Lower Bounds for Non-Convex Online Optimization: The Role and Efficiency of Second-Order Information

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley[◇]

October 14, 2024

Abstract

In this paper, we investigate the role of second-order information in non-convex stochastic optimization, focusing on the power and limitations of utilizing such information. Specifically, we consider the twin tasks of finding approximate *first-order* (FOSPs) and *second-order stationary points* (SOSPs) for functions with Lipschitz continuous gradients and Hessians. We analyze the efficiency of algorithms designed to locate these points, emphasizing the complexity of accessing and leveraging stochastic Hessian information. Our results highlight the conditions under which second-order information enhances optimization performance, providing a detailed characterization of the associated oracle complexity. Our analysis reveals that utilizing stochastic Hessian information can reduce the oracle complexity for finding stationary points, with a specific focus on the improvements under certain conditions. We demonstrate that our proposed algorithms achieve optimal performance in these scenarios, offering significant advancements over traditional first-order methods. Our findings offer insights into developing more efficient algorithms for non-convex stochastic optimization by examining the interplay between gradient and Hessian estimators.

Keywords: Non-Convex Optimization, Stochastic Gradient Descent, Second-Order Stationary Points, Hessian Estimator, Oracle Complexity, Lipschitz Continuity

1 Introduction

Non-convex optimization plays a central role in modern machine learning, where many fundamental problems, such as training deep neural networks, are inherently non-convex. Traditional optimization algorithms, including *Stochastic Gradient Descent* (SGD), have shown significant success in practice. However, these methods often converge to local minima or saddle points, which may not be satisfactory solutions. This has led to a growing interest in algorithms that can find *first-order stationary points* (FOSPs) or *second-order stationary points* (SOSPs), the latter being more likely to correspond to global minima in non-convex settings. The utilization of second-order information, particularly the Hessian, can provide critical insights into the curvature of the objective function, enabling more efficient navigation of the optimization landscape. However, accessing and effectively using this information in stochastic settings poses significant challenges. The computation of exact Hessians is often impractical due to the high dimensionality and large scale of modern datasets, necessitating the development of stochastic estimators.

In this paper, we explore the potential of stochastic second-order methods in non-convex optimization. We focus on algorithms that find approximate FOSPs and SOSPs and analyze their oracle complexity, providing a rigorous evaluation of their performance. Our study encompasses functions with Lipschitz continuous gradients and Hessians, ensuring the broad applicability of our

results to various machine learning problems. We contribute to the understanding of how second-order information can be leveraged to improve optimization efficiency. By examining the interplay between gradient and Hessian estimators, we highlight the conditions under which these methods can significantly outperform first-order techniques. Our findings offer valuable insights into the development of more robust and efficient optimization algorithms for non-convex stochastic problems.

Mathematically, let $F : \mathbb{R}^d \to \mathbb{R}$ have Lipschitz continuous gradient and Hessian, and consider the task of finding an (ϵ, δ)-second-order stationary point (SOSP), that is, $x \in \mathbb{R}^d$ such that

$$\|\nabla F(x)\| \le \epsilon \quad \text{and} \quad \nabla^2 F(x) \succeq -\delta I \tag{1}$$

This task plays a central role in the study of non-convex optimization: for functions satisfying a weak strict saddle condition [GHJY15], exact SOSPs (with $\epsilon = \delta = 0$) are local minima, and therefore the condition (1) serves as a proxy for approximate local optimality.[1] Moreover, for a growing set of non-convex optimization problems arising in machine learning, SOSPs are in fact *global minima* [GHJY15, GLM16, SQW18, MWCC20]. Consequently, there has been intense recent interest in the design of efficient algorithms for finding approximate SOSPs [JGN$^+$17, AZ18a, CDHS18, FLLZ18, TSJ$^+$18, XJY18, FLZ19].

In stochastic approximation tasks—particularly those motivated by machine learning—access to the objective function is often restricted to stochastic estimates of its gradient; for each query point $x \in \mathbb{R}^d$ we observe $\widehat{\nabla F}(x, z)$, where $z \sim P_z$ is a random variable such that

$$\mathbb{E}\big[\widehat{\nabla F}(x, z)\big] = \nabla F(x) \qquad \text{and} \quad \mathbb{E}\,\|\widehat{\nabla F}(x, z) - \nabla F(x)\|^2 \le \sigma_1^2 \tag{2}$$

This restriction typically arises due to computational considerations (when $\widehat{\nabla F}(\cdot, z)$ is much cheaper to compute than $\nabla F(\cdot)$, as in empirical risk minimization or Monte Carlo simulation), or due to fundamental online nature of the problem at hand (e.g., when x represents a routing scheme and z represents traffic on a given day). However, for many problems with additional structure, we have access to extra information. For example, we often have access to stochastic second-order information in the form of a Hessian estimator $\widehat{\nabla^2 F}(x, z)$ satisfying

$$\mathbb{E}\big[\widehat{\nabla^2 F}(x, z)\big] = \nabla^2 F(x) \qquad \text{and} \quad \mathbb{E}\,\|\widehat{\nabla^2 F}(x, z) - \nabla^2 F(x)\|_{\mathrm{op}}^2 \le \sigma_2^2 \tag{3}$$

In this paper, we characterize the extent to which the stochastic Hessian information (3), as well as higher-order information, contributes to the efficiency of finding first- and second-order stationary points. We approach this question from the perspective of *oracle complexity* [NY83], which measures efficiency by the number of queries to estimators of the form (2)—and possibly (3)—required to satisfy the condition (1).

In the remainder of this section we overview our results in greater detail. Unless otherwise stated, we assume F has both Lipschitz gradient and Hessian. To simplify the overview, we focus on dependence on ϵ^{-1} and δ^{-1} while keeping the other parameters—namely the initial optimality gap $F(x^{(0)}) - \inf_{x \in \mathbb{R}^d} F(x)$, the Lipschitz constants of ∇F and $\nabla^2 F$, and the variances of their estimators—held fixed. Our main theorems give explicit dependence on these parameters.

[1] However, it is NP-Hard to decide whether a SOSP is a local minimum or a high-order saddle point [MK87].

1.1 First-order stationary points: Lower bounds

For functions with Lipschitz gradient and Hessian, we prove an $\Omega(\epsilon^{-3.5})$ lower bound on the minimax oracle complexity of algorithms for finding stationary points using *only* stochastic gradients (2).[2] This lower bound is an extension of the results of [ACD+23], who showed that for functions with Lipschitz gradient but *not* Lipschitz Hessian, the optimal rate is $\Theta(\epsilon^{-4})$ using *only* stochastic gradients (2). Together with our new $O(\epsilon^{-3})$ upper bound, this lower bound reveals that stochastic Hessian-vector products offer an $\Omega(\epsilon^{-0.5})$ improvement in the oracle complexity for finding stationary points in the single-point query model. This contrasts the noiseless optimization setting, where finite gradient differences can approximate Hessian-vector products arbitrarily well, meaning these oracle models are equivalent.

The limitations of higher-order information ($p > 2$). For algorithms that can query both stochastic gradients and stochastic Hessians, we prove a lower bound of $\Omega(\epsilon^{-3})$ on the oracle complexity of finding an expected ϵ-stationary point. This proves that our $O(\epsilon^{-3})$ upper bound is optimal in the leading order term in ϵ, despite using only stochastic Hessian-vector products rather than full stochastic Hessian queries.

Notably, our $\Omega(\epsilon^{-3})$ lower bound extends to settings where stochastic higher-order oracles are available, i.e, when the first p derivatives are Lipschitz and we have bounded-variance estimators $\{\widehat{\nabla^q F}(\cdot,\cdot)\}_{q\le p}$. The lower bound holds for any finite p: for $p = 1$ the complexity is $\Theta(\epsilon^{-4})$ [ACD+23] while for all $p \ge 2$ it is $\Theta(\epsilon^{-3})$. This means that smoothness and stochastic derivatives beyond the second-order cannot improve the leading term in rates of convergence to stationarity, establishing a fundamental limitation of stochastic high-order information. This highlights another contrast with the noiseless setting, where pth order methods enjoy improved complexity for every p [CDHS20].

As we discuss in Appendix A, for multi-point stochastic oracles (4), the rate $O(\epsilon^{-3})$ is attainable even without stochastic Hessian access. Moreover, our $\Omega(\epsilon^{-3})$ lower bound for stochastic pth order oracles holds even when multi-point queries are allowed. Consequently, when viewed through the lens of worst-case oracle complexity, our lower bounds show that even stochastic Hessian information is not helpful in the multi-point setting.

1.2 First-order stationary points: Upper bounds

Using stochastic gradients and stochastic Hessian-vector products as primitives, we design a new variance-reduced gradient estimator. Plugging it into standard stochastic gradient descent (SGD), we obtain an algorithm that returns a point $\widehat{x}$ satisfying $\mathbb{E}\,\|\nabla F(\widehat{x})\| \le \epsilon$ and requires $O(\epsilon^{-3})$ stochastic gradient and Hessian-vector product queries in expectation. In comparison, vanilla SGD requires $O(\epsilon^{-4})$ queries [GL13], and the previously best known rate under our assumptions was $O(\epsilon^{-3.5})$, by both cubic-regularized Newton's method and a restarted variant of SGD [TSJ+18, FLZ19].

Our approach builds on a line of work by [FLLZ18, ZXG20, WJZ+19, CO19] that also develop algorithms with complexity $O(\epsilon^{-3})$, but require a "multi-point" oracle in which algorithm can query the stochastic gradient at multiple points for the same random seed. Specifically, in the n-point variant of this model, the algorithm can query at the set of points $(x_1, \ldots, x_n)$ and receive

$$\widehat{\nabla F}(x_1, z), \ldots, \widehat{\nabla F}(x_n, z) \quad \text{where} \quad z \overset{\text{i.i.d.}}{\sim} P_z \tag{4}$$

[2] We formally prove our results for the structured class of *zero-respecting algorithms* [CDHS20]; the lower bounds extend to general randomized algorithms via similar arguments to [ACD+23].

and where the estimator $\widehat{\nabla F}(x,z)$ is unbiased and has bounded variance in the sense of (2). The aforementioned works achieve $O(\epsilon^{-3})$ complexity using $n = 2$ simultaneous queries, while our new algorithm achieves the same rate using $n = 1$ (i.e., z is drawn afresh at each query), but using stochastic Hessian-vector products in addition to stochastic gradients. However, we show in Appendix A that under the statistical assumptions made in these works, the two-point stochastic gradient oracle model is *strictly* stronger than the single-point stochastic gradient/Hessian-vector product oracle we consider here. On the other hand, unlike our algorithm, these works do not require Lipschitz Hessian.

The algorithms that achieve complexity $O(\epsilon^{-3})$ using two-point queries work by estimating gradient differences of the form $\nabla F(x) - \nabla F(x')$ using $\widehat{\nabla F}(x,z) - \widehat{\nabla F}(x',z)$ and applying recursive variance reduction [NLST17]. Our primary algorithmic contribution is a second-order stochastic estimator for $\nabla F(x) - \nabla F(x')$ which avoids simultaneous queries while maintaining comparable error guarantees. To derive our estimator, we note that

$$\nabla F(x) - \nabla F(x') = \int_0^1 \nabla^2 F(xt + x'(1-t))(x - x')dt$$

and use K queries to the stochastic Hessian estimator (3) to numerically approximate this integral.[3] Specifically, our estimator takes the form

$$\frac{1}{K}\sum_{k=0}^{K-1} \widehat{\nabla^2 F}\left(x \cdot \left(1 - \frac{k}{K}\right) + x' \cdot \frac{k}{K}, z^{(i)}\right)(x - x') \tag{5}$$

where $z^{(i)} \overset{\text{i.i.d.}}{\sim} P_z$. Unlike the usual estimator $\widehat{\nabla F}(x,z) - \widehat{\nabla F}(x',z)$, the estimator (5) is biased. Nevertheless, we show that *choosing K dynamically* according to $K \propto \|x - x'\|^2$ provides adequate control over both bias and variance while maintaining the desired query complexity. Combining the integral estimator (5) with recursive variance reduction, we attain $O(\epsilon^{-3})$ complexity.[4]

1.3 Second-order stationary points

A lower bound for finding second-order stationary points. We prove a minimax lower bound which establishes that the stochastic second-order oracle complexity of finding (ϵ, δ)-SOSPs is $\Omega(\epsilon^{-3} + \delta^{-5})$. Consequently, the algorithms we develop have optimal worst-case complexity in the regimes $\delta = O(\epsilon^{2/3})$ and $\delta = \Omega(\epsilon^{0.5})$. Compared to our lower bounds for finding ε-stationary points, proving the $\Omega(\delta^{-5})$ lower bound requires a more substantial modification of the constructions of [CDHS20] and [ACD+23]. In fact, our lower bound is new even in the noiseless regime (i.e., $\sigma_1 = \sigma_2 = 0$), where it becomes $\Omega(\epsilon^{-1.5} + \delta^{-3})$; this matches the guarantee of the cubic-regularized Newton's method [NP06] and consequently characterizes the optimal rate for finding approximate SOSPs using noiseless second-order methods.

Upper bounds for general δ. We incorporate our recursive variance-reduced Hessian-vector product-based gradient estimator into an algorithm that combines SGD with negative curvature search. Under the slightly stronger (relative to (3)) assumption that the stochastic Hessians have almost surely bounded error, we prove that—with constant probability—the algorithm returns an

[3] More precisely, our estimator (5) only requires stochastic Hessian-vector products, whose computation is often roughly as expensive as that of a stochastic gradient [Pea94].

[4] The reader may refer to Table 1 for a succinct comparison of upper bounds.

(ϵ, δ)-SOSP after performing $O(\epsilon^{-3} + \epsilon^{-2}\delta^{-2} + \delta^{-5})$ stochastic gradient and Hessian-vector product queries.

1.4 More related work

We briefly survey additional lower and upper complexity bounds related to our work and place our results within their context. The works of [MS13, ASS19, AH18] delineate the second-order oracle complexity of *convex* optimization in the noiseless setting; [AS17] treat the finite-sum setting.

For functions with Lipschitz gradient and Hessian, oracle access to the Hessian significantly accelerates convergence to ε-approximate global minima, reducing the complexity from $\Theta(\varepsilon^{-0.5})$ to $\Theta(\varepsilon^{-0.286})$. However, since the hard instances for first-order convex optimization are quadratic [NY83, AS16, Sim18], assuming Lipschitz continuity of the Hessian does not improve the complexity if one only has access to a first-order oracle. This contrasts the case for finding ϵ-approximate stationary points of *non-convex* functions with noiseless oracles. There, Lipschitz continuity of the Hessian improves the first-order oracle complexity from $\Theta(\epsilon^{-2})$ to $O(\epsilon^{-1.75})$, with a lower bound of $\Omega(\epsilon^{-1.714})$ for deterministic algorithms [CDHS17, CDHS21]. Additional access to full Hessian further improves this complexity to $\Theta(\epsilon^{-1.5})$, and for pth-order oracles with Lipschitz pth derivative, the complexity further improves to $\Theta(\epsilon^{-(1+\frac{1}{p})})$ [CDHS20].

