A Stochastic Tensor Method for Non-convex Optimization

Aurelien Lucchi Jonas Kohler
Department of Computer Science, ETH Zürich

November 26, 2019

Abstract

We present a stochastic optimization method that uses a fourth-order regularized model to find local minima of smooth and potentially non-convex objective functions. This algorithm uses sub-sampled derivatives instead of exact quantities and its implementation relies on tensor-vector products only. The proposed approach is shown to find an (ϵ_1, ϵ_2)-second-order critical point in at most $\mathcal{O}\left(\max\left(\epsilon_1^{-4/3}, \epsilon_2^{-2}\right)\right)$ iterations, thereby matching the rate of deterministic approaches. Furthermore, we discuss a practical implementation of this approach for objective functions with a finite-sum structure, as well as characterize the total computational complexity, for both sampling with and without replacement. Finally, we identify promising directions of future research to further improve the complexity of the discussed algorithm.

1 Introduction

We consider the problem of optimizing an objective function of the form

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right], \tag{1}$$

where $f(\mathbf{x}) \in C^3(\mathbb{R}^d, \mathbb{R})$ is a not necessarily convex loss function defined over n datapoints.

Our setting is one where access to the exact function gradient ∇f is computationally expensive (e.g. large-scale setting where n is large) and one therefore wants to access only stochastic evaluations ∇f_i, potentially over a mini-batch. In such settings, stochastic gradient descent (SGD) has long been the method of choice in the field of machine learning. Despite the uncontested empirical success of SGD to solve difficult optimization problems – including training deep neural networks – the convergence speed of SGD is known to slow down close to saddle points or in ill-conditioned landscapes [38, 23]. While gradient descent requires $\mathcal{O}(\epsilon^{-2})$ oracle evaluations[1] to reach an ϵ-approximate *first-order* critical point, the complexity worsens to $\mathcal{O}(\epsilon^{-4})$ for SGD. In contrast, high-order methods – including e.g. regularized Newton methods and trust-region methods – exploit curvature information, allowing them to enjoy faster convergence to a *second-order* critical point.

In this work, we focus our attention on regularized high-order methods to optimize Eq. (1), which construct and optimize a local Taylor model of the objective in each iteration with an additional step length penalty term that depends on how well the model approximates the real objective. This paradigm goes back to Trust-Region and Cubic Regularization methods, which also

[1] In "oracle evaluations" we include the number of function and gradient evaluations as well as evaluations of higher-order derivatives.

make use of regularized models to compute their update step [20, 40, 16]. For the class of second-order Lipschitz smooth functions, [40] showed that the Cubic Regularization framework finds an (ϵ_1, ϵ_2)-approximate second-order critical point in at most $\max\left(\mathcal{O}(\epsilon_1^{-3/2}), \mathcal{O}(\epsilon_2^{-2})\right)$ iterations[2], thus achieving the best possible rate in this setting [14]. Recently, stochastic extensions of these methods have appeared in the literature such as [19, 34, 46, 51]. These will be discussed in further details in Section 2.

Since the use of second derivatives can provide significant theoretical speed-ups, a natural question is whether higher-order derivatives can result in further improvements. This questions was answered affirmatively in [8] who showed that using derivatives up to order $p \geq 1$ allows convergence to an ϵ_1-approximate first-order critical point in at most $\mathcal{O}(\epsilon_1^{-(p+1)/p})$ evaluations. This result was extended to ϵ_2-second-order stationarity in [18], which proves an $\mathcal{O}(\epsilon_2^{-(p+1)/(p-1)})$ rate. Yet, these results assume a deterministic setting where access to exact evaluations of the function derivatives is needed and – to the best of our knowledge – the question of using high-order ($p \geq 3$) derivatives in a stochastic setting has received little consideration in the literature so far. We focus our attention on the case of computing derivative information of up to order $p = 3$. It has recently been shown in [39] that, while optimizing degree four polynomials is NP-hard in general [31], the specific models that arise from a third-order Taylor expansion with a quartic regularizer can still be optimized efficiently.

The main contribution of this work (Thm. 6) is to demonstrate that a stochastic fourth-order regularized method finds an (ϵ_1, ϵ_2) second-order stationary point in $\max\left(\mathcal{O}(\epsilon_1^{-3/2}), \mathcal{O}(\epsilon_2^{-2})\right)$ iterations . Therefore, it matches the results obtained by deterministic methods while relying only on *inexact* derivative information. In order to prove this result, we develop a novel tensor concentration inequality (Thm. 7) for sums of tensors of any order and make explicit use of the finite-sum structure given in Eq. (1). Together with existing matrix and vector concentration bounds [48], this allows us to define a sufficient amount of samples needed for convergence. We thereby provide theoretically motivated sampling schemes for the derivatives of the objective – for both sampling with and without replacement – and discuss a practical implementation of the resulting approach, which we name STM (Stochastic Tensor Method). We also compute the total computational complexity of this algorithm (including the complexity of optimizing the model at each iteration) and find that the complexity in terms of ϵ is superior to other state-of-the-art stochastic algorithms such as [52, 46, 3]. Importantly, the implementation of this approach does not require computing high-order derivatives directly, which would potentially render it as too computationally and memory expensive. Instead, the necessary derivative information is accessed only indirectly via tensor-vector and matrix-vector products.

2 Related work

Sampling techniques for first-order methods. In large-scale learning ($n \gg d$) most of the computational cost of traditional deterministic optimization methods is spent in computing the exact gradient information. A common technique to address this issue is to use sub-sampling to compute an unbiased estimate of the gradient. The simplest instance is SGD whose convergence does not depend on the number of datapoints n. However, the variance in the stochastic gradient estimates slows its convergence down. The work of [26] explored a sub-sampling technique in the

[2]The $\max\left(\mathcal{O}(\epsilon_1^{-3/2}), \mathcal{O}(\epsilon_2^{-2})\right)$ complexity ignores the cost of optimizing the model at each iteration. This is an issue we will revisit later on.

case of convex functions, showing that it is possible to maintain the same convergence rate as full-gradient descent by carefully increasing the sample size over time. Another way to recover a linear rate of convergence for strongly-convex functions is to use variance reduction [33, 24, 43, 32, 22]. The convergence of SGD and its variance-reduced counterpart has also been extended to non-convex functions [28, 42] but the guarantees these methods provide are only in terms of first-order stationarity. However, the work of [27, 44, 21] among others showed that SGD can achieve stronger guarantees in the case of strict-saddle functions. Yet, the convergence rate has a polynomial dependency to the dimension d and the smallest eigenvalue of the Hessian which can make this method fairly impractical.

Second-order methods. For second-order methods that are not regularized or that make use of positive definite Hessian approximations (e.g. Gauss-Newton), the problem of avoiding saddle points is even worse as they might be attracted by saddle points or even local maximima [23]. Another predominant issue is the computation (and storage) of the Hessian matrix, which can be partially addressed by Quasi-Newton methods such as (L-)BFGS. An increasingly popular alternative is to use sub-sampling techniques to approximate the Hessian matrix, such as in [10] and [25]. The latter method uses a low-rank approximation of the Hessian to reduce the complexity per iteration. However, this yields a composite convergence rate: quadratic at first but only linear near the minimizer.

Cubic regularization and trust region methods. Trust region methods are among the most effective algorithmic frameworks to avoid pitfalls such as local saddle points in non-convex optimization. Classical versions iteratively construct a local quadratic model and minimize it within a certain radius wherein the model is trusted to be sufficiently similar to the actual objective function. This is equivalent to minimizing the model function with a suitable *quadratic* penalty term on the stepsize. Thus, a natural extension is the cubic regularization method introduced by [40] that uses a *cubic* over-estimator of the objective function as a regularization technique for the computation of a step to minimize the objective function. The drawback of their method is that it requires computing the exact minimizer of the cubic model, thus requiring the exact gradient and Hessian matrix. However finding a global minimizer of the cubic model $m_k(\mathbf{s})$ may not be essential in practice and doing so might be prohibitively expensive from a computational point of view. [16] introduced a method named ARC which relaxed this requirement by letting $\mathbf{s}_k = \operatorname{argmin}_{\mathbf{s}} m_k(\mathbf{s})$ be an approximation to the minimizer. The model defined by the adaptive cubic regularization method introduced two further changes. First, instead of computing the exact Hessian $\mathbf{H}_k$ it allows for a symmetric approximation $\mathbf{B}_k$. Second, it introduces a dynamic step-length penalty parameter σ_k instead of using the global Lipschitz constant. Our approach relies on the same adaptive framework.

There have been efforts to further reduce the computational complexity of optimizing the model. For example, [2] refined the approach of [40] to return an approximate local minimum in time which is linear in the input. Similar improvements have been made by [11] and [30]. These methods provide alternatives to minimize the cubic model and can thus be seen as complementary to our approach. Finally, [9] proposed a stochastic trust region method but their analysis does not specify any accuracy level required for the estimation of the stochastic Hessian. [19] also analyzed a probabilistic cubic regularization variant that allows for approximate second-order models. [34] provided an explicit derivation of sampling conditions to preserve the worst-case complexity of ARC. Other works also derived similar stochastic extensions to cubic regularization, including [51] and [46]. The worst-case rate derived in the latter includes the complexity of a specific model solver

introduced in [11].

High-order models. A hybrid algorithm suggested in [4] adds occasional third-order steps to a cubic regularization method, thereby obtaining provable convergence to some type of third-order local minima. Yet, high-order derivatives can also directly be applied within the regularized Newton framework, as done e.g. by [8] that extended cubic regularization to a p-th high-order model (Taylor approximation of order p) with a $(p+1)$-th order regularization, proving iteration complexities of order $\mathcal{O}(\epsilon^{-(p+1)/p})$ for first-order stationarity. These convergence guarantees are extended to the case of inexact derivatives in [7] but the latter only discusses a practical implementation for the case $p = 2$. The convergence guarantees of p-th order models are extended to second-order stationarity in [18]. Notably, all these approaches require optimizing a $(p+1)$-th order polynomial, which is known to be a difficult problem. Recently, [39, 29] introduced an implementable method for the case $p = 3$ in the deterministic case and for convex functions. We are not aware of any work that relies on high-order derivatives ($p \geq 4$) to construct a model to optimize smooth functions and we therefore focus our work on the case $p = 3$.

