampling. 2016.

Peng Xu, Farbod Roosta-Khorasan, and Michael W Mahoney. Second-order optimization for non-convex machine learning: An empirical study. *arXiv preprint arXiv:1708.07827*, 2017a.

Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *arXiv preprint arXiv:1708.07164*, 2017b.

Yi Xu, Jing Rong, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, pages 5531–5541, 2018b.

Chun Yu and Weixin Yao. Robust linear regression: A review and comparison. *Communications in Statistics-Simulation and Computation*, 46(8):6261–6282, 2017.

Yaodong Yu, Difan Zou, and Quanquan Gu. Saving gradient and negative curvature computations: Finding local minima more efficiently. *arXiv preprint arXiv:1712.03950*, 2017.

Yaodong Yu, Pan Xu, and Quanquan Gu. Third-order smoothness helps: Faster stochastic optimization algorithms for finding local minima. In *Advances in Neural Information Processing Systems*, pages 4526–4536, 2018.

Junyu Zhang, Lin Xiao, and Shuzhong Zhang. Adaptive stochastic variance reduction for subsampled Newton method with cubic regularization. *arXiv preprint arXiv:1811.11637*, 2018.

Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient Langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022, 2017.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularized Newton methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5990–5999. PMLR, 2018a.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3922–3933, 2018b.