Notation. We let $\mathcal{C}^p$ denote the class of p-times differentiable real-valued functions, and let $\nabla^q F$ denote the qth derivative of a given function $F \in \mathcal{C}^p$ for $q \in \{1, \dots, p\}$. Given a function $F \in \mathcal{C}^1$, we let $\nabla_i F(x) \coloneqq [\nabla F(x)]_i = \frac{\partial}{\partial x_i} F(x)$. When $F \in \mathcal{C}^2$ is twice differentiable, we define, $\nabla^2_{ij} f(x) := \left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(x)$, and similarly define $[\nabla^p f(x)]_{i_1, i_2, \dots, i_p} = \frac{\partial^p}{\partial x_{i_1} \cdots \partial x_{i_p}} f(x)$ for pth-order derivatives. For a vector $x \in \mathbb{R}^d$, $\|x\|$ denotes the Euclidean norm and $\|x\|_\infty$ denotes the ℓ_∞ norm. For matrices $A \in \mathbb{R}^{d \times d}$, $\|A\|_{\mathrm{op}}$ denotes the operator norm. More generally, for symmetric pth order tensors T, we define the operator norm via $\|T\|_{\mathrm{op}} = \sup_{\|v\|=1} |\langle T, v^{\otimes p} \rangle|$, and we let $T[v^{(1)}, \dots, v^{(p)}] = \left\langle T, v^{(1)} \otimes \cdots \otimes v^{(p)} \right\rangle$. Note that for a vector $x \in \mathbb{R}^d$ the operator norm $\|x\|_{\mathrm{op}}$ coincides with the Euclidean norm $\|x\|$. We let $\mathbb{S}^d$ denote the space of symmetric matrices in $\mathbb{R}^{d \times d}$. We let $\mathbb{B}_r(x)$ denote the Euclidean ball of radius r centered at $x \in \mathbb{R}^d$ (with dimension clear from context). We adopt non-asymptotic big-O notation, where $f = O(g)$ for $f, g : \mathcal{X} \to \mathbb{R}_+$ if $f(x) \le C g(x)$ for some constant $C > 0$.

2 Setup

We study the problem of finding ϵ-stationary and (ϵ, δ)-second order stationary points in the standard oracle complexity framework [NY83], which we briefly review here.

Function classes. We consider p-times differentiable functions satisfying standard regularity conditions, and define

$$\mathcal{F}_p(\Delta, L_{1:p}) = \left\{ F : \mathbb{R}^d \to \mathbb{R} \;\middle|\; \begin{array}{l} F \in \mathcal{C}^p, \quad F(0) - \inf_x F(x) \le \Delta, \\ \|\nabla^q F(x) - \nabla^q F(y)\|_{\mathrm{op}} \le L_q \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^d,\ q \in [p] \end{array} \right\}$$

so that $L_{1:p} \coloneqq (L_1, \dots, L_p)$ specifies the Lipschitz constants of the qth order derivatives $\nabla^q F$ with respect to the operator norm. We make no restriction on the ambient dimension d.

Oracles. For a given function $F \in \mathcal{F}_p(\Delta, L_{1:p})$, we consider a class of stochastic pth order oracles defined by a distribution P_z over a measurable set $\mathcal{Z}$ and an estimator

$$\mathsf{O}_F^p(x, z) \coloneqq \left(\widehat{F}(x, z), \widehat{\nabla F}(x, z), \widehat{\nabla^2 F}(x, z), \ldots, \widehat{\nabla^p F}(x, z)\right) \tag{6}$$

where $\{\widehat{\nabla^q F}(\cdot, z)\}_{q=0}^p$ are unbiased estimators of the respective derivatives. That is, for all x, $\mathbb{E}_{z\sim P_z}[\widehat{F}(x, z)] = F(x)$ and $\mathbb{E}_{z\sim P_z}[\widehat{\nabla^q F}(x, z)] = \nabla^q F(x)$ for all $q \in [p]$.[5]

Given variance parameters $\sigma_{1:p} = (\sigma_1, \ldots, \sigma_p)$, we define the *oracle class* $\mathcal{O}_p(F, \sigma_{1:p})$ to be the set of all stochastic pth-order oracles for which the variance of the derivative estimators satisfies

$$\mathbb{E}_{z\sim P_z} \left\|\widehat{\nabla^q F}(x, z) - \nabla^q F(x)\right\|_{\mathrm{op}}^2 \le \sigma_q^2 \qquad q \in [p] \tag{7}$$

The upper bounds in this paper hold even when $\sigma_0^2 \coloneqq \max_{x\in\mathbb{R}^d} \mathrm{Var}(\widehat{F}(x, z))$ is infinite, while our lower bounds hold when $\sigma_0 = 0$, so to reduce notation, we leave dependence on this parameter tacit.

Optimization protocol. We consider stochastic pth-order optimization algorithms that access an unknown function $F \in \mathcal{F}_p(\Delta, L_{1:p})$ through multiple rounds of queries to a stochastic pth-order oracle $(\mathsf{O}_F^p, P_z) \in \mathcal{O}_p(F, \sigma_{1:p})$. When queried at $x^{(t)}$ in round t, the oracle performs an independent draw of $z^{(t)} \sim P_z$ and answers with $\mathsf{O}_F^p(x^{(t)}, z^{(t)})$. Algorithm queries depend on F only through the oracle answers; see e.g. [ACD+23, Section 2] for a more formal treatment.

3 Complexity of finding first-order stationary points: Lower bounds

We focus in this and next sections on the task of finding ϵ-approximate stationary points satisfying $\|\nabla F(x)\| \le \epsilon$. As prior work observes, cf., [CDHS17, AZ18a], stationary point search is a useful primitive for achieving the end goal of finding second-order stationary points (1). We begin with describing algorithmic lower bounds (for general pth-order) on the complexity of finding stationary points with stochastic second-order oracles, and then proceed to present algorithms that admit matching upper bounds.

In this section, we present with proofs a lower bound of $O(\epsilon^{-3})$ for the class of *zero-respecting* algorithms, which subsumes the majority of existing optimization methods (see Section 3.1 for a formal definition). In special, this lower bound holds even when one is given access to stochastic higher derivatives of *any* order. We believe that existing techniques [CDHS20, ACD+23] can strengthen our lower bounds to apply to general randomized algorithms; for brevity, we do not pursue it here.

The lower bounds in this section closely follow a recent construction by [ACD+23, Section 3], who prove lower bounds for stochastic first-order methods. To establish complexity bounds for pth-order methods, we extend the 'probabilistic zero-chain' gradient estimator introduced in [ACD+23] to high-order derivative estimators.The most technically demanding part of our proof is a careful scaling of the basic construction to simultaneously meet multiple Lipschitz continuity and variance constraints. Deferring the proof details to Section 3.1, our lower bound is as follows.

[5] For $p \ge 2$ we assume without loss of generality that $\widehat{\nabla^p F}(x, z)$ is a symmetric tensor.

Theorem 1. *For all $p \in \mathbb{N}$, $\Delta, L_{1:p}, \sigma_{1:p} > 0$ and $\epsilon \le O(\sigma_1)$, there exists $F \in \mathcal{F}_p(\Delta, L_{1:p})$ and $(\mathsf{O}_F^p, P_z) \in \mathcal{O}_p(F, \sigma_{1:p})$, such that for any pth-order zero-respecting algorithm, the number of queries required to obtain an ϵ-stationary point with constant probability is bounded from below by*

$$\Omega(1) \cdot \frac{\Delta \sigma_1^2}{\epsilon^3} \min\left\{ \min_{q \in \{2,\dots,p\}} \left(\frac{\sigma_q}{\sigma_1}\right)^{\frac{1}{q-1}}, \min_{q' \in \{1,\dots,p\}} \left(\frac{L_{q'}}{\epsilon}\right)^{1/q'} \right\} \tag{8}$$

A construction of dimension $\Theta\Big(\frac{\Delta}{\epsilon} \min\Big\{\min_{q\in\{2,\dots,p\}} \left(\frac{\sigma_q}{\sigma_1}\right)^{\frac{1}{q-1}}, \min_{q'\in\{1,\dots,p\}} \left(\frac{L_{q'}}{\epsilon}\right)^{1/q'}\Big\}\Big)$ realizes this lower bound.

For second-order methods (with $p = 2$), Theorem 1 specializes to the oracle complexity lower bound

$$\Omega(1) \cdot \min\left\{ \frac{\Delta \sigma_1 \sigma_2}{\epsilon^3}, \frac{\Delta L_2^{0.5} \sigma_1}{\epsilon^{3.5}}, \frac{\Delta L_1 \sigma_1^2}{\epsilon^4} \right\} \tag{9}$$

which is tight in that it matches (up to numerical constants) the convergence rate of Algorithm 2 in the regime where $\Delta \sigma_1 \sigma_2 \epsilon^{-3}$ dominates both the upper bound in Theorem 2 and expression (9). The lower bound (9) is also tight when the second-order information is not available or reliable (σ_2 is infinite or very large, respectively): Standard SGD matches the ϵ^{-4} term [GL13], while more sophisticated variants based on restarting [FLZ19] and normalized updates with momentum [CM20] match the $\epsilon^{-3.5}$ term (the former up to logarithmic factors)—neither of these algorithms requires stochastic second derivative estimation.

Theorem 1 implies that while higher-order methods (with $p > 2$) might achieve better dependence on the variance parameters than the upper bounds for Algorithm 2 or Algorithm 3, they cannot improve the ϵ^{-3} scaling. This highlights a fundamental limitation for higher-order methods in stochastic non-convex optimization which does not exist in the noiseless case. Indeed, without noise the optimal rate for finding ϵ-stationary point with a pth order method is $\Theta(\epsilon^{-1+\frac{1}{p}})$ [CDHS20].

Altogether, the results presented in this section fully characterize (with respect to dependence on ϵ) the complexity of finding ϵ-stationary points with stochastic second-order methods and beyond in the single-point query model. We briefly remark that lower bound in (8) immediately extends to multi-point queries, which shows that even second-order methods offer little benefit once two or more simultaneous queries are allowed.

3.1 Proof of Theorem 1

In this subsection, we prove Theorem 1. We begin by generalizing the lower bound framework of [ACD+23]—which centers around the notion of zero-respecting algorithms and stochastic gradient estimators called *probabilistic zero-chains*—to higher-order derivatives. Given a qth-order tensor $T \in \mathbb{R}^{\otimes q d}$, we define support $\{T\} \coloneqq \{i \in [d] \mid T_i \neq 0\}$, where T_i is the $(q-1)$-order subtensor defined by $[T_i]_{j_1,\dots,j_{q-1}} = T_{i,j_1,\dots,j_{q-1}}$. Given a tuple of tensors $\mathcal{T} = \big(T^{(1)}, T^{(2)}, \dots\big)$, we let support $\{\mathcal{T}\} \coloneqq \bigcup_i \text{support}\{T^{(i)}\}$ be the union of the supports of $T^{(i)}$. Lastly, given an algorithm A and a an oracle O_F^p, we let $x^{(t)}_{\mathsf{A}[\mathsf{O}_F^p]}$ denote the (possibly randomized) tth query point generated by A when fed by information from O (i.e., $x^{(t)}_{\mathsf{A}[\mathsf{O}_F^p]}$ is a measurable function of $\big\{\mathsf{O}_F^p(x^{(i)}, z^{(i)})\big\}_{i=1}^{t-1}$, and possibly a random seed $r^{(t)}$).

Definition 1. *A stochastic pth-order algorithm* A *is* zero-respecting *if for any function* F *and any pth-order oracle* O_F^p*, the iterates* $\{x^{(t)}\}_{t\in\mathbb{N}}$ *produced by* A *by querying* O_F^p *satisfy*

$$\operatorname{support}\big(x^{(t)}\big) \subseteq \bigcup_{i<t} \operatorname{support}\big(\mathsf{O}_F^p(x^{(i)}, z^{(i)})\big), \quad \text{for all } t \in \mathbb{N} \tag{10}$$

with probability one with respect the randomness of the algorithm and the realizations of $\{z^{(t)}\}_{t\in\mathbb{N}}$.

Given $x \in \mathbb{R}^d$, we define

$$\operatorname{prog}_\alpha(x) \coloneqq \max\{i \geq 0 \mid |x_i| > \alpha\} \;\; \text{(where we set } x_0 \coloneqq 1) \tag{11}$$

which represents the highest index of x whose entry is α-far from zero, for some threshold $\alpha \in [0, 1)$. To lighten notation, we further let $\operatorname{prog} \coloneqq \operatorname{prog}_0$. For a tensor T, we let $\operatorname{prog}(T) \coloneqq \max\{\operatorname{support}\{T\}\}$ denote the highest index in support $\{T\}$ (where $\operatorname{prog}(T) \coloneqq 0$ if $\operatorname{support}\{T\} = \emptyset$), and let $\operatorname{prog}(\mathcal{T}) \coloneqq \max_i \operatorname{prog}(T^{(i)})$ be the overall maximal index of $\operatorname{prog}(T^{(i)})$ for a tuple of tensors $\mathcal{T} = \big(T^{(1)}, T^{(2)}, \ldots\big)$.

Definition 2. *A collection of derivative estimators* $\widehat{\nabla^1 F}(x, z), \ldots, \widehat{\nabla^p F}(x, z)$ *for a function* F *forms a probability-*ρ *zero-chain if*

$$\Pr\Big(\exists x \;\; | \;\; \operatorname{prog}(\widehat{\nabla^1 F}(x, z), \ldots, \widehat{\nabla^p F}(x, z)) = \operatorname{prog}_{\frac{1}{4}}(x) + 1\Big) \leq \rho$$

and

$$\Pr\Big(\exists x \;\; | \;\; \operatorname{prog}(\widehat{\nabla^1 F}(x, z), \ldots, \widehat{\nabla^p F}(x, z)) = \operatorname{prog}_{\frac{1}{4}}(x) + i\Big) = 0 \;\; i > 1$$

No constraint is imposed for $i \leq \operatorname{prog}_{\frac{1}{4}}(x)$.

We note that the constant $1/4$ is used here for compatibility with the analysis in [ACD+23, Section 3]. Any non-negative constant less than $1/2$ would suffice in its place. The next lemma formalizes the idea that any zero-respecting algorithm interacting with a probabilistic zero-chain must wait many rounds to activate all the coordinates.