Finally, another line of works [3, 52] considers methods that do not use high-order derivatives explicitly within a regularized Newton framework but rather rely on other routines – such as Oja's algorithm – to explicitly find negative curvature directions and couple those with SGD steps.

3 Formulation

3.1 Notation & Assumptions

First, we lay out some standard assumptions regarding the function f as well as the required approximation quality of the high-order derivatives.

Assumption 1 (Continuity)**.** *The functions $f_i \in C^3(\mathbb{R}^d, \mathbb{R})$, $\nabla f_i, \nabla^2 f_i$, $\nabla^3 f_i$ are Lipschitz continuous for all i, with Lipschitz constants L_f, L_g, L_b and L_t respectively.*

In the following, we will denote the directional derivative of the function f at $\mathbf{x}$ along the directions $\mathbf{h}_j \in \mathbb{R}^d, j = 1 \dots p$ as

$$\nabla^p f(\mathbf{x})[\mathbf{h}_1, \dots \mathbf{h}_p]. \tag{2}$$

For instance, $\nabla f(\mathbf{x})[\mathbf{h}] = \nabla f(\mathbf{x})^\top \mathbf{h}$ and $\nabla^2 f(\mathbf{x})[\mathbf{h}]^2 = \mathbf{h}^\top \nabla^2 f(\mathbf{x}) \mathbf{h}$.

Assumption 1 implies that for each $p = 0 \dots 3$,

$$\|\nabla^p f_i(\mathbf{x}) - \nabla^p f_i(\mathbf{y})\|_{[p]} \leq L_p \|\mathbf{x} - \mathbf{y}\| \tag{3}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and where $L_0 = L_f, L_1 = L_g, L_2 = L_b, L_3 = L_t$.

As in [18], $\|\cdot\|_{[p]}$ is the tensor norm recursively induced by the Euclidean norm $\|\cdot\|$ on the space of p-th order tensors.

3.2 Stochastic surrogate model

We construct a surrogate model to optimize f based on a truncated Taylor approximation as well as a power prox function weighted by a sequence $\{\sigma_k\}_k$ that is controlled adaptively according to the fit of the model to the function f. Since the full Taylor expansion of f requires computing

high-order derivatives that are expensive, we instead use an inexact model defined as

$$m_k(\mathbf{s}) = \phi_k(\mathbf{s}) + \frac{\sigma_k}{4}\|\mathbf{s}\|_2^4,$$
$$\phi_k(\mathbf{s}) = f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{B}_k \mathbf{s} + \frac{1}{6}\mathbf{T}_k[\mathbf{s}]^3 \tag{4}$$

where $\mathbf{g}_k, \mathbf{B}_k$ and $\mathbf{T}_k$ approximate the derivatives $\nabla f(\mathbf{x}_k), \nabla^2 f(\mathbf{x}_k)$ and $\nabla^3 f(\mathbf{x}_k)$ through sampling as follows. Three sample sets $\mathcal{S}^g, \mathcal{S}^b$ and $\mathcal{S}^t$ are drawn and the derivatives are then estimated as

$$\mathbf{g}_k = \frac{1}{|\mathcal{S}^g|}\sum_{i\in\mathcal{S}^g} \nabla f_i(\mathbf{x}_k), \mathbf{B}_k = \frac{1}{|\mathcal{S}^b|}\sum_{i\in\mathcal{S}^b} \nabla^2 f_i(\mathbf{x}_k),$$
$$\mathbf{T}_k = \frac{1}{|\mathcal{S}^t|}\sum_{i\in\mathcal{S}^t} \nabla^3 f_i(\mathbf{x}_k). \tag{5}$$

We will later see that the implementation of the algorithm we analyze does not require the computation of the Hessian or the third-order tensor – both of which would require significant computational resources – but instead directly compute Tensor-vector products with a complexity of order $O(d)$.

We will make use of the following condition in order to reach an ϵ-critical point:

Condition 1. *For a given ϵ accuracy, one can choose the size of the sample sets $\mathcal{S}^g, \mathcal{S}^b, \mathcal{S}^t$ for sufficiently small $\kappa_g, \kappa_b, \kappa_t > 0$ such that:*

$$\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\| \leq \kappa_g \epsilon \tag{6}$$
$$\|(\mathbf{B}_k - \nabla^2 f(\mathbf{x}_k))\mathbf{s}\| \leq \kappa_b \epsilon^{2/3}\|\mathbf{s}\|, \quad \forall \mathbf{s} \in \mathbb{R}^d \tag{7}$$
$$\|\mathbf{T}_k[\mathbf{s}]^2 - \nabla^3 f(\mathbf{x}_k)[\mathbf{s}]^2\| \leq \kappa_t \epsilon^{1/3}\|\mathbf{s}\|^2, \quad \forall \mathbf{s} \in \mathbb{R}^d. \tag{8}$$

In Lemma 8, we prove that we can choose the size of each sample set $\mathcal{S}^g, \mathcal{S}^b$ and $\mathcal{S}^t$ to satisfy the conditions above, without requiring knowledge of the length of the step $\|\mathbf{s}_k\|$. We will present a convergence analysis of STM, proving that the convergence properties of the deterministic methods [8, 39] can be retained by a sub-sampled version at the price of slightly worse constants.

3.3 Algorithm

The optimization algorithm we consider is detailed in Algorithm 1. At iteration step k, we sample three sets of datapoints from which we compute stochastic estimates of the derivatives of f so as to satisfy Cond. 1. We then obtain the step $\mathbf{s}_k$ by solving the problem

$$\mathbf{s}_k = \arg\min_{\mathbf{s}\in\mathbb{R}^d} m_k(\mathbf{s}), \tag{12}$$

either exactly or approximately (details will follow shortly) and update the regularization parameter σ_k depending on ρ_k, which measures how well the model approximates the real objective. This is accomplished by differentiating between different types of iterations. Successful iterations (for which $\rho_k \geq \eta_1$) indicate that the model is, at least locally, an adequate approximation of the objective such that the penalty parameter is decreased in order to allow for longer steps. We denote the index set of all successful iterations between 0 and k by $\mathcal{S}_k = \{0 \leq j \leq k | \rho_j \geq \eta_1\}$. We also denote by $\mathcal{U}_k$ its complement in $\{0, \dots k\}$ which corresponds to the index set of unsuccessful iterations.

Algorithm 1 Stochastic Tensor Method (STM)

Input:
Starting point $\mathbf{x}_0 \in \mathbb{R}^d$ (e.g $\mathbf{x}_0 = \mathbf{0}$)
$0 < \gamma_1 < 1 < \gamma_2 < \gamma_3, 1 > \eta_2 > \eta_1 > 0$, and $\sigma_0 > 0, \sigma_{min} > 0$
for $k = 0, 1, \ldots,$ until convergence **do**
  Sample gradient $\mathbf{g}_k$, Hessian $\mathbf{B}_k$ and $\mathbf{T}_k$ such that Eq. (6), Eq. (7) & Eq. (8) hold.
  Obtain $\mathbf{s}_k$ by solving $m_k(\mathbf{s}_k)$ (Eq. (4)) such that Condition 2 holds.
  Compute $f(\mathbf{x}_k + \mathbf{s}_k)$ and

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{f(\mathbf{x}_k) - \phi_k(\mathbf{s}_k)}. \tag{9}$$

  Set

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k \geq \eta_1 \\ \mathbf{x}_k & \text{otherwise.} \end{cases} \tag{10}$$

  Set

$$\sigma_{k+1} = \begin{cases} [\max\{\sigma_{min}, \gamma_1\sigma_k\}, \sigma_k] & \text{if } \rho_k > \eta_2 \text{ (very successful iteration)} \\ [\sigma_k, \gamma_2\sigma_k] & \text{if } \eta_2 \geq \rho_k \geq \eta_1 \text{ (successful iteration)} \\ [\gamma_2\sigma_k, \gamma_3\sigma_k] & \text{otherwise (unsuccessful iteration).} \end{cases} \tag{11}$$

end for

Exact model minimization Solving Eq. (4) requires minimizing a nonconvex multivariate polynomials. As pointed out in [39, 5], this problem is computationally expensive to solve in general and further research is needed to establish whether one could design a practical minimization method. In [39], the authors demonstrated that an appropriately regularized Taylor approximation of convex functions is a convex multivariate polynomial, which can be solved using the framework of relatively smooth functions developed in [35]. As long as the involved models are convex, this solver could be used in Algorithm 1 but it is unclear how to generalize this method to the non-convex case. Fortunately, we will see next that exact model minimizers is not even needed to establish global convergence guarantees for our method.

Approximate model minimization Exact minimization of the model in Eq. (4) is often computationally expensive, especially given that it is required for every parameter update in Algorithm 1. In the following, we explore an approach to approximately minimize the model while retaining the convergence guarantees of the exact minimization. Instead of requiring the exact optimality conditions to hold, we use weaker conditions that were also used in prior work [8, 18]. First, we define two criticality measures based on first- and second-order information:

$$\begin{aligned} \chi_{f,1}(\mathbf{x}_k) &:= \|\nabla f(\mathbf{x}_k)\|, \\ \chi_{f,2}(\mathbf{x}_k) &:= \max\left(0, -\lambda_{min}(\nabla^2 f(\mathbf{x}_k))\right), \end{aligned} \tag{13}$$

where $\lambda_{min}(\nabla^2 f(\mathbf{x}))$ is the minimum eigenvalue of the Hessian matrix $\nabla^2 f(\mathbf{x})$.