Lemma 1. *Let* $\widehat{\nabla^1 F}(x, z), \ldots, \widehat{\nabla^p F}(x, z)$ *be a collection of probability-*ρ *zero-chain derivative estimators for* $F : \mathbb{R}^T \to \mathbb{R}$*, and let* O_F^p *be an oracle with* $\mathsf{O}_F^p(x, z) = (\widehat{\nabla^q F}(x, z))_{q\in[p]}$*. Let* $\big\{x^{(t)}_{\mathsf{A}[\mathsf{O}_F]}\big\}$ *be a sequence of queries produced by* $\mathsf{A} \in \mathcal{A}_{\mathsf{zr}}(K)$ *interacting with* O_F^p*. Then, with probability at least* $1 - \gamma$*,*

$$\operatorname{prog}\Big(x^{(t)}_{\mathsf{A}[\mathsf{O}_F^p]}\Big) < T, \quad \text{for all } t \leq \frac{T - \log(1/\gamma)}{2\rho}$$

The proof of Lemma 1 is a simple adaptation of the proof of Lemma 1 of [ACD+23] to high-order zero-respecting methods—we provide it here for completeness. The proof idea is that any zero-respecting algorithm must activate coordinates in sequence, and must wait on average at least $\Omega(1/\rho)$ rounds between activations, leading to a total wait time of $\Omega(T/\rho)$ rounds.

Proof. Let $\{\widehat{\nabla^q F}(x^{(i)}, z^{(i)})\}_{q\in[p]}$ denote the oracle responses for the ith query made at the point $x^{(i)}$, and let $\mathcal{G}^{(i)}$ be the natural filtration for the algorithm's iterates, the oracle randomness, and the oracle answers up to time i. We measure the progress of the algorithm through two quantities:

$$\pi^{(t)} \coloneqq \max_{i\leq t} \operatorname{prog}\Big(x^{(i)}\Big) = \max\Big\{j \leq d \mid x_j^{(i)} \neq 0 \text{ for some } i \leq t\Big\}$$

$$\gamma^{(t)} := \max_{i \le t} \operatorname{prog}\Big(\nabla^q F(x^{(i)}, z^{(i)})\Big)$$
$$= \max\Big\{j \le d \mid \nabla^q f(x^{(i)}, z^{(i)})_j \neq 0 \text{ for some } i \le t \text{ and } q \in [p]\Big\}$$

Note that $\pi^{(t)}$ is the largest non-zero coordinate in $\operatorname{support}\{(x^{(i)})_{i \le t}\}$, and that $\pi^{(0)} = 0$ and $\gamma^{(0)} = 0$. Thus, for any zero-respecting algorithm

$$\pi^{(t)} \le \gamma^{(t-1)} \tag{12}$$

for all t. Moreover, observe that with probability one,

$$\operatorname{prog}\Big(\nabla^q F(x^{(t)}, z^{(t)})\Big) \le 1 + \operatorname{prog}_{\frac{1}{4}}(x^{(t)}) \le 1 + \operatorname{prog}(x^{(t)}) \le 1 + \pi^{(t)} \le 1 + \gamma^{(t-1)} \tag{13}$$

where the first inequality follows by the zero-chain property. Further, using the ρ-zero chain property, it follows that conditioned on $\mathcal{G}^{(i)}$, with probability at least $1 - \rho$,

$$\operatorname{prog}\Big(\nabla^q F(x^{(t)}, z^{(t)})\Big) \le \operatorname{prog}_{\frac{1}{4}}(x^{(t)}) \le \operatorname{prog}(x^{(t)}) \le \pi^{(t)} \le \gamma^{(t-1)} \tag{14}$$

Combining (13) and (14), we have that conditioned on $\mathcal{G}^{(i-1)}$,

$$\gamma^{(t-1)} \le \gamma^{(t)} \le \gamma^{(t-1)} + 1 \qquad \text{and} \qquad \Pr\Big[\gamma^{(t)} = \gamma^{(t-1)} + 1\Big] \le \rho$$

Thus, denoting the increments $\iota^{(t)} := \gamma^{(t)} - \gamma^{(t-1)}$, we have via the Chernoff method,

$$\Pr\Big[\gamma^{(t)} \ge T\Big] = \Pr\left[\sum_{j=1}^{t} \iota^{(j)} \ge T\right] \le \frac{\mathbb{E}\Big[\exp\Big(\sum_{j=1}^{t} \iota^{(j)}\Big)\Big]}{\exp(T)} = e^{-T}\,\mathbb{E}\left[\prod_{i=1}^{t} \mathbb{E}\Big[\exp\Big(\iota^{(i)}\Big) \mid \mathcal{G}^{(i-1)}\Big]\right]$$
$$\le e^{-T}(1 - \rho + \rho \cdot e)^t \le e^{2\rho t - T}$$

Thus, $\Pr\big[\gamma^{(t)} \ge T\big] \le \gamma$ for all $t \le \frac{T - \log(1/\gamma)}{2\rho}$; combined with (12), this yields the desired result. □

In light of Lemma 1, our lower bound strategy is as follows. We construct a function $F \in \mathcal{F}_p(\Delta, L_p)$ that both admits probability-ρ zero-chain derivative estimators and has large gradients for all $x \in \mathbb{R}^T$ with $\operatorname{prog}\big(x^{(i)}\big) < T$. Together with Lemma 1, this ensures that any zero-respecting algorithm interacting with a pth-order oracle must perform $\Omega(T/\rho)$ steps to make the gradient of F small. We make this approach concrete by adopting the construction used in [ACD+23], and adjusting it so as to be consistent with the additional high-order Lipschitz and variance parameters. For each $T \in \mathbb{N}$, we define

$$F_T(x) \coloneqq -\Psi(1)\Phi(x_1) + \sum_{i=2}^{T} [\Psi(-x_{i-1})\Phi(-x_i) - \Psi(x_{i-1})\Phi(x_i)] \tag{15}$$

where the component functions Ψ and Φ are

$$\Psi(x) = \begin{cases} 0, & x \le 1/2, \\ \exp\Big(1 - \frac{1}{(2x-1)^2}\Big), & x > 1/2 \end{cases} \qquad \text{and} \qquad \Phi(x) = \sqrt{e} \int_{-\infty}^{x} e^{-\frac{1}{2}t^2} dt \tag{16}$$

We start by collecting some relevant properties of F_T.

Lemma 2 ([CDHS20])**.** *The function F_T satisfies:*

(i) $F_T(0) - \inf_x F_T(x) \le \Delta_0 \cdot T$, *where* $\Delta_0 = 12$.

(ii) For $p \ge 1$, *the pth order derivatives of* F_T *are* ℓ_p*-Lipschitz continuous, where* $\ell_p \le e^{\frac{5}{2}p\log p + cp}$ *for a numerical constant* $c < \infty$.

(iii) For all $x \in \mathbb{R}^T$, $p \in \mathbb{N}$ *and* $i \in [T]$, *we have* $\|\nabla_i^p F_T(x)\|_{\mathrm{op}} \le \ell_{p-1}$.

(iv) For all $x \in \mathbb{R}^T$ *and* $p \in \mathbb{N}$, $\mathrm{prog}(\nabla^p F_T(x)) \le \mathrm{prog}_{\frac{1}{2}}(x) + 1$.

(v) For all $x \in \mathbb{R}^T$, *if* $\mathrm{prog}_1(x) < T$ *then* $\|\nabla F_T(x)\| \ge |\nabla_{\mathrm{prog}_1(x)+1} F_T(x)| > 1$.

Proof. Parts (i) and (ii) follow from Lemma 3 in [CDHS20] and its proof; Part (iii) is proven in Section 5; Part (iv) follows from Observation 3 in [CDHS20] and Part (v) is the same as Lemma 2 in [CDHS20]. □

The derivative estimators we use are defined as

$$\left[\widehat{\nabla^q F_T}(x,z)\right]_i := \left(1 + \mathbb{1}_{i > \mathrm{prog}_{\frac{1}{4}}(x)}\left(\frac{z}{\rho} - 1\right)\right) \cdot \nabla_i^q F_T(x) \tag{17}$$

where $z \sim \mathrm{Bernoulli}(\rho)$.

Lemma 3. *The estimators* $\widehat{\nabla^q F_T}$ *form a probability-*ρ *zero-chain, are unbiased for* $\nabla^q F_T$, *and satisfy*

$$\mathbb{E}\,\|\widehat{\nabla^q F_T}(x,z) - \nabla^q F_T(x)\|^2 \le \frac{\ell_{q-1}^2(1-\rho)}{\rho}, \quad \text{for all } x \in \mathbb{R}^T \tag{18}$$

Proof. First, we observe that $\mathbb{E}\left[\widehat{\nabla^q F_T}(x,z)\right] = \nabla^q F_T(x)$ for all $x \in \mathbb{R}^T$, as $\mathbb{E}[z/\rho] = 1$. Second, we argue that the probability-ρ zero-chain property holds. Recall that $\mathrm{prog}_\alpha(x)$ is non-increasing in α (in particular, $\mathrm{prog}_{\frac{1}{4}}(x) \ge \mathrm{prog}_{\frac{1}{2}}(x)$). Therefore, by Lemma 2.(iv), $[\widehat{\nabla^q F_T}(x,z)]_i = \nabla_i F_T(x) = 0$ for all $i > \mathrm{prog}_{\frac{1}{4}}(x) + 1$, all $x \in \mathbb{R}^T$ and all $z \in \{0,1\}$. In addition, since $z \sim \mathrm{Bernoulli}(\rho)$, we have $\Pr\Big(\exists x \;\mid\; \mathrm{prog}(\widehat{\nabla^1 F_T}(x,z), \ldots, \widehat{\nabla^p F_T}(x,z)) = \mathrm{prog}_{\frac{1}{4}}(x) + 1\Big) \le \rho$, establishing that the oracle is a probability-ρ zero-chain.

To bound the variance of the derivative estimators, we observe that $\widehat{\nabla^q F_T}(x,z) - \nabla^q F_T(x)$ has at most one nonzero $(q-1)$-subtensor in the coordinate $i_x = \mathrm{prog}_{\frac{1}{4}}(x) + 1$. Therefore,

$$\mathbb{E}\|\widehat{\nabla^q F_T}(x,z) - \nabla^q F_T(x)\|^2 = \left\|\nabla_{i_x}^q F_T(x)\right\|^2 \mathbb{E}\left(\frac{z}{\rho} - 1\right)^2 = \left\|\nabla_{i_x}^q F_T(x)\right\|^2 \frac{1-\rho}{\rho} \le \frac{(1-\rho)\ell_{q-1}^2}{\rho}$$

where the final inequality is due to Lemma 2.(iii), establishing the variance bound in (18). □

Proof of Theorem 1. We now prove the Theorem 1 by scaling the construction F_T appropriately. Let Δ_0 and ℓ_2 be the numerical constants in Lemma 2. Let the accuracy parameter ϵ, initial suboptimality Δ, derivative order $p \in \mathbb{N}$, smoothness parameters $L_1, \ldots, L_p$, and variance parameters $\sigma_1, \ldots, \sigma_p$ be fixed. We set

$$F_T^\star(x) = \alpha F_T(\beta x)$$

for some scalars α and β to be determined. The relevant properties of $F_T^\star$ scale as follows

$$F_T^\star(0) - \inf_x F_T^\star(x) = \alpha\big(F_T(0) - \inf_x F_T(\alpha x)\big) \le \alpha\Delta_0 T \tag{19}$$

$$\big\|\nabla^{q+1} F_T^\star(x)\big\| = \alpha\beta^{q+1}\big\|\nabla^{q+1} F_T(\beta x)\big\| \le \alpha\beta^{q+1}\ell_q \tag{20}$$

$$\|\nabla F_T^\star(x)\| \ge \alpha\beta\|\nabla F_T(x)\| \ge \alpha\beta \ \forall x \text{ s.t., } \mathrm{prog}_1(x) < T \tag{21}$$

The corresponding scaled derivative estimators $\widehat{\nabla^q F_T^\star}(x,z) = \alpha\beta^q\widehat{\nabla^q F_T}(\beta x, z)$ clearly form a probability-ρ zero-chain. Therefore, by Lemma 1, we have that for every zero respecting algorithm A interacting with $\mathsf{O}_{F_T^\star}^p$, with probability at least $1/2$, $\mathrm{prog}\Big(x^{(t)}_{\mathsf{A}[\mathsf{O}_{F_T^\star}^p]}\Big) < T$ for all $t \le (T-1)/2\rho$. Hence, since $\mathrm{prog}_1(x) \le \mathrm{prog}(x)$ for any $x \in \mathbb{R}^T$, we have by Lemma 2,

$$\mathbb{E}\|\nabla F_T^\star\big(x^{(t)}_{\mathsf{A}[\mathsf{O}_F^p]}\big)\| = \alpha\beta\mathbb{E}\|\nabla F_T\big(\beta x^{(t)}_{\mathsf{A}[\mathsf{O}_F^p]}\big)\| \ge \frac{\alpha\beta}{2} \quad \forall t \le (T-1)/2\rho \tag{22}$$

We bound the variance of the scaled derivative estimators as

$$\mathbb{E}\|\widehat{\nabla^q F_T^\star}(x,z) - \nabla^q F_T^\star(x)\|^2 = \alpha^2\beta^{2q}\mathbb{E}\left\|\widehat{\nabla^q F_T}(\beta x, z) - \nabla^q F_T(\beta x)\right\|^2 \le \frac{\alpha^2\beta^{2q}\ell_{q-1}^2(1-\rho)}{\rho}$$

where the last inequality follows by Lemma 3. Our goal now is to meet the following set of constraints:

- Δ-constraint: $\alpha\Delta_0 T \le \Delta$
- L_q-constraint: $\alpha\beta^{q+1}\ell_q \le L_q$, for $q \in [p]$
- ϵ-constraint: $\frac{\alpha\beta}{2} \ge \epsilon$
- σ_q-constraint: $\frac{\alpha^2\beta^{2q}\ell_{q-1}^2(1-\rho)}{\rho} \le \sigma_q^2$, for $q \in [p]$

Generically, since there are more inequalities to satisfy than the number of degrees of freedom (α, β, T and ρ) in our construction, not all inequalities can be activated (that is, met by equality) simultaneously. Different compromises will yield different rates.

First, to have a tight dependence in terms of ϵ, we activate the ϵ-constraint by setting $\alpha = 2\epsilon/\beta$. Next, we activate the σ_1-constraint, by setting $\rho = \min\{(\alpha\beta\ell_0/\sigma_1)^2, 1\} = \min\{(2\epsilon\ell_0/\sigma_1)^2, 1\}$. The bound on the variance of the qth-order derivative now reads

$$\frac{\alpha^2\beta^{2q}\ell_{q-1}^2(1-\rho)}{\rho} \le \frac{\sigma_1^2\alpha^2\beta^{2q}\ell_{q-1}^2}{(\alpha\beta\ell_0)^2} = \frac{\ell_{q-1}^2\beta^{2(q-1)}\sigma_1^2}{\ell_0^2} \qquad q = 2, \ldots, p$$

Since β is the only degree of freedom which can be tuned to meet though (not necessarily activate) the σ_q-constraint for $q = 2, \ldots, p$ and the L_q-constraints for $q = 1, \ldots, p$, we are forced to set

$$\beta = \min_{\substack{q=2,\ldots,p \\ q'=1,\ldots,p}} \min\left\{\left(\frac{\ell_0\sigma_q}{\ell_{q-1}\sigma_1}\right)^{\frac{1}{q-1}}, \left(\frac{L_{q'}}{2\epsilon\ell_{q'}}\right)^{1/q'}\right\} \tag{23}$$

Lastly, we activate the Δ-constraint by setting

$$T = \left\lfloor \frac{\Delta}{\alpha\Delta_0} \right\rfloor = \left\lfloor \frac{\Delta\beta}{2\Delta_0\epsilon} \right\rfloor$$

Assuming $(2\epsilon\ell_0/\sigma_1)^2 \le 1$ and $T \ge 3$, we have by (22) that the number of oracle queries required to obtain an ϵ-stationary point for $G_T^\star$ is bounded from below by

$$\begin{aligned}
\frac{T-1}{2\rho} &= \frac{1}{2\rho}\left(\left\lfloor \frac{\Delta\beta}{2\Delta_0\epsilon}\right\rfloor - 1\right) \\
&\overset{(\star)}{\ge} \frac{1}{2\rho}\cdot\frac{\Delta\beta}{4\Delta_0\epsilon} \\
&\ge \frac{\sigma_1^2}{2(2\ell_0\epsilon)^2}\cdot\frac{\Delta}{4\Delta_0\epsilon}\cdot\min_{\substack{q=2,\dots,p\\ q'=1,\dots,p}}\min\left\{\left(\frac{\ell_0\sigma_q}{\ell_{q-1}\sigma_1}\right)^{\frac{1}{q-1}}, \left(\frac{L_{q'}}{2\epsilon\ell_{q'}}\right)^{1/q'}\right\} \\
&\ge \frac{\Delta\sigma_1^2}{2^5\Delta_0\ell_0^2\epsilon^3}\cdot\min_{\substack{q=2,\dots,p\\ q'=1,\dots,p}}\min\left\{\left(\frac{\ell_0\sigma_q}{\ell_{q-1}\sigma_1}\right)^{\frac{1}{q-1}}, \left(\frac{L_{q'}}{2\epsilon\ell_{q'}}\right)^{1/q'}\right\} \qquad (24)
\end{aligned}$$

where $(\star)$ uses $\lfloor\xi\rfloor - 1 \ge \xi/2$ whenever $\xi \ge 3$, implying the desired bound. Lastly, we note that one can obtain tight lower complexity bounds for deterministic oracles by setting $\rho = 1$. Following the same chain of inequalities as in (24), in this case we get a lower oracle-complexity bound of

$$\frac{\Delta}{8\Delta_0\epsilon}\min_{q=1,\dots,p}\left(\frac{L_q}{2\epsilon\ell_q}\right)^{1/q} \qquad (25)$$

□

4 Complexity of finding first-order stationary points: Upper bounds

In this section, we present stochastic second-order methods with upper bounds that matches the $\Omega(\epsilon^{-3})$-complexity lower bound for finding ϵ-stationary points, as presented in Theorem 1. Our algorithms rely on *recursive variance reduction* [NLST17, FLLZ18, ZXG20]: we sequentially estimate the gradient at the points $\{x^{(t)}\}_{t\ge0}$ by accumulating cheap estimators of $\nabla F(x^{(\tau)}) - \nabla F(x^{(\tau-1)})$ for $\tau = t_0+1,\dots,t$, where at iteration t_0 we reset the gradient estimator by computing a high-accuracy approximation of $\nabla F(x^{(t_0)})$ with many oracle queries. Our implementation of recursive variance reduction, Algorithm 1, differs from previous approaches [FLLZ18, ZXG20, WJZ$^+$19] in three aspects.

First, in Line 8 we estimate differences of the form $\nabla F(x^{(\tau)}) - \nabla F(x^{(\tau-1)})$ by averaging stochastic Hessian-vector products. This allows us to do away with multi-point queries and operate under weaker assumptions than prior work (see Appendix A), but it also introduces bias to our estimator, which makes its analysis more involved. This is the key novelty in our algorithm. Second, rather than resetting the gradient estimator every fixed number of steps, we reset with a user-defined probability b (Line 4); this makes the estimator stateless and greatly simplifies its analysis, especially in our algorithms for finding SOSPs, where we use a varying value of b. Finally, we dynamically select the batch size K for estimating gradient differences based on the distance between iterates (Line 2), while prior work uses a constant batch size. Our dynamic batch size scheme is crucial for controlling the bias in our estimator, while still allowing for large step sizes as in [WJZ$^+$19]. The core of our analysis is the following lemma, which bounds the gradient estimation error and expected oracle complexity. To state the lemma, we let $\{x^{(t)}\}_{t\ge0}$ be sequence of queries to Algorithm 1, and let $g^{(t)} = \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b}(x^{(t)}, x^{(t-1)}, g^{(t-1)})$ be the sequence of estimates it returns.

Algorithm 1 Recursive variance reduction with stochastic Hessian-vector products (SPIDER-RVR)

// Gradient estimator for $F \in \mathcal{F}_2(\Delta, L_{1:2})$ given stochastic oracle in $\mathcal{O}_2(F, \sigma_{1:2})$

function SPIDER-RVR-GRADIENT-ESTIMATOR$_{\epsilon,b}(x,\, x_{\text{prev}},\, g_{\text{prev}})$:

  Set $K = \left\lceil \frac{5(\sigma_2^2 + L_2\epsilon)}{b\epsilon^2} \cdot \|x - x_{\text{prev}}\|^2 \right\rceil$ and $n = \left\lceil \frac{5\sigma_1^2}{\epsilon^2} \right\rceil$

  Sample $C \sim \text{Bernoulli}(b)$

  if C is 1 or g_{prev} is $\perp$ **then**

    Query the oracle n times at x and set $g \leftarrow \frac{1}{n}\sum_{j=1}^{n} \widehat{\nabla F}(x, z^{(j)})$, where $z^{(j)} \overset{\text{i.i.d.}}{\sim} P_z$.

  else

    Define $x^{(k)} := \frac{k}{K}x + \left(1 - \frac{k}{K}\right)x_{\text{prev}}$ for $k \in \{0, \ldots, K\}$.

    Query the oracle at the set of points $\left(x^{(k)}\right)_{k=0}^{K-1}$ to compute

$$g \leftarrow g_{\text{prev}} + \sum_{k=1}^{K} \widehat{\nabla^2 F}(x^{(k-1)}, z^{(k)})\left(x^{(k)} - x^{(k-1)}\right) \quad \text{where } z^{(k)} \overset{\text{i.i.d.}}{\sim} P_z.$$

  end if

  return g

end function

Algorithm 2 Stochastic gradient descent with SPIDER-RVR

Require: Oracle $(\mathsf{O}_F^2, P_z) \in \mathcal{O}_2(F, \sigma_{1:2})$ for $F \in \mathcal{F}_2(\Delta, L_1, L_2)$. Precision parameter ϵ

Set $\eta = \frac{1}{2\sqrt{L_1^2 + \sigma_2^2 + \epsilon L_2}}$, $T = \left\lceil \frac{2\Delta}{\eta\epsilon^2} \right\rceil$, $b = \min\left\{1, \frac{\eta\epsilon\sqrt{\sigma_2^2 + \epsilon L_2}}{\sigma_1}\right\}$

Initialize $x^{(0)}, x^{(1)} \leftarrow 0$, $g^{(0)} \leftarrow \perp$

for $t = 1$ to T **do**

  $g^{(t)} \leftarrow$ SPIDER-RVR-Gradient-Estimator$_{\epsilon,b}(x^{(t)}, x^{(t-1)}, g^{(t-1)})$

  $x^{(t+1)} \leftarrow x^{(t)} - \eta g^{(t)}$

end for

return $\widehat{x}$ chosen uniformly at random from $\left\{x^{(t)}\right\}_{t=1}^{T}$

Proposition 1. *For any oracle in $\mathcal{O}_2(F, \sigma_{1:2})$ and $F \in \mathcal{F}_2(\Delta, L_{1:2})$, Algorithm 1 guarantees that*

$$\mathbb{E}\,\|g^{(t)} - \nabla F(x^{(t)})\|^2 \le \epsilon^2$$

for all $t \ge 1$. Furthermore, conditional on $x^{(t-1)}$, $x^{(t)}$ and $g^{(t-1)}$, the t^{th} execution of Algorithm 1 with reset probability b uses at most

$$O\left(1 + b\frac{\sigma_1^2}{\epsilon^2} + \left\|x^{(t)} - x^{(t-1)}\right\|^2 \cdot \frac{\sigma_2^2 + \epsilon L_2}{b\epsilon^2}\right)$$

stochastic gradient and Hessian-vector product queries in expectation.

We prove the proposition in Section 4.1 by bounding the per-step variance using the Hessian-vector product oracle's variance bound (7), and by bounding the per-step bias relative to $\nabla F(x^{(t)}) - \nabla F(x^{(t-1)})$ using the Lipschitz continuity of the Hessian.

Our first algorithm for finding ϵ-stationary points, Algorithm 2, is simply stochastic gradient descent using the SPIDER-RVR gradient estimator (Algorithm 1); we bound its complexity by $O(\epsilon^{-3})$. Before stating the result formally, we briefly sketch the analysis here. Standard analysis of SGD with step size $\eta \le \frac{1}{2L_1}$ shows that its iterates satisfy $\mathbb{E}\|\nabla F(x^{(t)})\|^2 \le \frac{1}{\eta}\mathbb{E}[F(x^{(t+1)}) - F(x^{(t)})] +$

$O(1) \cdot \mathbb{E}\,\|g^{(t)} - \nabla F(x^{(t)})\|^2$. Telescoping over T steps, using Proposition 1 and substituting in the initial suboptimality bound Δ, this implies that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\,\|\nabla F(x^{(t)})\|^2 \le \frac{\Delta}{\eta T} + O(\epsilon^2) \tag{26}$$

Taking $T = \Omega(\frac{\Delta}{\eta\epsilon^2})$, we are guaranteed that a uniformly selected iterate has expected norm $O(\varepsilon)$.

To account for oracle complexity, we observe from Proposition 1 that T calls to Algorithm 1 require at most $T(\frac{\sigma_1^2 b}{\epsilon^2} + 1) + \sum_{t=1}^{T} \mathbb{E}\,\|x^{(t)} - x^{(t-1)}\|^2 \cdot \big(\frac{\sigma_2^2 + L_2\epsilon}{b\epsilon^2}\big)$ oracle queries in expectation. Using $x^{(t)} - x^{(t-1)} = \eta g^{(t-1)}$, Proposition 1 and (26) imply that $\sum_{t=1}^{T} \mathbb{E}\,\|x^{(t)} - x^{(t-1)}\|^2 \le O(T\epsilon^2)$. We then choose b to out the terms $T\big(\frac{\sigma_1^2 b}{\epsilon^2}\big)$ and $T\big(\frac{\sigma_2^2 + L_2\epsilon}{b}\big)$. This gives the following complexity guarantee, which we prove in Section 4.2.

Theorem 2. *For any function $F \in \mathcal{F}_2(\Delta, L_1, L_2)$, stochastic second-order oracle in $\mathcal{O}_2(F, \sigma_1, \sigma_2)$, and $\epsilon < \min\big\{\sigma_1, \sqrt{\Delta L_1}\big\}$, with probability at least $\frac{3}{4}$, Algorithm 2 returns a point $\widehat{x}$ such that $\|\nabla F(\widehat{x})\| \le \epsilon$ and performs at most*

$$O\Big(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3} + \frac{\Delta L_2^{0.5}\sigma_1}{\epsilon^{2.5}} + \frac{\Delta L_1}{\epsilon^2}\Big)$$

stochastic gradient and Hessian-vector product queries.

The oracle complexity of Algorithm 2 depends on the Lipschitz parameters of F only through lower-order terms in ϵ, with the leading term scaling only with the variance of the gradient and Hessian estimators. In the low noise regime where $\sigma_1 < \epsilon$ and $\sigma_2 < \max\{L_1, \sqrt{L_2\epsilon}\}$, the complexity becomes $O(\Delta L_1\epsilon^{-2} + \Delta L_2^{0.5}\epsilon^{-1.5})$ which is simply the maximum of the noiseless guarantees for gradient descent and Newton's method. We remark, however, that in the noiseless regime $\sigma_1 = \sigma_2 = 0$, a slightly better guarantee $O(\Delta L_1^{0.5} L_2^{0.25}\epsilon^{-1.75} + \Delta L_2^{0.5}\epsilon^{-1.5})$ is achievable [CDHS17].

In the noiseless setting, any algorithm that uses only first-order and Hessian-vector product queries must have complexity scaling with L_1, but full Hessian access can remove this dependence [CDHS21]. We show that the same holds true in the stochastic setting: Algorithm 3, a subsampled cubic regularized trust-region method using Algorithm 1 for gradient estimation, enjoys a complexity bound independent of L_1. We defer the analysis to Section 4.3 and state the guarantee as follows.

Theorem 3. *For any function $F \in \mathcal{F}_2(\Delta, \infty, L_2)$, stochastic second order oracle in $\mathcal{O}_2(F, \sigma_1, \sigma_2)$, and $\epsilon < \sigma_1$, with probability at least $\frac{3}{4}$, Algorithm 3 returns a point $\widehat{x}$ such that $\|\nabla F(\widehat{x})\| \le \epsilon$ and performs at most*

$$O\Big(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3} \cdot \log^{0.5} d + \frac{\Delta L_2^{0.5}\sigma_1}{\epsilon^{2.5}}\Big)$$

stochastic gradient and Hessian queries.

The guarantee of Theorem 3 constitutes an improvement in query complexity over Theorem 2 in the regime $L_1 \gtrsim (1 + \frac{\sigma_1}{\epsilon})(\sigma_2 + \sqrt{L_2\epsilon})$. However, depending on the problem, full stochastic Hessians can be up to d times more expensive to compute than stochastic Hessian-vector products.