The same criticality measures are defined for the model $m_k(\mathbf{s})$,

$$\begin{aligned} \chi_{m,1}(\mathbf{x}_k, \mathbf{s}) &:= \|\nabla_{\mathbf{s}} m_k(\mathbf{s})\|, \\ \chi_{m,2}(\mathbf{x}_k, \mathbf{s}) &:= \max\left(0, -\lambda_{min}(\nabla^2_{\mathbf{s}} m_k(\mathbf{s}))\right). \end{aligned} \tag{14}$$

Finally, we state the approximate optimality condition required to find the step $\mathbf{s}_k$.

Condition 2. *For each outer iteration k, the step $\mathbf{s}_k$ is computed so as to approximately minimize the model $m_k(\mathbf{s}_k)$ in the sense that the following conditions hold:*

$$\begin{aligned} m_k(\mathbf{s}_k) &< m_k(\mathbf{0}) \\ \chi_{m,i}(\mathbf{x}_k, \mathbf{s}_k) &\leq \theta \|\mathbf{s}_k\|^{4-i}, \quad \theta > 0, \quad \text{for } i = 1, 2. \end{aligned} \tag{15}$$

Practical implementation In Section 4.3, we will discuss how a gradient-based method can be used to approximately optimize the model defined in Eq. (4). Such an algorithm only needs to access the first derivative of the model w.r.t. $\mathbf{s}$, defined as

$$\nabla_{\mathbf{s}} m_k(\mathbf{s}) = \mathbf{g}_k + \mathbf{B}_k \mathbf{s} + \frac{1}{2}\mathbf{T}_k[\mathbf{s}]^2 + \sigma_k \mathbf{s}\|\mathbf{s}\|^2, \tag{16}$$

and it therefore has a low computational complexity per-iteration.

4 Analysis

4.1 Worst-case complexity

In the following, we provide a proof of convergence of STM to a second-order critical point, i.e. a point $\mathbf{x}^*$ such that

$$\chi_{f,i}(\mathbf{x}^*) \leq \epsilon_i \quad \text{for } i = 1, 2. \tag{17}$$

The high-level idea of the analysis is to first show that the model decreases proportionally to the criticality measures at each iteration and then relate the model decrease to the function decrease. Since the function f is lower bounded, it can only decrease a finite number of times, which therefore implies convergence. We start with a bound on the model decrease in terms of the step length $\|\mathbf{s}\|$.

Lemma 2. *For any $\mathbf{x}_k \in \mathbb{R}^d$, the step $\mathbf{s}_k$ (satisfying Cond. 2) is such that*

$$\phi_k(\mathbf{0}) - \phi_k(\mathbf{s}_k) > \frac{\sigma_k}{4}\|\mathbf{s}_k\|^4. \tag{18}$$

In order to complete our claim of model decrease, we prove that the length of the step $\mathbf{s}_k$ can not be arbitrarily small compared to the gradient and the Hessian of the objective function.

Lemma 3. *Suppose that Condition 1 holds with the choice $\kappa_g = \frac{1}{4}$, $\kappa_b = \frac{1}{4}$, $\kappa_t = \frac{1}{2}$. For any $\mathbf{x}_k \in \mathbb{R}^d$, the length of the step $\mathbf{s}_k$ (satisfying Cond. 2) is such that*

$$\|\mathbf{s}_k\| \geq \kappa_k^{-1/3} \left(\chi_{f,1}(\mathbf{x}_k + \mathbf{s}_k) - \frac{1}{2}\epsilon_1 \right)^{1/3}, \tag{19}$$

where $\kappa_k = \left(\sigma_k + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$.

Lemma 4. *Suppose that Condition 1 holds with the choice $\kappa_g = \frac{1}{4}$, $\kappa_b = \frac{1}{4}$, $\kappa_t = \frac{1}{2}$. For any $\mathbf{x}_k \in \mathbb{R}^d$, the length of the step $\mathbf{s}_k$ (satisfying Cond. 2) is such that*

$$\|\mathbf{s}_k\| \geq \kappa_{k,2}^{-1/2} \left(\chi_{f,2}(\mathbf{x}_k + \mathbf{s}_k) - \frac{1}{2}\epsilon_2 \right)^{1/2}, \tag{20}$$

where $\kappa_{k,2} = \left(3\sigma_k + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$.

A key lemma to derive a worst-case complexity bound on the total number of iterations required to reach a second-order critical point is to bound the number of unsuccessful iterations $|\mathcal{U}_k|$ as a function of the number of successful ones $|\mathcal{S}_k|$, that have occurred up to some iteration $k > 0$.

Lemma 5. *The steps produced by Algorithm 1 guarantee that if $\sigma_k \leq \sigma_{max}$ for $\sigma_{max} > 0$, then $k \leq C(\gamma_1, \gamma_2, \sigma_{max}, \sigma_0)$ where*

$$C(\gamma_1, \gamma_2, \sigma_{max}, \sigma_0) := \quad \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2}\right) |\mathcal{S}_k| + \frac{1}{\log \gamma_2} \log\left(\frac{\sigma_{max}}{\sigma_0}\right).$$

The proof of this Lemma can be found in [16] (Theorem 2.1) and is also restated in the Appendix. A closed-form expression for σ_{max} is provided in Lemma 16 in Appendix.

We are now ready to state the main result of this section that provides a bound on the number of iterations required to reach a second-order critical point.

Theorem 6 (Outer worst-case complexity)**.** *Let f_{low} be a lower bound on f and assume Conditions 1 and 2 hold. Let $\kappa_s = \left(\sigma_{max} + \frac{L_t}{2} + \theta + \frac{1}{4}\right), \kappa_{s,2} = \left(3\sigma_{max} + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$ and $\kappa_{max} = \max(\sqrt[3]{2}\kappa_s^{4/3}, 2\kappa_{s,2}^2)$. Then, given $\epsilon_1, \epsilon_2 > 0$, Algorithm 1 needs at most*

$$\left\lceil \mathcal{K}_{succ}(\epsilon) := \frac{8\kappa_{max}(f(\mathbf{x}_0) - f_{low})}{\eta_1 \sigma_{min}} \max(\epsilon_1^{-4/3}, \epsilon_2^{-2}) \right\rceil$$

successful iterations and

$$\mathcal{K}_{outer}(\epsilon) := \lceil C(\gamma_1, \gamma_2, \sigma_{max}, \sigma_0) \cdot \mathcal{K}_{succ}(\epsilon) \rceil. \tag{21}$$

total outer iterations to reach an iterate $\mathbf{x}^*$ such that both $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon_1$ and $\lambda_{min}(\nabla^2 f(\mathbf{x}^*)) \geq -\epsilon_2$.

As in [18], we obtain a worst-complexity bound of the order $\mathcal{O}(\max(\epsilon_1^{-\frac{4}{3}}, \epsilon_2^{-2}))$, except that we do not require the exact computation of the function derivatives but instead rely on approximate sampled quantities. By using third-order derivatives, STM also obtains a faster rate (in terms of outer iterations) than the one achieved by a sampled variant of cubic regularization [34] (at most $\mathcal{O}(\epsilon^{-3/2})$ iterations for $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$ and $\mathcal{O}(\epsilon^{-3})$ to reach approximate nonnegative curvature). The worst-case complexity bound stated in Theorem 6 does not take into account the complexity of the subsolver used to find the (approximate) solution of $\mathbf{s}_k = \arg\min_{\mathbf{s}} m_k(\mathbf{s})$. This is discussed in detail in Section 4.3.

4.2 Sampling conditions

We now show that one can use random sampling without replacement and choose the size of the sample sets in order to satisfy Cond. 1 with high probability. The complete proof is available in Appendix B.1. The case of sampling with replacement is also discussed in Appendix B.2.

First, we need to develop a new concentration bound for tensors based on the spectral norm. Existing tensor concentration bounds are not applicable to our setting. Indeed, [36] relies on a different norm which can not be translated to the spectral norm required in our analysis while the bound derived in [49] relies on a specific form of the input tensor.

Formally, let $(\Omega, \mathcal{F}, P)$ be a probability space and let $\mathcal{X}$ be a real (m, d) random tensor, i.e. a measurable map from Ω to $\mathbb{T}_{m,d}$ (the space of real tensors of order m and dimension d).

Our goal is to derive a concentration bound for a sum of n identically distributed tensors sampled *without* replacement, i.e. we consider

$$\mathcal{X} = \sum_{i=1}^{n} \mathcal{Y}_i,$$

where each tensor $\mathcal{Y}_i$ is sampled from a population $\mathcal{A}$ of size $N > n$.

The concentration result derived in the next theorem is based on the proof technique introduced in [45] which we adapt for sums of random variables.

Theorem 7 (Tensor Hoeffding-Serfling Inequality)**.** *Let $\mathcal{X}$ be a sum of n tensors $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ sampled without replacement from a finite population $\mathcal{A}$ of size N. Let $\mathbf{u}_1, \dots \mathbf{u}_k$ be such that $\|\mathbf{u}_i\| = 1$ and assume that for each tensor i, $a \leq \mathcal{Y}_i(\mathbf{u}_1, \dots, \mathbf{u}_k) \leq b$. Let $\sigma := (b-a)$, then we have*

$$P(\,\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) \leq k_0^{(\sum_{i=1}^k d_i)} \cdot 2\exp\left(-\frac{t^2 n^2}{2\sigma^2(n+1)(1-n/N)}\right),$$

where $k_0 = \left(\frac{2k}{\log(3/2)}\right)$.

Based on Theorem 7 as well as standard concentration bounds for vectors and matrices (see e.g. [48]), we prove the required sampling conditions of Cond. 1 hold for the specific sample sizes given in the next lemma.