Algorithm 3 Subsampled cubic-regularized trust-region method with SPIDER-RVR

Require: Oracle $(\mathsf{O}_F^2, P_z) \in \mathcal{O}_2(F, \sigma_{1:2})$ for $F \in \mathcal{F}_2(\Delta, \infty, L_2)$. Precision parameter ϵ

1: Set $M = 5\max\Big\{L_2, \frac{\epsilon\sigma_2^2\log(d)}{\sigma_1^2}\Big\}$, $\eta = 25\sqrt{\frac{\epsilon}{M}}$, $T = \Big\lceil\frac{5\Delta}{3\eta\epsilon}\Big\rceil$ and $n_H = \Big\lceil\frac{22\sigma_2^2\eta^2\log(d)}{\epsilon^2}\Big\rceil$

2: Set $b = \min\Big\{1, \frac{\eta\sqrt{\sigma_2^2+\epsilon L_2}}{25\sigma_1}\Big\}$

3: Initialize $x^{(0)}, x^{(1)} \leftarrow 0$, $g^{(0)} \leftarrow \perp$

4: **for** $t = 1$ to T **do**

5: Query the oracle n_H times at $x^{(t)}$ and compute

$$H^{(t)} \leftarrow \frac{1}{n_H}\sum_{j=1}^{n_H}\widehat{\nabla^2 F}(x^{(t)}, z^{(t,j)}) \quad \text{where} \quad z^{(t,j)} \overset{\text{i.i.d.}}{\sim} P_z$$

6: $g^{(t)} \leftarrow \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b}\big(x^{(t)}, x^{(t-1)}, g^{(t-1)}\big)$

7: Set the next point $x^{(t+1)}$ as

$$x^{(t+1)} \leftarrow \operatorname*{arg\,min}_{y:\|y-x^{(t)}\|\le\eta}\big\langle g^{(t)}, y - x^{(t)}\big\rangle + \frac{1}{2}\big\langle y - x^{(t)}, H^{(t)}(y - x^{(t)})\big\rangle + \frac{M}{6}\big\|y - x^{(t)}\big\|^3$$

8: **end for**

9: **return** $\widehat{x}$ chosen uniformly at random from $\big\{x^{(t)}\big\}_{t=2}^{T+1}$

4.1 Proof of Proposition 1

In this section we prove Proposition 1 on variance-reduced gradient estimator (SPIDER-RVR). First, we formally describe the protocol in which our optimization algorithms query the gradient estimator SPIDER-RVR-Gradient-Estimator described in Algorithm 1, and define some additional notation.

Given a function $F \in \mathcal{F}_2(\Delta, L_1, L_2)$ and a stochastic second-order oracle in $\mathcal{O}_2(F, \sigma_{1:2})$, the optimization algorithm interacts with SPIDER-RVR-Gradient-Estimator by sequentially querying points $\big\{x^{(t)}\big\}_{t=1}^{\infty}$ with reset probabilities $\big\{b^{(t)}\big\}_{t=1}^{\infty}$, to obtain estimates $g^{(t)}$ for $\nabla F(x^{(t)})$ for each time t; that is,

$$\begin{aligned} &x^{(t)} = \mathsf{A}^{(t)}(g^{(0)}, g^{(1)}, \ldots, g^{(t-1)}; r^{(t-1)})\ b^{(t)} = \mathsf{B}^{(t)}(r^{(t-1)}) \ \text{ and} \\ &g^{(t)} = \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b^{(t)}}(x^{(t)}, x^{(t-1)}, g^{(t-1)}) \end{aligned} \tag{27}$$

where $\mathsf{A}^{(t)}, \mathsf{B}^{(t)}$ are measurable mappings modeling the optimization algorithm and $\{r^{(t)}\}$ is an independent sequence of random seeds.[6] That is, Proposition 1 holds for any sequence of queries where $x^{(t)}$, and $b^{(t)}$ are adapted to the filtration

$$\mathcal{G}^{(t)} = \sigma\Big(\{g^{(j)}, r^{(j)}\}_{j<t}\Big)$$

but $b^{(t)}$ is independent of $\mathcal{G}^{(t-1)}$ and $g^{(t-1)}$.

Proposition 1 is an immediate consequence of Lemma 4 and Lemma 5, proven below, which respectively establish the estimator's error and complexity bounds.

[6] This level of formalism is not used within the proof, but we include it here for clarity.

Lemma 4. *Given a function $F \in \mathcal{F}_2(\Delta, \infty, L_2)$, a stochastic oracle in $\mathcal{O}_2(F, \sigma_{1:2})$, and initial points $x^{(0)}$ and $g^{(0)} = \perp$, let $\{g^{(t)}\}_{t\geq 0}$ denote the sequence of gradient estimates at $\{x^{(t)}\}_{t\geq 0}$ respectively, returned by* SPIDER-RVR-Gradient-Estimator *under the protocol* (27)*. Then, for all $t \geq 1$,*

$$\mathbb{E}\left\|g^{(t)} - \nabla F(x^{(t)})\right\|^2 \leq \epsilon^2$$

Proof. We prove that

$$\mathbb{E}\left\|g^{(t)} - \nabla F(x^{(t)})\right\|^2 \leq \left(1 - \frac{\mathbb{E}[b^{(t)}]}{2}\right)\mathbb{E}\left\|g^{(t-1)} - \nabla F(x^{(t-1)})\right\|^2 + \frac{\mathbb{E}[b^{(t)}]}{2}\epsilon^2$$

whence the result follows by a simple induction whose basis is

$$\mathbb{E}\left\|g^{(1)} - \nabla F(x^{(1)})\right\|^2 \leq \frac{\sigma_1^2}{n} \leq \epsilon^2$$

Let $C^{(t)}$ denote the value of the coin toss in the t^{th} call to Algorithm 1 (Line 3), recalling that $C^{(t)} \sim \text{Bernoulli}(b^{(t)})$. Writing $\mathfrak{e}^{(t)} = g^{(t)} - \nabla F(x^{(t)})$ for brevity, we have

$$\mathbb{E}\left[\left\|\mathfrak{e}^{(t)}\right\|^2 \;\middle|\; b^{(t)}\right] = b^{(t)}\,\mathbb{E}\left[\left\|\mathfrak{e}^{(t)}\right\|^2 \;\middle|\; C^{(t)} = 1\right] + (1 - b^{(t)})\,\mathbb{E}\left[\left\|\mathfrak{e}^{(t)}\right\|^2 \;\middle|\; C^{(t)} = 0\right] \tag{28}$$

Clearly,

$$\mathbb{E}\left[\left\|\mathfrak{e}^{(t)}\right\|^2 \;\middle|\; C^{(t)} = 1\right] \leq \frac{\sigma_1^2}{n} = \frac{\epsilon^2}{5} \tag{29}$$

Moreover, conditional on $C^{(t)} = 0$, we have from the definition of the gradient estimator that

$$\mathfrak{e}^{(t)} = \mathfrak{e}^{(t-1)} + \psi^{(t)}$$

where

$$\psi^{(t)} \coloneqq \sum_{k=1}^{K^{(t)}} \widehat{\nabla^2 F}(x^{(t,k-1)}, z^{(t,k)})\left(x^{(t,k)} - x^{(t,k-1)}\right) - \nabla F(x^{(t)}) + \nabla F(x^{(t-1)})$$

and

$$K^{(t)} = \left\lceil \frac{5\left(\sigma_2^2 + L_2\epsilon\right)}{b^{(t)}\epsilon^2} \cdot \|x^{(t)} - x^{(t-1)}\|^2 \right\rceil \tag{30}$$

where $x^{(t,k)}$ and $x^{(t,k)}$ respectively denote the values of $x^{(k)}$ and $z^{(k)}$ (defined on Line 8) during the t^{th} call to Algorithm 1.

We may therefore decompose the error conditional on $C^{(t)} = 0$ as

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathfrak{e}^{(t)}\right\|^2 \;\middle|\; C^{(t)} = 0\right] &\overset{(i)}{=} \mathbb{E}\left\|\mathfrak{e}^{(t-1)} + \mathbb{E}\left[\psi^{(t)} \mid \mathcal{G}^{(t)}\right]\right\|^2 + \mathbb{E}\left\|\psi^{(t)} - \mathbb{E}\left[\psi^{(t)} \mid \mathcal{G}^{(t)}\right]\right\|^2 \\
&\overset{(ii)}{\leq} \mathbb{E}\left[\left(1 + \frac{b^{(t)}}{2}\right)\left\|\mathfrak{e}^{(t-1)}\right\|^2\right] + \mathbb{E}\left[\left(1 + \frac{2}{b^{(t)}}\right)\left\|\mathbb{E}\left[\psi^{(t)} \mid \mathcal{G}^{(t)}\right]\right\|^2\right] + \mathbb{E}\left\|\psi^{(t)} - \mathbb{E}\left[\psi^{(t)} \mid \mathcal{G}^{(t)}\right]\right\|^2
\end{aligned} \tag{31}$$

where (i) is due to $\mathfrak{e}^{(t-1)} \in \mathcal{G}^{(t)}$ and (ii) is due to Young's inequality.

The facts that $z^{(t,k)}$ is independent from $\mathcal{G}^{(t)}$, that $\nabla F(x^{(t)}) - \nabla F(x^{(t-1)}) \in \mathcal{G}^{(t)}$, and that $\widehat{\nabla^2 F}(\cdot)$ is unbiased give

$$\mathbb{E}\Big[\psi^{(t)} \;\Big|\; \mathcal{G}^{(t)}\Big] = \sum_{k=1}^{K^{(t)}} \nabla^2 F(x^{(t,k-1)})\Big(x^{(t,k)} - x^{(t,k-1)}\Big) - \nabla F(x^{(t)}) + \nabla F(x^{(t-1)})$$

for every t. Consequently, the scaling (30) and Hessian estimator variance bound imply

$$\begin{aligned}
&\mathbb{E}\Bigg[\Big\|\psi^{(t)} - \mathbb{E}\Big[\psi^{(t)} \mid \mathcal{G}^{(t)}\Big]\Big\|^2 \;\Big|\; \mathcal{G}^{(t)}\Bigg] \\
&\quad\overset{(\star)}{=} \frac{1}{(K^{(t)})^2}\sum_{k=1}^{K^{(t)}} \mathbb{E}\Bigg[\Big\|(\widehat{\nabla^2 F}(x^{(t,k-1)}, z^{(t,k)}) - \nabla^2 F(x^{(t,k-1)}))(x^{(t)} - x^{(t-1)})\Big\|^2 \;\Big|\; \mathcal{G}^{(t)}\Bigg] \\
&\quad\le \frac{1}{(K^{(t)})^2}\sum_{k=1}^{K^{(t)}} \mathbb{E}\Bigg[\Big\|\widehat{\nabla^2 F}(x^{(t,k-1)}, z^{(t,k)}) - \nabla^2 F(x^{(t,k-1)})\Big\|_{\mathrm{op}}^2 \;\Big|\; \mathcal{G}^{(t)}\Bigg]\|x^{(t)} - x^{(t-1)}\|^2 \\
&\quad\le \sigma_2^2 \cdot \frac{\|x^{(t)} - x^{(t-1)}\|^2}{K^{(t)}} \le b^{(t)} \cdot \frac{\epsilon^2}{5}
\end{aligned} \tag{32}$$

where the equality $(\star)$ above is due to the fact that $z^{(t,1)}, \ldots, z^{(t,K^{(t)})}$ are i.i.d., as well as $x^{(t,k)} - x^{(t,k-1)} = \frac{1}{K^{(t)}}(x^{(t)} - x^{(t-1)})$.

Next, we observe that Taylor's theorem and fact that F has L_2-Lipschitz Hessian implies that $\|\nabla F(x') - \nabla F(x) - \nabla^2(x)F(x' - x)\| \le \frac{L_2}{2}\|x' - x\|^2$ for all $x, x' \in \mathbb{R}^d$. Therefore,

$$\begin{aligned}
\Big\|\mathbb{E}\Big[\psi^{(t)} \;\Big|\; \mathcal{G}^{(t)}\Big]\Big\| &= \Bigg\|\sum_{k=1}^{K^{(t)}} \nabla F(x^{(t,k)}) - \nabla F(x^{(t,k-1)}) - \nabla^2 F(x^{(t,k-1)})\Big(x^{(t,k)} - x^{(t,k-1)}\Big)\Bigg\| \\
&\le \sum_{k=1}^{K^{(t)}} \Big\|\nabla F(x^{(t,k)}) - \nabla F(x^{(t,k-1)}) - \nabla^2 F(x^{(t,k-1)})\Big(x^{(t,k)} - x^{(t,k-1)}\Big)\Big\| \\
&\le K^{(t)} \cdot \frac{L_2}{2} \cdot \left(\frac{\|x^{(t)} - x^{(t-1)}\|}{K^{(t)}}\right)^2 \le b^{(t)} \cdot \frac{\epsilon}{50}
\end{aligned} \tag{33}$$

where we used (30) again.

Substituting back through equations (33), (32), (31), (29) and (28), we have

$$\begin{aligned}
\mathbb{E}\big\|\mathfrak{e}^{(t)}\big\|^2 &\le \mathbb{E}\left[b^{(t)} \cdot \frac{\epsilon^2}{5} + (1 - b^{(t)})\left((1 + \frac{b^{(t)}}{2})\big\|\mathfrak{e}^{(t-1)}\big\|^2 + (1 + \frac{2}{b^{(t)}})(\frac{b^{(t)}\epsilon}{50})^2 + b^{(t)} \cdot \frac{\epsilon^2}{5}\right)\right] \\
&\le \left(1 - \frac{\mathbb{E}[b^{(t)}]}{2}\right)\mathbb{E}\left\|g^{(t-1)} - \nabla F(x^{(t-1)})\right\|^2 + \frac{\mathbb{E}[b^{(t)}]}{2}\epsilon^2 \le \epsilon^2
\end{aligned}$$

as required; the second inequality follows from algebraic manipulation and the fact that $\mathfrak{e}^{(t-1)}$ is independent of $b^{(t)}$ by assumption. □

The following lemma bounds the number of oracle queries made per call to the gradient estimator.

Lemma 5. *The expected number of stochastic oracle queries made by* SPIDER-RVR-Gradient-Estimator *when called a single time with arguments* $(x,\, x_{\text{prev}}, g_{\text{prev}})$ *and parameters* (ϵ, b) *is at most*

$$6\left(1 + \frac{b\sigma_1^2}{\epsilon^2} + \frac{(\sigma_2^2 + L_2\epsilon)\cdot\|x - x_{\text{prev}}\|^2}{b\epsilon^2}\right)$$

Proof. Let m denote the number of oracle calls made by the gradient estimator when invoked with arguments $(x,\, x_{\text{prev}}, g_{\text{prev}})$. For any call to the estimator, there are two cases, either (a) $C = 1$, or (b) $C = 0$. In the first case, the gradient estimator queries the oracle n times at the point x and returns the empirical average of the returned stochastic estimates (see Line 5 in Algorithm 1). Thus, $m = n$ for this case. In the second case, the estimator queries the oracle once for each point in the set $\left(x^{(k-1)}\right)_{k=1}^{K}$, and updates the gradient using a stochastic path integral as in Line 8. Thus, $m = K$ for this case.

Combining the two cases, using $C \sim \text{Bernoulli}(b)$ and substituting in the values of n and K, we get

$$\begin{aligned}\mathbb{E}[m] &= \Pr(C=1)\,\mathbb{E}[m \mid C=1] + \Pr(C=0)\,\mathbb{E}[m \mid C=0] = \mathbb{E}[b\cdot n + (1-b)\cdot K] \\ &= \left\lceil \frac{5b\sigma_1^2}{\epsilon^2} \right\rceil + \left\lceil \frac{5(\sigma_2^2 + L_2\epsilon)\cdot\|x - x_{\text{prev}}\|^2}{b\epsilon^2} \right\rceil \le 6\left(\frac{b\sigma_1^2}{\epsilon^2} + \frac{(\sigma_2^2 + L_2\epsilon)\cdot\|x - x_{\text{prev}}\|^2}{b\epsilon^2} + 1\right)\end{aligned}$$

where the final inequality follows from $\lceil x \rceil \le x + 1$. □

4.2 Proof of Theorem 2

In the following, we first show that Algorithm 2 returns a point $\widehat{x}$ such that, $\mathbb{E}[\|\nabla F(\widehat{x})\|] \le 32\epsilon$. We then bound the expected number of oracle queries used throughout the execution.[7] The following lemma characterizes the effect of gradient descent update step used by Algorithm 2 and later in Algorithm 4.

Lemma 6. *Given a function* $F \in \mathcal{F}_2(\Delta, L_1, \infty)$*, a point* x*, and gradient estimator* g *at* x*, define*

$$y := x - \eta g$$

Then, for any $\eta \le \frac{1}{2L_1}$*, the point* y *satisfies*

$$F(x) - F(y) \ge \frac{\eta}{8}\|\nabla F(x)\|^2 - \frac{3\eta}{4}\|\nabla F(x) - g\|^2$$

We are ready for the

Proof of Theorem 2. Since, $\eta = \frac{1}{2\sqrt{L_1^2 + \bar{\sigma}_2^2 + \bar{\epsilon}L_2}} \le \frac{1}{2L_1}$ and F has L_1-Lipschitz gradient, Lemma 6 implies that the point $x^{(t+1)}$ computed using the update rule $x^{(t+1)} \leftarrow x^{(t)} - \eta g^{(t)}$ satisfies

$$\frac{\eta}{8}\left\|\nabla F(x^{(t)})\right\|^2 \le F(x^{(t)}) - F(x^{(t+1)}) + \frac{3\eta}{4}\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2$$

[7] In the proof, we show convergence to a 32ϵ-stationary point. A simple change of variable, i.e. running Algorithm 2 with $\epsilon \leftarrow \frac{\epsilon}{32}$, returns a point $\widehat{x}$ that enjoys the guarantee that $\|\nabla F(\hat{x})\| \le \epsilon$.

Telescoping the above from t from 0 to $T-1$, this implies

$$\frac{\eta}{8}\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)})\right\|^2 \le F(x^{(0)}) - F(x^{(T)}) + \frac{3\eta}{4}\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2$$
$$\le \Delta + \frac{3\eta}{4}\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2$$

where the last inequality follows from the fact that $F(x^{(0)}) - F(x^{(T)}) \le \Delta$. Next, taking expectation on both the sides—with respect to the stochasticity of the oracle and the algorithm's internal randomization—we get

$$\frac{\eta}{8}\,\mathbb{E}\left[\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)})\right\|^2\right] \le \Delta + \frac{3\eta}{4}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2\right]$$

Using Lemma 4, we have $\mathbb{E}\Big[\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2\Big] \le \epsilon^2$ for all $t \ge 1$. Dividing both the sides by $\frac{\eta T}{8}$, and plugging in the value of the parameters T and η, we get

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)})\right\|^2\right] \le \frac{8\Delta}{\eta T} + \frac{6}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(x^{(t)}) - g^{(t)}\right\|^2\right] \le \frac{8\Delta}{\eta T} + 6\epsilon^2 \le 14\epsilon^2 \tag{34}$$

Thus, for $\widehat{x}$ chosen uniformly at random from the set $\left(x^{(t)}\right)_{t=1}^{T}$, we have

$$\mathbb{E}\|\nabla F(\widehat{x})\|^2 = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla F(x^{(t)})\right\|^2 = \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla F(x^{(t)})\right\|^2\right] \le 14\epsilon^2$$

Finally, Markov's inequality implies that with probability at least $\frac{7}{8}$,

$$\|\nabla F(\widehat{x})\| \le 32\epsilon \tag{35}$$

The proof is completed by carefully bounding the oracle queries number, as is shown immediately. □

Bound on the number of oracle queries. Algorithm 2 queries the stochastic oracle in only when it invokes SPIDER-RVR in Line 4 to compute the gradient estimate $g^{(t)}$ at time t. Let M denote the total number of oracle calls made up until time T. Invoking Lemma 5 to bound the expected number of stochastic oracle calls for each $t \ge 1$, and ignoring all the mutiplicative constants, we get

$$\mathbb{E}[M] \le 5\sum_{t=1}^{T}\mathbb{E}\left[\frac{b\sigma_1^2}{\epsilon^2} + \frac{\left\|x^{(t+1)} - x^{(t)}\right\|^2 \cdot \left(\sigma_2^2 + \epsilon L_2\right)}{b\epsilon^2} + 1\right]$$
$$\overset{(i)}{\le} O\left(\sum_{t=1}^{T}\mathbb{E}\left[\frac{b\sigma_1^2}{\epsilon^2} + \frac{\left\|\eta g^{(t)}\right\|^2 \cdot \left(\sigma_2^2 + \epsilon L_2\right)}{b\epsilon^2} + 1\right]\right)$$
$$\overset{(ii)}{\le} O\left(\frac{\Delta}{\eta\epsilon^2}\cdot\left(\frac{b\sigma_1^2}{\epsilon^2} + \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|g^{(t)}\right\|^2\right]\cdot\frac{\eta^2\left(\sigma_2^2 + \epsilon L_2\right)}{b\epsilon^2} + 1\right)\right)$$

$$\overset{(iii)}{=} O\left(\frac{\Delta}{\eta\epsilon^2}\cdot\left(\frac{b\sigma_1^2}{\epsilon^2}+\frac{\eta^2\left(\sigma_2^2+\epsilon L_2\right)}{b}+1\right)\right) \tag{36}$$

where (i) is given by plugging in the update rule from Line 5 and by dropping multiplicative constants, (ii) is given by rearranging the terms, plugging in the value of T and using that $T \geq 1$ (to simplify the ceiling operator) under the assumption $\epsilon \leq \sqrt{\Delta L_1}$, and (iii) follows by observing that

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|g^{(t)}\right\|^2\right] \leq 2\,\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left\|g^{(t)}-\nabla F(x^{(t)})\right\|^2+\frac{1}{T}\sum_{t=1}^{T}\left\|\nabla F(x^{(t)})\right\|^2\right] \leq 30\epsilon^2$$

as a consequence of Lemma 4 and the bound in (34). Next, note that since we assume $\epsilon < \sigma_1$, and since we have $\eta \leq \frac{1}{2\sqrt{\sigma_2^2+\epsilon L_2}}$, the parameter b is equal to $\frac{\eta\epsilon\sqrt{\sigma_2^2+\epsilon L_2}}{\sigma_1}$ (as this is smaller than 1). Thus, plugging the value of b and η in the bound (36), we get

$$\begin{aligned}\mathbb{E}[m(T)] &= O\left(\frac{\Delta\sigma_1\sqrt{\sigma_2^2+\epsilon L_2}}{\epsilon^3}+\frac{\Delta\sqrt{L_1^2+\sigma_2^2+\epsilon L_2}}{\epsilon^2}\right)\\ &= O\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3}+\frac{\Delta\sigma_1\sqrt{L_2}}{\epsilon^{2.5}}+\frac{\Delta\sigma_2}{\epsilon^2}+\frac{\Delta L_1}{\epsilon^2}+\frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}}\right)\end{aligned}$$

Using Markov's inequality, we have that with probability at least $\frac{7}{8}$,

$$M \leq O\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3}+\frac{\Delta\sigma_1\sqrt{L_2}}{\epsilon^{2.5}}+\frac{\Delta\sigma_2}{\epsilon^2}+\frac{\Delta L_1}{\epsilon^2}+\frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}}\right) \tag{37}$$

The final statement follows by taking a union bound with failure probabilities for (35) and (37).

4.3 Proof of Theorem 3

In the following, we first show that Algorithm 3 returns a point $\hat{x}$, such that with probability at least $\frac{7}{8}$, $\|\nabla F(\hat{x})\| \leq 350\epsilon$. We then bound, with probability at least $\frac{7}{8}$, the total number of oracle queries made up until time T.

The following lemma is a standard result which bounds the expected error for the empirical Hessian.

Lemma 7. *Given a function $F \in \mathcal{F}_2(\Delta, \infty, L_2)$, a stochastic oracle in $\mathcal{O}_2(F, \sigma_{1:2})$ and a point x, let $H := \frac{1}{m}\sum_{i=1}^{m}\widehat{\nabla^2 F}(x, z^{(i)})$ denote the empirical Hessian at the point x estimated using m stochastic queries at x, where $z^{(i)} \overset{\text{i.i.d.}}{\sim} P_z$. Then*

$$\mathbb{E}\left[\left\|H-\nabla^2 F(x)\right\|_{\text{op}}^2\right] \leq \frac{22\sigma_2^2\log(d)}{m}$$

We also have the following descent lemma for cubic-regularized trust-region method

Proposition 2. *Given a function $F \in \mathcal{F}_2(\Delta, \infty, L_2)$, let gradient estimator $g \in \mathbb{R}^d$ and hessian estimator $H \in \mathbb{S}^d$ be random variables, define*

$$m_x(y) = F(x) + \langle g, y-x\rangle + \frac{H}{2}[y-x, y-x] + \frac{M}{6}\|y-x\|^3$$

and let $y \in \arg\min_{z \in \mathbb{B}_\eta(x)} m_x(z)$. *Then, for any* $M \geq 4L_2$ *and* $\eta \geq 0$, *the random variable* y *satisfies*

$$\mathbb{E}[F(x) - F(y)] \geq \frac{M\eta^3}{60} \Pr\left(\|\nabla F(y)\| \geq \frac{M\eta^2}{2}\right) - \frac{9}{\sqrt{M}} \cdot \mathbb{E}\Big[\|\nabla F(x) - g\|^2\Big]^{\frac{3}{4}} - \frac{5\eta^{\frac{3}{2}}}{\sqrt{M}} \cdot \mathbb{E}\Big[\big\|\nabla^2 F(x) - H\big\|_{\mathrm{op}}^2\Big]^{\frac{3}{4}}$$

where $\Pr(\cdot)$ *and* $\mathbb{E}[\cdot]$ *are taken with respect to the randomness over* H *and* g.

We are ready for

Proof of Theorem 3. Note that, using Lemma 4 and Lemma 7, we have for all $t \geq 0$,

$$\mathbb{E}\Big[\|\nabla F\Big(x^{(t)}\Big) - g^{(t)}\|\Big] \leq \epsilon^2 \quad \text{and} \quad \mathbb{E}\Big[\|\nabla^2 F\Big(x^{(t)}\Big) - H^{(t)}\|_{\mathrm{op}}\Big] \leq \frac{\epsilon^2}{\eta^2} \tag{38}$$

Thus, for each $t \geq 1$, invoking Proposition 2 and plugging in the bounds from (38), and using the value of η, we get

$$\begin{aligned}\mathbb{E}\Big[F(x^{(t)}) - F(x^{(t+1)})\Big] &\geq \frac{M\eta^3}{60} \Pr\left(\Big\|\nabla F(x^{(t+1)})\Big\| \geq \frac{M\eta^2}{2}\right) - \frac{14\epsilon^{\frac{3}{2}}}{\sqrt{M}} \\ &\geq \frac{240\epsilon^{\frac{3}{2}}}{\sqrt{M}} \left(\Pr\left(\Big\|\nabla F(x^{(t+1)})\Big\| \geq 350\epsilon\right) - \frac{1}{16}\right)\end{aligned}$$

Telescoping this inequality from $t = 1$ to T, we have that

$$\begin{aligned}\mathbb{E}\Big[F(x^{(1)}) - F(x^{(T+1)})\Big] &\geq \frac{240\epsilon^{\frac{3}{2}}}{\sqrt{M}} \cdot T \cdot \left(\frac{1}{T}\sum_{t=1}^{T} \Pr\left(\Big\|\nabla F(x^{(t+1)})\Big\| \geq 350\epsilon\right) - \frac{1}{16}\right) \\ &= \frac{240\epsilon^{\frac{3}{2}}}{\sqrt{M}} \cdot T \cdot \left(\Pr(\|\nabla F(\widehat{x})\| \geq 350\epsilon) - \frac{1}{16}\right)\end{aligned}$$

where the equality follows because $\widehat{x}$ is sampled uniformly at random from the set $\big\{x^{(t)}\big\}_{t=2}^{T+1}$. Next, using the fact that, $F(x^{(t)}) - F\big(x^{(T+1)}\big) \leq \Delta$, rearranging the terms, and plugging in the value of T, we get

$$\Pr(\|\nabla F(\widehat{x})\| \geq 350\epsilon) \leq \frac{\Delta\sqrt{M}}{240\epsilon^{\frac{3}{2}}T} + \frac{1}{16} \leq \frac{1}{8}$$

Thus, with probability at least $\frac{7}{8}$,

$$\|\nabla F(\widehat{x})\| \leq 350\epsilon \tag{39}$$

Likewise, we complete the proof by carefully bounding the oracle queries number. □

Bound on the number of oracle queries. Algorithm 3 queries the stochastic oracle in Line 5 and Line 6 only to compute the respective Hessian and gradient estimates. Let M_h and M_g denote the total number of stochastic oracle queries made by Line 5 and Line 6 till time T respectively. Further, Let $M = M_h + M_g$ denote the total number of oracle queries made till time T.

In what follows, we first bound $\mathbb{E}[M_h]$ and $\mathbb{E}[M_g]$. Then, we invoke Markov's inequality to deduce that the desired bound on M holds with probability at least $\frac{7}{8}$.