Lemma 8. *Consider the sub-sampled gradient, Hessian and third-order tensor defined in Eq.* (5)*. Under Assumption 1, the sampling conditions in Eqs.* (6)*,* (7) *and* (8) *are satisfied with probability $1-\delta$, $\delta \in (0,1)$ for the following choice of the size of the sample sets $\mathcal{S}^g, \mathcal{S}^b$ and $\mathcal{S}^t$:*

$$\begin{aligned} n_g &= \tilde{\mathcal{O}}\left(\kappa_g^2 \epsilon^2 \,/ L_f^2 + 1\,/N\right)^{-1}, \\ n_b &= \tilde{\mathcal{O}}\left(\kappa_b^2 \epsilon^{4/3} \,\Big/ L_g^2 + 1\,/N\right)^{-1}, \\ n_t &= \tilde{\mathcal{O}}\left(\kappa_t^2 \epsilon^{2/3} \,\Big/ L_b^2 + 1\,/N\right)^{-1}, \end{aligned}$$

where $\tilde{\mathcal{O}}$ hides poly-logarithmic factors and a polynomial dependency to d.

4.3 Complexity subproblem

First-order guarantees We first discuss the scenario where we only require an ϵ-first-order critical point. In the next theorem, we state the total complexity of Algorithm 1, including the sampling complexities given in Lemma 8 as well as the subsolver complexity, which we denote by $\mathcal{K}(\epsilon)$.

Theorem 9 (Total worst-case complexity)**.** *Let f_{low} be a lower bound on f. Denote by $\mathcal{K}_{outer}(\epsilon)$ the number of outer iterations defined in Eq.* (21) *and by $\mathcal{K}(\epsilon)$ the complexity of the model subsolver, both specialized to the case of first-order criticality. Assume Condition 1 holds. Then, given $\epsilon > 0$, Algorithm 1 needs at most*

$$\mathcal{K}_{outer}(\epsilon) \cdot (n_g + n_b \mathcal{K}(\epsilon) + n_t \mathcal{K}(\epsilon))$$

(stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$.

We now derive a corollary for the case where the desired accuracy is high, i.e. $\epsilon \ll \frac{1}{N}$. We suppose the subsolver is a non-convex version of AGD whose worst-case complexity under Assumption 1 is proven to be $\mathcal{O}(\epsilon^{-5/3})$ in [13].

Corollary 10 (Total worst-case complexity - first-order stationarity)**.** *Under the same assumptions as in Theorem 9, Algorithm 1 with the non-convex AGD variant presented in [13] as subsolver needs at most $\tilde{\mathcal{O}}(N\epsilon^{-3})$ (stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$ and $\epsilon \ll \frac{1}{N}$.*

The final total complexity in terms of ϵ is an improvement over state-of-the-art methods such as NEON + SCSG [52] ($\mathcal{O}(\epsilon^{-3.33})$), SCR [46] and Natasha2 [3] ($\mathcal{O}(\epsilon^{-3.5})$). We expect that the dependency on N could be further reduced using the variance reduction technique introduced in [50]. Furthermore, we want to point out that the use of a specialized subproblem solver instead of AGD – as was done in [11] for the cubic model – could *significantly* improve the rate. For comparison, while the rate of AGD to reach $\|\nabla f(\mathbf{x})\| \leq \epsilon$ is $\mathcal{O}(\epsilon^{-5/3})$, the cubic solver from [11] achieves $\mathcal{O}(\epsilon^{-1})$.[3]

Second-order guarantees Unlike the first-order guarantees, the condition imposed on the subproblem solver now requires finding a second-order stationary point. A common solver used for trust-region methods and ARC is to apply a Lanczos method to build up evolving Krylov spaces, which can be constructed in a Hessian-free manner, i.e. by accessing the Hessians only indirectly via matrix-vector products. We are however not aware of any existing work analyzing the worst-case complexity of (a generalization of) this approach for higher-order polynomials. Instead, we suggest using the solver presented in [15] which is an accelerated gradient method for non-convex optimization that requires $\tilde{\mathcal{O}}(\epsilon^{-7/4})$ to find a point $\mathbf{x}^*$ such that $\chi_{f,2}(\mathbf{x}^*) \leq \sqrt{\epsilon_2}$.

Corollary 11 (Total worst-case complexity – second-order stationarity)**.** *Under the same assumptions as in Theorem 9, Algorithm 1 with the non-convex AGD variant presented in [15] as subsolver needs at most $\tilde{\mathcal{O}}(N\epsilon^{-15/4})$ (stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\chi_{f,2}(\mathbf{x}^*) \leq \epsilon_2$ and $\epsilon_2 \ll \frac{1}{N}$.*

As for the first-order guarantees, an interesting direction to improve the total complexity would be a subproblem solver specialized to the specific characteristics of the fourth-order model.

5 Conclusion

We presented a stochastic fourth-order optimization algorithm that preserves the guarantees of its deterministic counterpart for finding an approximate second-order critical point. There are numerous extensions that could follow from this work, including the following.

Algorithmic improvements. Prior work such as [3, 52] has relied on using variance reduction to achieve faster rates. A variance-reduced variant of cubic regularization has also been shown in [50] to reduce the per-iteration sample complexity and one would therefore except similar improvements can be made to the quartic model. Finally, one could incorporate acceleration as in [37].

Model solver Perhaps the most promising direction for future work is to design efficient algorithms to solve high-order polynomial models. While there has been some significant work published for cubic models [11, 12], the problem has been relatively unexplored for higher-order models.

Finally, one relevant application for the type of high-order algorithms we developed is training deep neural networks as in [46, 1]. An interesting direction for future research would therefore be to design a practical implementation of STM for training neural networks based on efficient tensor-vector products similarly to the fast Hessian-vector products proposed in [41].

[3] [11] reports a rate in terms of function suboptimality, which we here naively convert using smoothness.

Acknowledgements The authors would like to thank Coralia Cartis for helpful discussions on an early draft of this paper, as well as for pointing out additional relevant work.

References

[1] Leonard Adolphs, Jonas Kohler, and Aurelien Lucchi. Ellipsoidal trust region methods and the marginal value of hessian information for neural network training. *arXiv preprint arXiv:1905.09201*, 2019.

[2] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*, 2016.

[3] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. In *Advances in Neural Information Processing Systems*, pages 2675–2686, 2018.

[4] Anima Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. *arXiv preprint arXiv:1602.05908*, 2016.

[5] Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[6] Rémi Bardenet, Odalric-Ambrym Maillard, et al. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

[7] Stefania Bellavia, Gianmarco Gurioli, Benedetta Morini, and Philippe Toint. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. 2018.

[8] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Ph L Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1-2):359–368, 2017.

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

[10] Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

[11] Yair Carmon and John C. Duchi. Gradient descent efficiently finds the cubic-regularized nonconvex newton step. *https://arxiv.org/abs/1612.00547*, 2016.

[12] Yair Carmon and John C Duchi. Analysis of krylov subspace solutions of regularized nonconvex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 10705–10715, 2018.

[13] 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 *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 654–663. JMLR. org, 2017.

[14] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

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

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

[17] Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011.

[18] 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.

[19] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, pages 1–39, 2015.

[20] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[21] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. *arXiv preprint arXiv:1803.05999*, 2018.

[22] Hadi Daneshmand, Aurélien Lucchi, and Thomas Hofmann. Starting small - learning with adaptive sample sizes. In *International Conference on Machine Learning*, 2016.

[23] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional nonconvex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

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

[25] Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems*, pages 3052–3060, 2015.

[26] Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[27] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*, pages 797–842, 2015.

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

[29] Geovani Nunes Grapiglia and Yurii Nesterov. On inexact solution of auxiliary problems in tensor methods for convex optimization. *arXiv preprint arXiv:1907.13023*, 2019.

[30] Elad Hazan and Tomer Koren. A linear-time algorithm for trust region problems. *Mathematical Programming*, 158(1-2):363–381, 2016.

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

[32] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems 28*, pages 2296–2304. Curran Associates, Inc., 2015.

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

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

[35] Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

[36] Ziyan Luo, Liqun Qi, and Ph L Toint. Bernstein concentration inequalities for tensors via einstein products. *arXiv preprint arXiv:1902.03056*, 2019.

[37] Yu Nesterov. Accelerating the cubic regularization of newtons method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[38] Yurii Nesterov. Introductory lectures on convex optimization. applied optimization, vol. 87, 2004.

[39] Yurii Nesterov. Implementable tensor methods in unconstrained convex optimization. Technical report, CORE Discussion Paper, Université Catholique de Louvain, Belgium, 2015.

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

[41] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

[42] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. *arXiv preprint arXiv:1603.06160*, 2016.

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

[44] Ju Sun, Qing Qu, and John Wright. When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*, 2015.

[45] Ryota Tomioka and Taiji Suzuki. Spectral norm of random tensors. *arXiv preprint arXiv:1407.1870*, 2014.

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

[47] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

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

[49] Roman Vershynin. Concentration inequalities for random tensors. *arXiv preprint arXiv:1905.00802*, 2019.

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

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

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

Appendix

A Main analysis

In the following section, we provide detailed proofs for all the lemmas and theorems presented in the main paper. First, we present some basic results regarding the surrogate model that will be handy throughout the proofs.