(i) **Bound on $\mathbb{E}[M_h]$.** Since the algorithm queries the stochastic Hessian oracle n_H times per iteration, $M_h = T \cdot n_H$. Plugging the values of T, n_H and M as specified in Algorithm 3, and ignoring multiplicative constant, we get

$$\begin{aligned}\mathbb{E}[M_h] &= \left\lceil \frac{5\Delta\sqrt{M}}{3\epsilon^{1.5}} \right\rceil \cdot \left\lceil \frac{22\sigma_2^2\eta^2\log(d)}{\epsilon^2} \right\rceil \\ &\leq O\left(\frac{\Delta\sqrt{M}}{\epsilon^{1.5}} + \frac{\Delta\sigma_2^2\log(d)}{\epsilon^{2.5}\sqrt{M}} \right) \\ &\leq O\left(\frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}} + \frac{\Delta\sigma_2}{\epsilon^2} + \frac{\Delta\sigma_1\sigma_2\sqrt{\log(d)}}{\epsilon^3} \right) \end{aligned} \tag{40}$$

where the first inequality above follows from the fact that $\frac{\Delta\sqrt{M}}{\epsilon^{1.5}} \geq 1$ under the natural choice for the precision parameter $\epsilon \leq \Delta^{\frac{2}{3}} M^{\frac{1}{3}}$ and using the identity $\lceil x \rceil \leq x + 1$ for $x \geq 0$.

(ii) **Bound on $\mathbb{E}[M_g]$.** Invoking Lemma 5 for each $t \geq 1$, we get

$$\begin{aligned}\mathbb{E}[M_g] &= 6\sum_{t=1}^{T} \mathbb{E}\left[\frac{b\sigma_1^2}{\epsilon^2} + \frac{(\sigma_2^2 + L_2\epsilon) \cdot \left\| x^{(t)} - x^{(t-1)} \right\|^2}{b\epsilon^2} + 1 \right] \\ &\overset{(i)}{=} O\left(T \cdot \left(\frac{b\sigma_1^2}{\epsilon^2} + \frac{(\sigma_2^2 + L_2\epsilon)\cdot\eta^2}{b\epsilon^2} + 1 \right) \right) \\ &\overset{(ii)}{=} O\left(\frac{\Delta}{\eta\epsilon} \cdot \left(\frac{b\sigma_1^2}{\epsilon^2} + \frac{(\sigma_2^2 + L_2\epsilon)\cdot\eta^2}{b\epsilon^2} + 1 \right) \right) \end{aligned} \tag{41}$$

where (i) follows by observing $\left\| x^{(t)} - x^{(t-1)} \right\| \leq \eta$ due to the update rule in Line 7 and (ii) is given by plugging in the value of $T \leq O(\frac{\Delta}{\eta\epsilon})$ for the natural choice of parameter $\epsilon = O(\Delta^{\frac{2}{3}} M^{\frac{1}{3}})$. Next, note that since $M > L_2$, and since we assume $\epsilon < \sigma_1$, the parameter b is equal to $\frac{\eta\sqrt{\sigma_2^2 + \epsilon L_2}}{25\sigma_1}$ (which is smaller than 1). Thus, plugging the value of b and η in the bound (41), we get

$$\begin{aligned}\mathbb{E}[M_g] &= O\left(\frac{\Delta\sigma_1\sqrt{\sigma_2^2 + \epsilon L_2}}{\epsilon^3} + \frac{\Delta\sqrt{M}}{\epsilon^{1.5}} \right) \\ &= O\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3} + \frac{\Delta\sigma_1\sqrt{L_2}}{\epsilon^{2.5}} + \frac{\Delta\sigma_2}{\epsilon^2}\sqrt{\log(d)} + \frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}} \right) \end{aligned} \tag{42}$$

where the second equality follows by using that $\epsilon \leq \sigma_1$ to simplify the term $\frac{\Delta\sqrt{M}}{\epsilon^{1.5}}$.

Algorithm 4 Stochastic gradient descent with negative curvature search and SPIDER-RVR

Require: Oracle $(\mathsf{O}_F^2, P_z) \in \overline{\mathcal{O}}_2(F, \sigma_1, \bar{\sigma}_2)$ for $F \in \mathcal{F}_2(\Delta, L_1, L_2)$. Precision parameters ϵ, δ.

Set $\eta = \min\left\{ \frac{\delta}{\epsilon L_2}, \frac{1}{2\sqrt{L_1^2+\bar{\sigma}_2^2+\epsilon L_2}} \right\}$, $T = \left\lceil \frac{20\Delta L_2^2}{\delta^3} + \frac{2\Delta}{\eta\epsilon^2} \right\rceil$, $p = \frac{\delta^3}{\delta^3+10\Delta L_2^2\eta\epsilon^2}$, $\gamma = \frac{\delta}{40^2 L_2}$.

Set $b_g = \min\{1, \frac{\eta\epsilon\sqrt{\bar{\sigma}_2^2+\epsilon L_2}}{\sigma_1}\}$ and $b_H = \min\{1, \frac{\delta\sqrt{\bar{\sigma}_2^2+\epsilon L_2}}{\sigma_1 L_2}\}$.

Initialize $x^{(0)}, x^{(1)} \leftarrow 0$, $g^{(1)} \leftarrow \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b_g}(x^{(1)}, x^{(0)}, \perp)$.

for $t = 1$ to T **do**

  Sample $Q_t \sim \text{Bernoulli}(p)$.

  if $Q_t = 1$ **then**

    $x^{(t+1)} \leftarrow x^{(t)} - \eta \cdot g^{(t)}$.

    $g^{(t+1)} \leftarrow \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b_g}(x^{(t+1)}, x^{(t)}, g^{(t)})$.

  else

    $u^{(t)} \leftarrow \mathsf{Oja}\big(x^{(t)}, \mathsf{O}_F^2, 2\delta, \gamma\big)$. // Oja's algorithm

    if $u^{(t)} \equiv \perp$ **then**

      $x^{(t+1)} \leftarrow x^{(t)}$.

      $g^{(t+1)} \leftarrow g^{(t)}$.

    else

      Sample $r^{(t)} \sim \text{Uniform}(\{-1, 1\})$.

      $x^{(t+1)} \leftarrow x^{(t)} + \frac{\delta}{L_2} \cdot r^{(t)} \cdot u^{(t)}$.

      $g^{(t+1)} \leftarrow \mathsf{SPIDER\text{-}RVR\text{-}Gradient\text{-}Estimator}_{\epsilon,b_H}(x^{(t+1)}, x^{(t)}, g^{(t)})$.

    end if

  end if

end for

return $\widehat{x}$ chosen uniformly at random from $\big(x^{(t)}\big)_{t=0}^{T-1}$.

Adding (42) and (40), the total number of oracle queries made by Algorithm 3 till time T is bounded, in expectation, by

$$\mathbb{E}[M] = \mathbb{E}[M_g + M_h] = O\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3}\sqrt{\log(d)} + \frac{\Delta\sigma_1\sqrt{L_2}}{\epsilon^{2.5}} + \frac{\Delta\sigma_2}{\epsilon^2}\sqrt{\log(d)} + \frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}} \right)$$

Using Markov's inequality, we get that, with probability at least $\frac{7}{8}$,

$$M \le O\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3}\sqrt{\log(d)} + \frac{\Delta\sigma_1\sqrt{L_2}}{\epsilon^{2.5}} + \frac{\Delta\sigma_2}{\epsilon^2}\sqrt{\log(d)} + \frac{\Delta\sqrt{L_2}}{\epsilon^{1.5}} \right) \tag{43}$$

The final statement follows by taking a union bound for the failure probability of (39) and (43).

5 Complexity of finding second-order stationary points: Lower bounds

Having established rates of convergence lower and upper bounds for finding ϵ-stationary points, we now turn our attention to (ϵ, δ)-second order stationary points, which have the additional requirement that $\lambda_{\min}(\nabla^2 F(x)) \ge -\delta$, i.e. that F is δ-weakly convex around x. This and next sections follows the general organization of the prequel: we first develop improved lower bounds

that apply to a broad class of algorithms, and then design and analyze an algorithm with nearly-matching upper bounds.

In this section, we develop lower complexity bounds for the task of finding (ϵ, δ)-stationary points. To do so, we prove new lower bounds for the simpler sub-problem of finding a *δ-weakly convex* point, i.e., a point x such that $\lambda_{\min}(\nabla^2 F(x)) \geq -\delta$ (with no restriction on $\|\nabla F(x)\|$). Lower bounds for finding (ϵ, δ)-SOSPs follow as the maximum (or, equivalently, the sum) of lower bounds we develop here and the lower bounds for finding ε-stationary points given in Theorem 4. To see why this is so, let F_ϵ and F_δ be hard instances for finding ϵ-stationary and δ-weakly-convex points respectively, and consider the "direct sum" $F_{\epsilon,\delta}(x) := \frac{1}{2}F_\epsilon(x_1, \ldots, x_d) + \frac{1}{2}F_\delta(x_{d+1}, \ldots, x_{2d})$; this is a hard instance for finding (ϵ, δ)-SOSPs that inherits all the regularity properties of its constituent functions.

The basic construction we use here is a modification of the zero-chain introduced in [CDHS20] (see (15) in Section 3.1) in which large $\lambda_{\min}(\nabla^2 F(x))$ is possible only when essentially none of the entries of x is zero. Given $T > 0$, we define the hard function

$$G_T(x) := \Psi(1)\Lambda(x_1) + \sum_{i=2}^{T} \left[\Psi(-x_{i-1})\Lambda(-x_i) + \Psi(x_{i-1})\Lambda(x_i)\right] \tag{44}$$

where $\Psi(x) := \exp(1 - \frac{1}{(2x-1)^2})\mathbb{1}_{x>\frac{1}{2}}$ (as in [CDHS20]) and $\Lambda(x) := 8(e^{\frac{-x^2}{2}} - 1)$.

Our design for the function Λ guarantees that any query whose last coordinate is zero has significant negative curvature, while maintaining the original chain structure which guarantees that zero-respecting algorithms require many queries before "discovering" the last coordinate. We complete the construction by specifying a collection of stochastic derivative estimators similar to those in Section 5, except for that we choose the stochastic gradient estimator $\widehat{\nabla G_T}$ to be exactly equal to ∇G_T, so that the lower bound holds even for $\sigma_1 = 0$; Appropriately scaling G_T allows us to tune the Lipschitz constants of its derivatives and the variance of the estimators, thereby establishing the following complexity bounds (see Appendix B for a full derivation).

Theorem 4. *Let $p \geq 2$ and $\Delta, L_{1:p}, \sigma_{1:p} > 0$ be fixed. If $\delta \leq O(\min\{\sigma_2, L_1\})$, then there exists $F \in \mathcal{F}_p(\Delta, L_{1:p})$ and $(\mathsf{O}_F^p, P_z) \in \mathcal{O}_p(F, \sigma_{1:p})$ such that for any stochastic pth-order zero-respecting algorithm, the number of queries to O_F^p required to obtain a δ-weakly convex point with constant probability is at least*

$$\Omega(1) \cdot \begin{cases} \frac{\Delta\sigma_2^2 L_2^2}{\delta^5} & p = 2 \\ \frac{\Delta\sigma_2^2}{\delta^3} \min\left\{ \min_{q\in\{3,\ldots,p\}} \left(\frac{\sigma_q}{\sigma_2}\right)^{\frac{2}{q-2}}, \min_{q'\in\{2,\ldots,p\}} \left(\frac{L_{q'}}{\delta}\right)^{\frac{2}{q'-1}} \right\} & p > 2 \end{cases} \tag{45}$$

A construction of dimension $\Theta\Big(\frac{\Delta}{\delta} \min\Big\{ \min_{q\in\{3,\ldots,p\}} \left(\frac{\sigma_q}{\sigma_2}\right)^{\frac{2}{q-2}}, \min_{q'\in\{2,\ldots,p\}} \left(\frac{L_{q'}}{\delta}\right)^{\frac{2}{q'-1}} \Big\}\Big)$ realizes the lower bound.

Theorem 4 is new even in the noiseless case (in which $\sigma_1 = \cdots = \sigma_p = 0$), where it specializes to

$$\Omega(1) \cdot \frac{\Delta}{\delta} \min_{q\in\{2,\ldots,p\}} \left(\frac{L_q}{\delta}\right)^{\frac{2}{q-1}} \tag{46}$$

For the class $\mathcal{F}_p(\Delta, L_p)$, the lower bound (46) further simplifies to $\Delta L_p^{\frac{2}{p-1}} \delta^{-\frac{p+1}{p-1}}$, which is attained by the pth-order regularization method given in [CGT17, Theorem 3.6]. Together, these results

characterize the deterministic complexity of finding δ-weakly convex points with noiseless pth-order methods.

Returning to the stochastic setting, the bound in Theorem 4, when combined with Theorem 1, implies the following oracle complexity lower bound bound for finding (ε, δ)-SOSPs with zero-respecting stochastic second-order methods ($p = 2$):

$$\Omega(1) \cdot \left(\min\left\{ \frac{\Delta\sigma_1\sigma_2}{\epsilon^3}, \frac{\Delta L_2^{0.5}\sigma_1}{\epsilon^{3.5}}, \frac{\Delta L_1\sigma_1^2}{\epsilon^4} \right\} + \frac{\Delta\sigma_2^2 L_2^2}{\delta^5} \right) \tag{47}$$

Our lower bound matches the $\epsilon^{-3}+\delta^{-5}$ terms in the upper bound given by Theorem 5, but does not match the mixed term $\epsilon^{-2}\delta^{-2}$ appearing in the upper bound.[8] Overall, the rates match whenever $\delta = \Omega(\epsilon^{0.5})$ or $\delta = O(\epsilon^{2/3})$.

Theorem 4 suggest that in the stochastic regime, the rate does not improve beyond δ^{-3} for $p \geq 3$, while the optimal rate in the noiseless regime, $\delta^{-\frac{p+1}{p-1}}$, continues improving for all p.[9] However, we are not yet aware of an algorithm using stochastic third-order information or higher that can achieve the δ^{-3} complexity bound.