Surrogate model Recall that we use the following surrogate model:

$$m_k(\mathbf{s}) = \phi_k(\mathbf{s}) + \frac{\sigma_k}{4}\|\mathbf{s}\|_2^4,$$
$$\phi_k(\mathbf{s}) = f(\mathbf{x}_k) + \mathbf{g}_k^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{B}_k \mathbf{s} + \frac{1}{6}\mathbf{T}_k[\mathbf{s}]^3 \tag{22}$$

The first derivative of the model w.r.t. $\mathbf{s}$ is

$$\nabla_{\mathbf{s}} m_k(\mathbf{s}) = \mathbf{g}_k + \mathbf{B}_k \mathbf{s} + \frac{1}{2}\mathbf{T}_k[\mathbf{s}]^2 + \sigma_k \mathbf{s}\|\mathbf{s}\|^2. \tag{23}$$

For the second-order derivative $\nabla^2_{\mathbf{s}} m_k(\mathbf{s})$, we get:

$$\nabla^2_{\mathbf{s}} m_k(\mathbf{s}) = \mathbf{B}_k + \mathbf{T}_k[\mathbf{s}] + \frac{\sigma_k}{4}\nabla^2_{\mathbf{s}}\|\mathbf{s}\|^4, \tag{24}$$

where

$$\frac{1}{4}\nabla^2_{\mathbf{s}}\|\mathbf{s}\|^4 = \nabla_{\mathbf{s}}\, \mathbf{s}\|\mathbf{s}\|^2 = \|\mathbf{s}\|^2\mathbf{I} + 2\mathbf{s}\mathbf{s}^\top \succcurlyeq \|\mathbf{s}\|^2 \tag{25}$$

We now start with a bound on the model decrease in terms of the step length $\|\mathbf{s}\|$.

Lemma 12 (Lemma 4.1 restated)**.** *For any $\mathbf{x}_k \in \mathbb{R}^d$, the step $\mathbf{s}_k$ (satisfying Cond. 2) is such that*

$$\phi_k(\mathbf{0}) - \phi_k(\mathbf{s}_k) > \frac{\sigma_k}{4}\|\mathbf{s}_k\|^4. \tag{26}$$

Proof. Note that $m_k(\mathbf{0}) = f(\mathbf{x}_k)$. Using the optimality conditions introduced in Condition 2, we get

$$0 < m_k(\mathbf{0}) - m_k(\mathbf{s}_k) = \phi_k(\mathbf{0}) - \phi_k(\mathbf{s}_k) - \frac{\sigma_k}{4}\|\mathbf{s}_k\|^4, \tag{27}$$

which directly implies the desired result.

□

In order to complete our claim of model decrease started in Lemma 2, we prove that the length of the step $\mathbf{s}_k$ can not be arbitrarily small compared to the gradient of the objective function.

Lemma 13 (Lemma 4.2 restated)**.** *Suppose that Condition 1 holds with the choice $\kappa_g = \frac{1}{4}$, $\kappa_b = \frac{1}{4}$, $\kappa_t = \frac{1}{2}$. For any $\mathbf{x}_k \in \mathbb{R}^d$, the length of the step $\mathbf{s}_k$ (satisfying Cond. 2) is such that*

$$\|\mathbf{s}_k\| \geq \kappa_k^{-1/3}\left(\chi_{f,1}(\mathbf{x}_k + \mathbf{s}_k) - \frac{1}{2}\epsilon_1\right)^{1/3}, \tag{28}$$

where $\kappa_k = \left(\sigma_k + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$.

Proof. We start with the following bound,

$$\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| \leq \|\nabla f(\mathbf{x}_k + \mathbf{s}_k) - \nabla \phi_k(\mathbf{s}_k)\| + \|\nabla \phi_k(\mathbf{s}_k)\|. \tag{29}$$

We bound the second term in the RHS of Eq. (29) using the termination condition presented in Eq. (15). We get

$$\begin{aligned}
\|\nabla \phi_k(\mathbf{s}_k)\| &\leq \left\|\nabla \phi_k(\mathbf{s}_k) + \sigma_k \mathbf{s}_k \|\mathbf{s}_k\|^2\right\| + \sigma_k \|\mathbf{s}_k\|^3 \\
&= \|\nabla m_k(\mathbf{s}_k)\| + \sigma_k \|\mathbf{s}_k\|^3 \\
&\leq \theta \|\mathbf{s}_k\|^3 + \sigma_k \|\mathbf{s}_k\|^3 \\
&\leq (\theta + \sigma_k) \|\mathbf{s}_k\|^3.
\end{aligned} \tag{30}$$

For the first term in the RHS of Eq. (29) , we have

$$\begin{aligned}
\|\nabla f(\mathbf{x}_k + \mathbf{s}_k) - \nabla \phi_k(\mathbf{s}_k)\| &= \left\|\nabla f(\mathbf{x}_k + \mathbf{s}_k) - \mathbf{g}_k - \mathbf{B}_k \mathbf{s}_k - \frac{1}{2}\mathbf{T}_k[\mathbf{s}_k]^2\right\| \\
&\leq \left\|\nabla f(\mathbf{x}_k + \mathbf{s}_k) - \nabla f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)\mathbf{s}_k - \frac{1}{2}\nabla^3 f(\mathbf{x}_k)[\mathbf{s}_k]^2\right\| \\
&+ \|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\| + \|(\mathbf{B}_k - \nabla^2 f(\mathbf{x}_k))\mathbf{s}_k\| \\
&+ \frac{1}{2}\|\mathbf{T}_k[\mathbf{s}_k]^2 - \nabla^3 f(\mathbf{x}_k)[\mathbf{s}_k]^2\| \\
&\leq \frac{L_t}{2}\|\mathbf{s}_k\|^3 + \kappa_g \epsilon + \kappa_b \epsilon^{2/3}\|\mathbf{s}_k\| + \frac{1}{2}\kappa_t \epsilon^{1/3}\|\mathbf{s}_k\|^2,
\end{aligned} \tag{31}$$

where the last inequality uses Lemma 32 and Condition 1 where we set $\epsilon_1 = \epsilon$.

Next, we apply the Young's inequality for products which states that if $a, b \in \mathbb{R}_{\geq 0}$ and $p, q \in \mathbb{R}_{>1}$ such that $1/p + 1/q = 1$ then

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}. \tag{32}$$

We obtain

$$\begin{aligned}
\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| &\leq (\theta + \sigma_k)\|\mathbf{s}_k\|^3 + \frac{L_t}{2}\|\mathbf{s}_k\|^3 + \kappa_g \epsilon \\
&+ \frac{\kappa_b}{3}\left(2\epsilon + \|\mathbf{s}_k\|^3\right) + \frac{\kappa_t}{6}\left(\epsilon + 2\|\mathbf{s}_k\|^3\right) \\
&= \left(\sigma_k + \theta + \frac{L_t}{2} + \frac{\kappa_b}{3} + \frac{\kappa_t}{3}\right)\|\mathbf{s}_k\|^3 + \left(\kappa_g + \frac{2\kappa_b}{3} + \frac{\kappa_t}{6}\right)\epsilon,
\end{aligned} \tag{33}$$

therefore

$$\left(\sigma_k + \theta + \frac{L_t}{2} + \frac{\kappa_b}{3} + \frac{\kappa_t}{3}\right)^{-1}\left(\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| - \left(\kappa_g + \frac{2\kappa_b}{3} + \frac{\kappa_t}{6}\right)\epsilon\right) \leq \|\mathbf{s}_k\|^3. \tag{34}$$

Choosing $\kappa_g = \frac{1}{4}$, $\kappa_b = \frac{1}{4}$, $\kappa_t = \frac{1}{2}$,

$$\left(\sigma_k + \theta + \frac{L_t}{2} + \frac{1}{4}\right)^{-1}\left(\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| - \frac{1}{2}\epsilon\right) \leq \|\mathbf{s}_k\|^3. \tag{35}$$

□

We now prove that the length of the step $\mathbf{s}_k$ can not be arbitrarily small compared to $\chi_{f,2}$. We first need an additional lemma that relates the step length $\mathbf{s}_k$ to the second criticality measure $\chi_{f,2}$. Proving such result requires the following auxiliary lemma.

Lemma 14. *Suppose that Condition 1 holds. For all* $\mathbf{x}_k, \mathbf{s} \in \mathbb{R}^d$,

$$\|\nabla^2 f(\mathbf{x}_k + \mathbf{s}) - \nabla^2_{\mathbf{s}} \phi_k(\mathbf{s})\| \leq \left(\frac{L_t}{2} + \frac{\kappa_t}{2}\right) \|\mathbf{s}\|^2 + \left(\kappa_b + \frac{\kappa_t}{2}\right) \epsilon_2. \tag{36}$$

Proof. Recall that

$$\nabla^2_{\mathbf{s}} \phi_k(\mathbf{s}) = \mathbf{B}_k + \mathbf{T}_k[\mathbf{s}], \tag{37}$$

therefore

$$\begin{aligned}
&\|\nabla^2 f(\mathbf{x}_k + \mathbf{s}) - \nabla^2_{\mathbf{s}} \phi_k(\mathbf{s})\| \\
&= \|\nabla^2 f(\mathbf{x}_k + \mathbf{s}) - \mathbf{B}_k - \mathbf{T}_k[\mathbf{s}]\| \\
&\leq \|\nabla^2 f(\mathbf{x}_k + \mathbf{s}) - \nabla^2 f(\mathbf{x}_k) - \nabla^3 f(\mathbf{x}_k)[\mathbf{s}]\| + \|\nabla^2 f(\mathbf{x}_k) - \mathbf{B}_k\| \\
&+ \|\nabla^3 f(\mathbf{x}_k)[\mathbf{s}] - \mathbf{T}_k[\mathbf{s}]\| \\
&\leq \frac{L_t}{2}\|\mathbf{s}\|^2 + \kappa_b \epsilon_2 + \kappa_t \epsilon_2^{1/2} \|\mathbf{s}\|,
\end{aligned} \tag{38}$$

where the last inequality uses Lemma 32 and Condition 1 with $\epsilon_2 = \epsilon^{2/3}$.

We again apply the Young's inequality for products stated in Eq. 32 which yields

$$\|\nabla^2 f(\mathbf{x}_k + \mathbf{s}) - \nabla^2_{\mathbf{s}} \phi_k(\mathbf{s})\| \leq \frac{L_t}{2}\|\mathbf{s}\|^2 + \kappa_b \epsilon_2 + \frac{\kappa_t}{2}\|\mathbf{s}\|^2 + \frac{\kappa_t}{2}\epsilon_2. \tag{39}$$

□

Lemma 15 (Lemma 4.3 restated)**.** *Suppose that Condition 1 holds with the choice* $\kappa_g = \frac{1}{4}$, $\kappa_b = \frac{1}{4}$, $\kappa_t = \frac{1}{2}$. *For any* $\mathbf{x}_k \in \mathbb{R}^d$, *the length of the step* $\mathbf{s}_k$ *(satisfying Cond. 2) is such that*

$$\|\mathbf{s}_k\| \geq \kappa_{k,2}^{-1/2} \left(\chi_{f,2}(\mathbf{x}_k + \mathbf{s}_k) - \frac{1}{2}\epsilon_2\right)^{1/2}, \tag{40}$$

where $\kappa_{k,2} = \left(3\sigma_k + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$.

Proof. Using the definition of the model in Eq. (4) and the fact that $\min_{\mathbf{z}}[a(\mathbf{z}) + b(\mathbf{z})] \geq \min_{\mathbf{z}}[a(\mathbf{z})] + \min_{\mathbf{z}}[b(\mathbf{z})]$, we find that

$$\begin{aligned}
\lambda_{min}(\nabla^2 f(\mathbf{x}_k + \mathbf{s}_k)) &= \min_{\|\mathbf{y}\|=1} \nabla^2 f(\mathbf{x}_k + \mathbf{s}_k)[\mathbf{y}]^2 \\
&= \min_{\|\mathbf{y}\|=1} \left(\nabla^2 f(\mathbf{x}_k + \mathbf{s}_k) - \nabla^2_{\mathbf{s}} \phi_k(\mathbf{s}_k) - \frac{\sigma_k}{4} \nabla^2_{\mathbf{s}} \|\mathbf{s}_k\|^4 + \nabla^2_{\mathbf{s}} m_k(\mathbf{s}_k)\right)[\mathbf{y}]^2 \\
&\geq \min_{\|\mathbf{y}\|=1} \left(\nabla^2 f(\mathbf{x}_k + \mathbf{s}_k) - \nabla^2_{\mathbf{s}} \phi_k(\mathbf{s}_k)\right)[\mathbf{y}]^2 + \frac{\sigma_k}{4} \min_{\|\mathbf{y}\|=1} \left(-\nabla^2_{\mathbf{s}} \|\mathbf{s}_k\|^4\right)[\mathbf{y}]^2 \\
&\quad + \min_{\|\mathbf{y}\|=1} \nabla^2_{\mathbf{s}} m_k(\mathbf{s}_k)[\mathbf{y}]^2
\end{aligned} \tag{41}$$

Considering each term in turn, and using Lemma 14, we see that

$$\begin{aligned}
&\min_{\|\mathbf{y}\|=1} \left(\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k) - \nabla^2_{\mathbf{s}}\phi_k(\mathbf{s}_k)\right)[\mathbf{y}]^2 \\
&\quad\geq \min_{\|\mathbf{y}_1\|=\|\mathbf{y}_2\|=1} \left(\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k) - \nabla^2_{\mathbf{s}}\phi_k(\mathbf{s}_k)\right)[\mathbf{y}_1,\mathbf{y}_2] \\
&\quad\geq -\max_{\|\mathbf{y}_1\|=\|\mathbf{y}_2\|=1} \left|\left(\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k) - \nabla^2_{\mathbf{s}}\phi_k(\mathbf{s}_k)\right)[\mathbf{y}_1,\mathbf{y}_2]\right| \\
&\quad= -\|\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k) - \nabla^2_s\phi_k(\mathbf{s}_k)\|_{[2]} \\
&\quad\overset{(36)}{\geq} -\left(\frac{L_t}{2}+\frac{\kappa_t}{2}\right)\|\mathbf{s}_k\|^2 - \left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2. \qquad (42)
\end{aligned}$$

Using Eq. (25), we get that $\frac{1}{4}\nabla^2_{\mathbf{s}}\left(\|\mathbf{s}_k\|^4\right)[\mathbf{y}]^2 = 2(\mathbf{s}_k^\top\mathbf{y})^2 + \|\mathbf{s}_k\|^2\|\mathbf{y}\|^2$, therefore

$$\begin{aligned}
&\min_{\|\mathbf{y}\|=1} \left(-\nabla^2_{\mathbf{s}}(\|\mathbf{s}_k\|^4)\right)[\mathbf{y}]^2 \\
&\qquad= -\max_{\|\mathbf{y}\|=1} \nabla^2_{\mathbf{s}}(\|\mathbf{s}_k\|^4)[\mathbf{y}]^2 \\
&\qquad= -12\|\mathbf{s}_k\|^2. \qquad (43)
\end{aligned}$$

From Eq. (14), we have $\min_{\|\mathbf{y}\|=1}\nabla^2_s m_k(\mathbf{s}_k)[\mathbf{y}]^2 = \lambda_{min}(\nabla^2_{\mathbf{s}} m_k(\mathbf{s}_k))$. Combined with the last two equations, we get that

$$\begin{aligned}
-\lambda_{min}(\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k)) \leq &\left(\frac{L_t}{2}+\frac{\kappa_t}{2}\right)\|\mathbf{s}_k\|^2 \\
&+\left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2 + 3\sigma_k\|\mathbf{s}_k\|^2 - \min[0,\lambda_{min}(\nabla^2_{\mathbf{s}} m_k(\mathbf{s}_k))]. \qquad (44)
\end{aligned}$$

As the right hand side of the above equation is non-negative, we can rewrite Eq. (44) as

$$\begin{aligned}
\max[0,-\lambda_{min}(\nabla^2 f(\mathbf{x}_k+\mathbf{s}_k))] \leq &\left(\frac{L_t}{2}+\frac{\kappa_t}{2}+3\sigma_k\right)\|\mathbf{s}_k\|^2 + \left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2 \\
&+\max[0,-\lambda_{min}(\nabla^2_{\mathbf{s}} m_k(\mathbf{s}_k))]. \qquad (45)
\end{aligned}$$

Combining the above with Eq. (13) and Eq. (14), and with Eq. (15) for $i=2$, we conclude

$$\begin{aligned}
\chi_{f,2}(\mathbf{x}_k+\mathbf{s}_k) &\leq \left(\frac{L_t}{2}+\frac{\kappa_t}{2}+3\sigma_k\right)\|\mathbf{s}_k\|^2 + \left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2 + \chi_{m,2}(\mathbf{x}_k,\mathbf{s}_k) \\
&\leq \left(\frac{L_t}{2}+\frac{\kappa_t}{2}+3\sigma_k+\theta\right)\|\mathbf{s}_k\|^2 + \left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2, \qquad (46)
\end{aligned}$$

which implies

$$\left(\frac{L_t}{2}+\frac{\kappa_t}{2}+3\sigma_k+\theta\right)\|\mathbf{s}_k\|^2 \geq \chi_{f,2}(\mathbf{x}_k+\mathbf{s}_k) - \left(\kappa_b+\frac{\kappa_t}{2}\right)\epsilon_2. \qquad (47)$$

Choosing $\kappa_b = \frac{1}{4}, \kappa_t = \frac{1}{2}$, we conclude

$$\|\mathbf{s}_k\|^2 \geq \left(3\sigma_k+\frac{L_t}{2}+\theta+\frac{1}{4}\right)^{-1}\left(\chi_{f,2}(\mathbf{x}_k+\mathbf{s}_k) - \frac{1}{2}\epsilon_2\right). \qquad (48)$$

□

We now derive an upper bound on the regularization parameter σ_k. The proof is conceptually similar to Lemma 3.3 in [17].

Lemma 16. *Let Assumption 1 hold and assume Condition 2 holds. Also assume that*

$$\sigma_k > \hat{\sigma}_{sup} := \max\left(\frac{4\xi}{(1-\eta_2)}, \frac{\xi\,(4L_t + 2 + 8\theta)}{(1-\eta_2)\,\epsilon - 8\xi}\right), \tag{49}$$

where $\xi := \left(\epsilon\kappa_g + \frac{\epsilon^{2/3}\kappa_b}{2} + \frac{\epsilon^{1/3}\kappa_t + L_t}{6}\right)$. *Then iteration* k *is very successful and consequently* $\sigma_k \leq \gamma_3\hat{\sigma}_{sup} := \sigma_{max}$ *for all* k.