Bounding the operator norm of $\nabla_i^p F_T$. Here we complete the proof of Lemma 2 by proving Part (iii). Our proof follows along the lines of the proof of Lemma 3 of [CDHS20]. Let $x \in \mathbb{R}^T$ and $i_1, \ldots, i_p \in [T]$, and note that by the chain-like structure of F_T, $\partial_{i_1} \cdots \partial_{i_p} F_T(x)$ is non-zero if and only if $|i_j - i_k| \leq 1$ for any $j, k \in [p]$. A straightforward calculation yields

$$\begin{aligned} |\partial_{i_1} \cdots \partial_{i_p} F_T(x)| &\leq \max_{i \in [T]} \max_{\gamma \in \{0,1\}^{p-1} \cup \{0,-1\}^{p-1}} |\partial_{i+\gamma_1} \cdots \partial_{i+\gamma_{p-1}} \partial_i F_T(x)| \\ &\leq \max_{k \in [p]} \left\{ 2 \sup_{\xi \in \mathbb{R}} \left| \Psi^k(\xi) \right| \sup_{\xi' \in \mathbb{R}} \left| \Phi^{p-k}(\xi') \right| \right\} \leq \exp(2.5p \log p + 4p + 9) \leq \frac{\ell_{p-1}}{2^{p+1}} \end{aligned} \tag{48}$$

where the penultimate inequality is due to Lemma 1 of [CDHS20]. Therefore, for a fixed $i \in [T]$, we have

$$\begin{aligned} \|\nabla_i^p F_T(x)\|_{\mathrm{op}} &\overset{(a)}{=} \sup_{\|v\|=1} |\langle \nabla_i^p F_T(x), v \rangle| = \sup_{\|v\|=1} \left| \sum_{i_1, \ldots, i_{p-1} \in [T]} \partial_{i_1} \cdots \partial_{i_{p-1}} \partial_i F_T(x) v_{i_1} \cdots v_{i_{p-1}} \right| \\ &\overset{(b)}{\leq} \sum_{\gamma \in \{0,1\}^{p-1} \cup \{0,-1\}^{p-1}} |\partial_{i+\gamma_1} \cdots \partial_{i+\gamma_{p-1}} \partial_i F_T(x)| \overset{(c)}{\leq} (2^p - 1) \frac{\ell_{p-1}}{2^{p+1}} \leq \ell_{p-1} \end{aligned}$$

where (a) follows from the definition of the operator norm, (b) follows by the chain-like structure of F_T, and (c) follows from (48), concluding the proof.

6 Complexity of finding second-order stationary points: Upper bounds

Having described our developments for the task of finding ϵ-approximate first-order stationary points (satisfying (1) with $\delta = L_1$), we subsequently extend in this section our results to general δ.

[8] Young's inequality only gives $\epsilon^{-3} + \delta^{-5} \geq \Omega(\epsilon^{-9/5}\delta^{-2})$.

[9] Indeed, when high-order noise moments are assumed finite, the term $\min_{q \in \{3,\ldots,p\}} (\sigma_q/\sigma_2)^{\frac{2}{q-2}}$ can longer be disregarded. This, in turn, implies that for sufficiently small δ, one cannot improve over δ^{-3}-scaling, as seen by (45).

To be specific, we present in this section an algorithm that enjoys improved complexity for finding (ϵ, δ)-second-order stationary points that nearly matching the lower bound Theorem 4, and that achieves this using only stochastic gradient and Hessian-vector product queries.

To guarantee second-order stationarity, we follow the established technique of interleaving an algorithm for finding a first-order stationary point with negative curvature descent [CDHS17, AZ18a, FLLZ18]. However, we employ a randomized variant of this approach. Specifically, at every iteration we flip a biased coin to determine whether to perform a stochastic gradient step or a stochastic negative curvature descent step. Our algorithm estimates stochastic gradients using the SPIDER-RVR scheme (Algorithm 1), where the value of the restart probability b depends on the type of the previous step (gradient or negative curvature). To implement negative curvature descent, we apply Oja's method [Oja82, AZL17] which detects directions of negative curvature using only stochastic Hessian-vector product queries. For technical reasons pertaining to the analysis of Oja's method, we require the stochastic Hessians to be bounded almost surely, i.e., $\|\widehat{\nabla^2 F}(x,z) - \nabla^2 F(x)\|_{\mathrm{op}} \le \bar{\sigma}_2$ a.s.; we let $\overline{\mathcal{O}}_2(F, \sigma_1, \bar{\sigma}_2)$ denote the class of such bounded noise oracles. Under this assumption, Algorithm 4—whose description is deferred to the Appendix C—enjoys the following convergence guarantee.[10]

Theorem 5. *For any function $F \in \mathcal{F}_2(\Delta, L_{1:2})$, stochastic Hessian-vector product oracle in $\overline{\mathcal{O}}_2(F, \sigma_1, \bar{\sigma}_2)$, $\epsilon \le \min\{\sigma_1, \sqrt{\Delta L_1}\}$, and $\delta \le \min\{\bar{\sigma}_2, L_1, \sqrt{\epsilon L_2}\}$, with probability at least $\frac{5}{8}$ Algorithm 4 returns a point $\widehat{x}$ such that*

$$\|\nabla F(\widehat{x})\| \le \epsilon \quad \text{and} \quad \lambda_{\min}\left(\nabla^2 F(\widehat{x})\right) \ge -\delta$$

and performs at most

$$\widetilde{O}\left(\frac{\Delta\sigma_1\bar{\sigma}_2}{\epsilon^3} + \frac{\Delta L_2\sigma_1\bar{\sigma}_2}{\delta^2\epsilon^2} + \frac{\Delta L_2^2(\bar{\sigma}_2 + L_1)^2}{\delta^5} + \frac{\Delta L_1}{\epsilon^2}\right)$$

stochastic gradient and Hessian-vector product queries.

Similar to the case for finding ϵ-stationary points (see discussion preceding Theorem 3), using full stochastic Hessian information allows us to design an algorithm (Algorithm 5) which removes the dependence on L_1 from the theorem above. Moreover, estimating negative curvature directly from empirical Hessian estimates saves us the need to use Oja's method, which means that we do not need the additional boundedness assumption on the stochastic Hessian used by Algorithm 4. We state its complexity guarantee below.

Theorem 6. *For any function $F \in \mathcal{F}_2(\Delta, \infty, L_2)$, stochastic second order oracle in $\mathcal{O}_2(F, \sigma_1, \sigma_2)$, $\epsilon \le \sigma_1$, and $\delta \le \min\{\sigma_2, \sqrt{\epsilon L_2}, \Delta^{\frac{1}{3}} L_2^{\frac{2}{3}}\}$, with probability at least $\frac{3}{5}$ Algorithm 5 returns a point $\widehat{x}$ such that*

$$\lambda_{\min}\left(\nabla^2 F(\widehat{x})\right) \ge -\delta \qquad \text{and} \quad \|\nabla F(\widehat{x})\| \le \epsilon$$

and performs at most

$$\widetilde{O}\left(\frac{\Delta\sigma_1\sigma_2}{\epsilon^3} + \frac{\Delta L_2\sigma_1\sigma_2}{\delta^2\epsilon^2} + \frac{\Delta L_2^2\sigma_2^2}{\delta^5}\right)$$

stochastic gradient and Hessian queries.

[10] The notation $\widetilde{O}(\cdot)$ hides lower-order terms and logarithmic dependence on the dimension d. See the proof in Appendix C for the complete description of the algorithm and the full complexity bound, including lower order terms.

Method	Uses $\widehat{\nabla^2 F}$?	Complexity bound	Additional assumptions
SGD [GL13]	No	$O(\Delta L_1 \sigma_1^2 \epsilon^{-4})$	
Restarted SGD [FLZ19]	No	$O(\Delta L_2^{0.5} \sigma_1^2 \epsilon^{-3.5})^\dagger$	$\widehat{\nabla F}$ Lipschitz almost surely
Normalized SGD [CM20]	No	$O(\Delta L_2^{0.5} \sigma_1^2 \epsilon^{-3.5})^\dagger$	
Subsampled regularized Newton [TSJ+18]	Yes*	$O(\Delta L_2^{0.5} \sigma_1^2 \epsilon^{-3.5})^\dagger$	
Recursive variance reduction, e.g., [FLLZ18]	No	$O(\Delta \sigma_1 \sigma_{\text{mss}} \epsilon^{-3} + \Delta L_1 \epsilon^{-2})$	Mean-squared smoothness $\sigma_{\text{mss}} \le \sigma_2$, simultaneous queries (Appendix A)
SGD with SPIDER-RVR (Algorithm 2)	Yes*	$O(\Delta \sigma_1 \sigma_2 \epsilon^{-3} + \Delta L_2^{0.5} \sigma_1 \epsilon^{-2.5} + \Delta L_1 \epsilon^{-2})$	
Subsampled Newton with SPIDER-RVR (Algorithm 3)	Yes	$O(\Delta \sigma_1 \sigma_2 \epsilon^{-3} + \Delta L_2^{0.5} \sigma_1 \epsilon^{-2.5} + \Delta \sigma_2 \epsilon^{-2})$	

Table 1. Detailed comparison of guarantees for finding ϵ-stationary points (satisfying $\mathbb{E}\|\nabla F(x)\| \le \epsilon$) for a function F with L_1-Lipschitz gradients and L_2-Lipschitz Hessian. Here Δ is the initial optimality gap, and σ_p is the variance of $\widehat{\nabla^p F}$. Algorithms marked with * require only stochastic Hessian-vector products. Complexity bounds marked with † only show leading order term in ϵ.

7 Further discussions

In summary, this paper provides a fairly complete picture of the worst-case oracle complexity of finding stationary points with a stochastic second-order oracle: for ϵ-stationary points we characterize the leading term in ϵ^{-1} exactly and for (ϵ, δ)-SOSPs we characterize the leading term in δ^{-1} for a wide range of parameters. Nevertheless, our results point to a number of open questions.

Detailed comparison with existing rates. Table 1 provides a detailed comparison between our upper bounds on the complexity of finding ϵ-stationary points and those of prior work.

Benefits of higher-order information for δ-weakly convex points. Our lower and upper bounds (in Theorem 4 and Theorem 6) resolve the optimal rate to find an (ϵ, δ)-stationary point for $p = 2$, i.e., when F is second-order smooth and the algorithm can query stochastic gradient and Hessian information. Furthermore, Theorem 1 shows that higher order information ($p \ge 3$) cannot improve the dependence of the rate on the first-order stationarity parameter ϵ. However, our lower bound for dependence on δ scales as δ^{-5} for $p = 2$, but scales as δ^{-3} for $p \ge 3$. The weaker lower bound for $p \ge 3$ leaves open the possibility of a stronger upper bound using third-order information or higher.

Global methods. For statistical learning and sample average approximation problems, it is natural to consider problem instances of the form $F(x) = \mathbb{E}\big[\widehat{F}(x, z)\big]$. For this setting, a more powerful oracle model is the *global oracle*, in which samples $z^{(1)}, \ldots, z^{(n)}$ are drawn i.i.d. and the

learner observes the entire function $\widehat{F}(\cdot, z^{(t)})$ for each $t \in [n]$. Global oracles are more powerful than stochastic pth order oracles for every p, and lead to improved rates in the convex setting [FSS$^+$19]. Is it possible to beat the ϵ^{-3} lower bound for such oracles, or do our lower bounds extend to this setting?

Adaptivity and instance-dependent complexity. Our lower bounds show that stochastic higher-order methods cannot improve the ϵ^{-3} oracle complexity attained with stochastic gradients and Hessian-vector products. Furthermore, in the multi-point query model, stochastic second-order information does not even lead to improved rates over stochastic first-order information. However, these conclusions could be artifacts of our worst-case point of view—are there natural families of problem instances for which higher-order methods can adapt to additional problem structure and obtain stronger instance-dependent convergence guarantees? Developing a theory of instance-dependent complexity that can distinguish adaptive algorithms stands out as an exciting research prospect.

References

[ACD$^+$23] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199:165–214, 2023.

[AH18] Naman Agarwal and Elad Hazan. Lower bounds for higher-order convex optimization. In *Conference on Learning Theory*, volume 75, pages 774–792. PMLR, 2018.

[AS16] Yossi Arjevani and Ohad Shamir. On the iteration complexity of oblivious first-order optimization algorithms. In *International Conference on Machine Learning*, pages 908–916. PMLR, 2016.

[AS17] Yossi Arjevani and Ohad Shamir. Oracle complexity of second-order methods for finite-sum problems. In *International Conference on Machine Learning*, pages 205–213. PMLR, 2017.

[ASS19] Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178:327–360, 2019.

[AZ18a] Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex SGD. *Advances in Neural Information Processing Systems*, 31:1165–1175, 2018.

[AZ18b] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. *Advances in Neural Information Processing Systems*, 31:2675–2686, 2018.

[AZL17] Zeyuan Allen-Zhu and Yuanzhi Li. Follow the compressed leader: Faster algorithms for matrix multiplicative weight updates. In *International Conference on Machine Learning*, volume 70, pages 116–125. PMLR, 2017.

[CDHS17] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning*, volume 70, pages 654–663. PMLR, 2017.

[CDHS18] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[CDHS20] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184:71–120, 2020.

[CDHS21] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

[CGT17] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Improved second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *arXiv preprint arXiv:1708.04044*, 2017.

[CM20] Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In *International Conference on Machine Learning*, pages 2260–2268. PMLR, 2020.

[CO19] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. *Advances in Neural Information Processing Systems*, 32, 2019.

[FLLZ18] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31:689–699, 2018.

[FLZ19] Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex SGD escaping from saddle points. In *Conference on Learning Theory*, volume 99, pages 1192–1234. PMLR, 2019.

[FSS+19] Dylan J Foster, Ayush Sekhari, Ohad Shamir, Nathan Srebro, Karthik Sridharan, and Blake Woodworth. The complexity of making the gradient small in stochastic convex optimization. In *Conference on Learning Theory*, pages 1319–1345. PMLR, 2019.

[GHJY15] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842. PMLR, 2015.

[GL13] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[GLM16] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in Neural Information Processing Systems*, 29:2973–2981, 2016.

[JGN+17] 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*, volume 70, pages 1724–1732. PMLR, 2017.

[LJCJ17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via SCSG methods. *Advances in Neural Information Processing Systems*, 30:2348–2358, 2017.

[MJC+14] Lester Mackey, Michael I Jordan, Richard Y Chen, Brendan Farrell, and Joel A Tropp. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014.

[MK87] Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.

[MS13] Renato DC Monteiro and Benar Fux Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[MWCC20] Cong Ma, Kaizheng Wang, Yuejie Chi, and Yuxin Chen. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion, and blind deconvolution. *Foundations of Computational Mathematics*, 20:451–632, 2020.

[NLST17] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, volume 70, pages 2613–2621. PMLR, 2017.

[NP06] Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006.

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.

[Oja82] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

[Pea94] Barak A Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

[Sim18] Max Simchowitz. On the randomized complexity of minimizing a convex quadratic function. *arXiv preprint arXiv:1807.09386*, 2018.

[SQW18] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18:1131–1198, 2018.

[TSJ+18] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *Advances in Neural Information Processing Systems*, 31:2899–2908, 2018.

[WJZ+19] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. SpiderBoost and momentum: Faster stochastic variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

[XJY18] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *Advances in Neural Information Processing Systems*, 31:5530–5540, 2018.

[ZG19] Dongruo Zhou and Quanquan Gu. Lower bounds for smooth nonconvex finite-sum optimization. In *International Conference on Machine Learning*, pages 7574–7583. PMLR, 2019.

[ZXG18] Dongruo Zhou, Pan Xu, and Quanquan Gu. Finding local minima via stochastic nested variance reduction. *arXiv preprint arXiv:1806.08782*, 2018.

[ZXG20] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. *Journal of Machine Learning Research*, 21(103):1–63, 2020.