Proof. First, note that

$$\rho_k > \eta_2 \iff r_k := f(\mathbf{x}_k + \mathbf{s}_k) - f(\mathbf{x}_k) - \eta_2(\phi_k(\mathbf{s}_k) - f(\mathbf{x}_k)) < 0. \tag{50}$$

We rewrite r_k as

$$r_k = f(\mathbf{x}_k + \mathbf{s}_k) - \phi_k(\mathbf{s}_k) + (1-\eta_2)(\phi_k(\mathbf{s}_k) - f(\mathbf{x}_k)). \tag{51}$$

From the mean value theorem,

$$f(\mathbf{x}_k + \mathbf{s}_k) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^\intercal \nabla^2 f(\mathbf{x}_k)\mathbf{s}_k + \frac{1}{6}\nabla^3 f(\mathbf{x}_k + \alpha\mathbf{s}_k)[\mathbf{s}_k]^3 \tag{52}$$

for some $\alpha \in (0,1)$. Therefore we can bound the first term in r_k as

$$\begin{aligned}
f(\mathbf{x}_k + \mathbf{s}_k) - \phi_k(\mathbf{s}_k) &= (\nabla f(\mathbf{x}_k) - \mathbf{g}_k)^\intercal \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^\intercal \left(\nabla^2 f(\mathbf{x}_k) - B_k\right)\mathbf{s}_k \\
&\quad + \frac{1}{6}\left(\nabla^3 f(\mathbf{x}_k + \alpha\mathbf{s}_k) - \mathbf{T}_k\right)[\mathbf{s}_k]^3 \\
&= (\nabla f(\mathbf{x}_k) - \mathbf{g}_k)^\intercal \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^\intercal \left(\nabla^2 f(\mathbf{x}_k) - B_k\right)\mathbf{s}_k \\
&\quad + \frac{1}{6}\left(\nabla^3 f(\mathbf{x}_k) - \mathbf{T}_k\right)[\mathbf{s}_k]^3 + \frac{1}{6}\left(\nabla^3 f(\mathbf{x}_k + \alpha\mathbf{s}_k) - \nabla^3 f(\mathbf{x}_k)\right)[\mathbf{s}_k]^3 \\
&\leq \kappa_g\epsilon\|\mathbf{s}_k\| + \frac{1}{2}\kappa_b\epsilon^{2/3}\|\mathbf{s}_k\|^2 + \frac{1}{6}\left(\kappa_t\epsilon^{1/3}\right)\|\mathbf{s}_k\|^3 + \frac{L_t}{6}\|\mathbf{s}_k\|^4 \\
&\leq \underbrace{\left(\epsilon\kappa_g + \frac{\epsilon^{2/3}\kappa_b}{2} + \frac{\epsilon^{1/3}\kappa_t + L_t}{6}\right)}_{:=\xi}\max(\|\mathbf{s}_k\|, \|\mathbf{s}_k\|^4)
\end{aligned}$$

Given that Condition 2 holds per assumption, we can combine Eq. (18) from Lemma 2 with the above Eq. (53) to derive the upper bound σ_{sup} as follows.

Case I: If $\|\mathbf{s}_k\| \geq 1$ we have

$$r_k < 0 \iff \sigma_k > \frac{4\xi}{(1-\eta_2)}. \tag{53}$$

Case II: If $\|\mathbf{s}_k\| < 1$ we have

$$r_k < 0 \iff \sigma_k > \frac{4\xi}{(1-\eta_2)\|s_k\|^3}. \tag{54}$$

We need to further simplify the RHS in the equation above that contains the term $\|s_k\|^3$. To do so, we first use Lemma 3 to upper bound the right hand side as

$$\frac{4\xi}{(1-\eta_2)\|s_k\|^3} \leq \frac{4\xi\kappa_k}{(1-\eta_2)(\|\nabla f(\mathbf{x}_k+\mathbf{s}_k)\|-\frac{1}{2}\epsilon)} \leq \frac{2\cdot 4\xi\left(\sigma_k+\frac{L_t}{2}+\theta+\frac{1}{4}\right)}{(1-\eta_2)\epsilon}. \tag{55}$$

If σ_k is greater than the upper bound in Eq. (55), it is also greater than the RHS in Eq. (54). Consequently $\rho_k > \eta_2$ as soon as

$$\sigma_k > \frac{\xi\left(4L_t+2+8\theta\right)}{\left(1-\eta_2\right)\epsilon-8\xi} \tag{56}$$

As a result, we conclude that if $\sigma_0 < \hat{\sigma}_{sup}$, then $\sigma_k \leq \gamma_3\hat{\sigma}_{sup}$ for all k, where

$$\hat{\sigma}_{sup} := \max\left(\frac{4\xi}{(1-\eta_2)}, \frac{\xi\left(4L_t+2+8\theta\right)}{\left(1-\eta_2\right)\epsilon-8\xi}\right) \tag{57}$$

□

Lemma 17 (Lemma 4.4 restated)**.** *The steps produced by Algorithm 1 guarantee that if $\sigma_k \leq \sigma_{max}$ for $\sigma_{max} > 0$, then $k \leq C(\gamma_1, \gamma_2, \sigma_{max}, \sigma_0)$ where*

$$C(\gamma_1,\gamma_2,\sigma_{max},\sigma_0) := \left(1+\frac{|\log\gamma_1|}{\log\gamma_2}\right)|\mathcal{S}_k| + \frac{1}{\log\gamma_2}\log\left(\frac{\sigma_{max}}{\sigma_0}\right).$$

Proof. From the updates in Eq. (11), we get that for each two consecutive iterations

$$\begin{aligned}\gamma_1\sigma_j \leq \max(\gamma_1\sigma_j,\sigma_{min}) \leq \sigma_{j+1}, \quad j\in\mathcal{S}_k\\ \gamma_2\sigma_j \leq \sigma_{j+1}, \quad j\in\mathcal{U}_k,\end{aligned} \tag{58}$$

where $\sigma_{min} \leq \sigma_k\ \forall k$.

Thus we deduce inductively that

$$\sigma_0\gamma_1^{|\mathcal{S}_k|}\gamma_2^{|\mathcal{U}_k|} \leq \sigma_k. \tag{59}$$

We therefore obtain, using the bound $\sigma_k \leq \sigma_{max}$. that

$$|\mathcal{S}_k|\log\gamma_1 + |\mathcal{U}_k|\log\gamma_2 \leq \log\left(\frac{\sigma_{max}}{\sigma_0}\right), \tag{60}$$

which then implies that

$$|\mathcal{U}_k| \leq -|\mathcal{S}_k|\frac{\log\gamma_1}{\log\gamma_2} + \frac{1}{\log\gamma_2}\log\left(\frac{\sigma_{max}}{\sigma_0}\right) \tag{61}$$

Finally using the equality $k = |\mathcal{S}_k| + |\mathcal{U}_k|$ and the fact that $\log\gamma_1 < 0$ since $\gamma_1 < 1$ concludes the proof. □

Finally, we are ready to state the main result in this section that establishes a worst-case complexity rate to reach an (ϵ_1, ϵ_2)-second-order critical point.

Theorem 18 (Worst-case complexity, Theorem 4.5 restated)**.** *Let f_{low} be a lower bound on f and assume Conditions 1 and 2 hold. Let $\kappa_s = \left(\sigma_{max} + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$, $\kappa_{s,2} = \left(3\sigma_{max} + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$ and $\kappa_{max} = \max(\sqrt[3]{2}\kappa_s^{4/3}, 2\kappa_{s,2}^2)$. Then, given $\epsilon_1, \epsilon_2 > 0$, Algorithm 1 needs at most*

$$\left\lceil \mathcal{K}_{succ}(\epsilon) := \frac{8\kappa_{max}(f(\mathbf{x}_0) - f_{low})}{\eta_1 \sigma_{min}} \max(\epsilon_1^{-4/3}, \epsilon_2^{-2}) \right\rceil$$

successful iterations and

$$\mathcal{K}_{outer}(\epsilon) := \lceil C(\gamma_1, \gamma_2, \sigma_{max}, \sigma_0) \cdot \mathcal{K}_{succ}(\epsilon) \rceil. \tag{62}$$

total outer iterations to reach an iterate $\mathbf{x}^*$ such that both $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon_1$ and $\lambda_{min}(\nabla^2 f(\mathbf{x}^*)) \geq -\epsilon_2$.

Proof. First, let $\kappa_s = \left(\sigma_{max} + \frac{L_t}{2} + \theta + \frac{1}{4}\right)$. For each successful iteration k, the function decrease in terms of the first-order criticality measure is

$$\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \eta_1 (f(\mathbf{x}_k) - \phi_k(\mathbf{s}_k)) \\
&\overset{(18)}{\geq} \frac{1}{4}\eta_1 \sigma_{min} \|\mathbf{s}_k\|_2^4 \\
&\overset{(19)}{\geq} \frac{1}{4}\eta_1 \sigma_{min} \kappa_k^{-4/3} \left(\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| - \frac{1}{2}\epsilon_1 \right)^{4/3} \\
&\geq \frac{1}{4}\eta_1 \sigma_{min} \kappa_s^{-4/3} \left(\frac{1}{2}\epsilon_1 \right)^{4/3} \\
&\geq \frac{1}{8\sqrt[3]{2}}\eta_1 \sigma_{min} \kappa_s^{-4/3} \epsilon_1^{4/3}
\end{aligned} \tag{63}$$

where the fourth inequality uses the fact that $\|\nabla f(\mathbf{x}_k + \mathbf{s}_k)\| \geq \epsilon_1$ before termination.

Let's now consider the function decrease in terms of the second-order criticality measure. First, let $\kappa_{s,2} = \left(3\sigma_{max}\frac{L_t}{2} + \frac{1}{4}\right)$. For each successful iteration k, we have

$$\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \eta_1 (f(\mathbf{x}_k) - \phi_k(\mathbf{s}_k)) \\
&\overset{(18)}{\geq} \frac{1}{4}\eta_1 \sigma_{min} \|\mathbf{s}_k\|_2^4 \\
&\overset{(20)}{\geq} \frac{1}{4}\eta_1 \sigma_{min} \kappa_{k,2}^{-2} \left(\chi_{f,2}(\mathbf{x}_k + \mathbf{s}_k) - \frac{1}{2}\epsilon_2 \right)^2 \\
&\geq \frac{1}{4}\eta_1 \sigma_{min} \kappa_{k,2}^{-2} \left(\epsilon_2 - \frac{1}{2}\epsilon_2 \right)^2 \\
&\geq \frac{1}{4^2}\eta_1 \sigma_{min} \kappa_{s,2}^{-2} \epsilon_2^2
\end{aligned} \tag{64}$$

where the fourth inequality uses the fact that $\chi_{f,2}(\mathbf{x}_k + \mathbf{s}_k) \geq \epsilon_2$ before termination.

Thus on any successful iteration until termination we can guarantee the minimal of the two decreases in Eqs. (63) and (64), and hence,

$$f(\mathbf{x}_0) - f(\mathbf{x}_{k+1}) \geq \frac{1}{8}\eta_1 \sigma_{min} \min\left(\frac{\kappa_s^{-4/3}}{\sqrt[3]{2}}, \frac{\kappa_{s,2}^{-2}}{2} \right) \min(\epsilon_1^{4/3}, \epsilon_2^2) |\mathcal{S}_k| \tag{65}$$

Using that f is bounded below by f_{low}, we conclude

$$|\mathcal{S}_k| \leq \frac{8\max(\sqrt[3]{2}\kappa_s^{4/3}, 2\kappa_{s,2}^2)(f(\mathbf{x}_0) - f_{low})}{\eta_1 \sigma_{min}} \max(\epsilon_1^{-4/3}, \epsilon_2^{-2}). \tag{66}$$

Finally, we can use Lemma 5 to get a bound on the total number of iterations.

□

B Sampling conditions

In this section, we prove how to ensure that the sampling conditions in Eqs. (6)-(8) are satisfied. We discuss two cases: i) random sampling *without* replacement, and ii) sampling with replacement. Since sampling *without* replacement yields a lower sample complexity, we directly use the corresponding results in the main part of the paper.

B.1 Sampling without replacement

We prove that one can choose the size of the sample sets in order to satisfy the three sampling conditions presented in Eqs. (6)-(8). The proof consists in using concentration inequalities to prove that there exists a sample size such that the sampled quantity is close enough to the expected value. We will rely on two key results: the first one is the Matrix Hoeffding-Serfling Inequality derived in [50] while the second result is a tensor inequality which we name Tensor Hoeffding-Serfling inequality (see Theorem 19 below). To the best of our knowledge, the latter result is new and is based on an adaption of a result derived in [45].

B.1.1 Existing results

Theorem 19 (Matrix Hoeffding-Serfling Inequality [50])**.** *Let $\mathcal{A} := \{\mathbf{A}_1, \cdots, \mathbf{A}_N\}$ be a collection of real-valued matrices in $\mathbb{R}^{d_1 \times d_2}$ with bounded spectral norm, i.e., $\|\mathbf{A}_i\| \leqslant \sigma$ for all $i = 1, \ldots, N$ and some $\sigma > 0$. Let $\mathbf{X}_1, \cdots, \mathbf{X}_n$ be $n < N$ samples from $\mathcal{A}$ under the sampling without replacement. Denote $\mu := \frac{1}{N}\sum_{i=1}^N \mathbf{A}_i$. Then, for any $t > 0$,*

$$P\left(\left\|\frac{1}{n}\sum_{i=1}^n \mathbf{X}_i - \mu\right\| \geqslant t\right) \leq (d_1 + d_2)\exp\left(-\frac{nt^2}{8\sigma^2(1 + 1/n)(1 - n/N)}\right).$$

We will also need the following lemma which is derived in [6].

Lemma 20 ([6])**.** *Let $\mathcal{A} := \{y_1, \ldots y_N\}$ be a finite population of N points in $\mathbb{R}$ and $Y_1, \ldots, Y_n$ be a random sample drawn without replacement from $\mathcal{A}$. Define $X_n = \frac{1}{n}\sum_{k=1}^n (Y_k - \mu)$ where $\mu = \frac{1}{N}\sum_{i=1}^N y_i$. Assume that $a \leq Y_k \leq b$, then or any $s > 0$, it holds that*

$$\log \mathbb{E}\exp(sX_n) \leq \frac{(b-a)^2}{8}\frac{s^2}{n^2}(n+1)\left(1 - \frac{n}{N}\right).$$

B.1.2 Concentration bound for sum of i.i.d. tensors

We now extend Theorem 19 to the more general case where we consider a collection of real-valued tensors instead of matrices. Formally, let $(\Omega, \mathcal{F}, P)$ be a probability space and let $\mathcal{X}$ be a real (m, d) random tensor, i.e. a measurable map from Ω to $\mathbb{T}_{m,d}$ (the space of real tensors of order m and dimension d).

Our goal is to derive a concentration bound for a sum of n tensors

$$\mathcal{X} = \sum_{i=1}^{n} \mathcal{Y}_i,$$

where each $\mathcal{Y}_i$ is sampled without replacement from a population $\mathcal{A}$ of size N.

The concentration result derived in this section is based on the proof technique introduced in [45] which we adapt for sums of random variables. Formally, we consider a tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ of order k whose spectral norm is defined as

$$\|\mathcal{X}\| = \sup_{\substack{\mathbf{u}_1, \dots \mathbf{u}_k \\ \|\mathbf{u}_i\|=1}} \mathcal{X}(\mathbf{u}_1, \dots \mathbf{u}_k). \tag{67}$$

Note that for symmetric tensors, the spectral norm is simply equal to $\|\mathcal{X}\| = \sup_{\mathbf{u}|\|\mathbf{u}\|=1} \mathcal{X}(\mathbf{u}, \dots \mathbf{u})$ [4].

Proof idea In the following, we first provide a concentration bound for a fixed set of unit-length vectors $\mathbf{u}_1, \dots, \mathbf{u}_k$. (Lemma 21). We then extend this result to arbitrary vectors on the unit sphere in order to obtain a concentration bound for the tensor $\mathcal{X}$ by using a covering argument similar to [45].

Lemma 21. *Let $\mathcal{X}$ be a sum of n i.i.d. tensors $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ sampled without replacement from a finite population $\mathcal{A}$ of size N. Consider a fixed set of vectors $\mathbf{u}_1, \dots \mathbf{u}_k$ such that $\|\mathbf{u}_i\| = 1$ and assume that for each tensor i, $a \leq \mathcal{Y}_i(\mathbf{u}_1, \dots, \mathbf{u}_k) \leq b$. Let $\sigma := (b - a)$, then we have*

$$P\left(|\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k)]| \geq t\right) \leq 2 \exp\left(-\frac{2t^2 n^2}{\sigma^2 (n+1)(1 - n/N)}\right).$$

Proof. By Markov's inequality and Hoeffding's lemma, we have

$$\begin{aligned}
P\left(\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k)] \geq t\right) &= P\left(e^{s(\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k)])} \geq e^{st}\right) \\
&\leq e^{-st} \mathbb{E}\left[e^{\left(s(\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k)]\right)}\right] \\
&= e^{-st} \mathbb{E}\left[e^{\left(s(\sum_i \mathcal{Y}_i(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\sum_i \mathcal{Y}_i(\mathbf{u}_1, \dots, \mathbf{u}_k)]\right)}\right] \\
&\overset{(i)}{\leq} \exp\left(-st + \frac{\sigma^2}{8} \frac{s^2}{n^2} (n+1)\left(1 - \frac{n}{N}\right)\right),
\end{aligned}$$

where (i) follows from Lemma 20.

After minimizing over s, we obtain

$$P\left(\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k)] \geq t\right) \leq \exp\left(-\frac{2t^2 n^2}{\sigma^2 (n+1)(1 - n/N)}\right).$$

By symmetry, one can easily show that

$$P(\mathcal{X}(\mathbf{u}_1, \dots, \mathbf{u}_k) \leq -t) \leq \exp\left(-\frac{2t^2 n^2}{\sigma^2 (n+1)(1 - n/N)}\right).$$

We then complete the proof by taking the union of both cases. □

Theorem 22 (Theorem 4.6 restated)**.** *Let $\mathcal{X}$ be a sum of n tensors $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ sampled without replacement from a finite population $\mathcal{A}$ of size N. Let $\mathbf{u}_1, \dots \mathbf{u}_k$ be such that $\|\mathbf{u}_i\| = 1$ and assume that for each tensor i, $a \leq \mathcal{Y}_i(\mathbf{u}_1, \dots, \mathbf{u}_k) \leq b$. Let $\sigma := (b-a)$, then we have*

$$P(\,\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) \leq k_0^{(\sum_{i=1}^k d_i)} \cdot 2\exp\left(-\frac{t^2 n^2}{2\sigma^2(n+1)(1-n/N)}\right),$$

where $k_0 = \left(\frac{2k}{\log(3/2)}\right)$.

Proof. We use the same covering number argument as in [45]. The main idea is to create an ϵ-net of countable size to cover the space $S^{d_1-1}, \dots, S^{d_k-1}$. Formally, let $C_1, \dots, C_k$ be ϵ-covers of $S^{d_1-1}, \dots, S^{d_k-1}$. Then since $S^{d_1-1} \times \cdots \times S^{d_k-1}$ is compact, there exists a maximizer $(\mathbf{u}_1^*, \dots, \mathbf{u}_k^*)$ of (67). Using the ϵ-covers, we have

$$\|\mathcal{X}\| = \mathcal{X}(\bar{\mathbf{u}}_1 + \delta_1, \dots, \bar{\mathbf{u}}_k + \delta_k),$$

where $\bar{\mathbf{u}}_i \in C_i$ and $\|\delta_i\| \leq \epsilon$ for $i = 1, \dots, k$.

Now

$$\|\mathcal{X}\| \leq \mathcal{X}(\bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_k) + \left(\epsilon k + \epsilon^2 \binom{k}{2} + \cdots \epsilon^k \binom{k}{k}\right) \|\mathcal{X}\|.$$

Take $\epsilon = \frac{\log(3/2)}{k}$ then the sum inside the parenthesis can be bounded as follows:

$$\epsilon k + \epsilon^2 \binom{k}{2} + \cdots \epsilon^k \binom{k}{k} \leq \epsilon k + \frac{(\epsilon k)^2}{2!} + \cdots \frac{(\epsilon k)^k}{k!} \leq e^{\epsilon k} - 1 = \frac{1}{2}.$$

Thus we have

$$\|\mathcal{X}\| \leq 2 \max_{\bar{\mathbf{u}}_1 \in C_1, \dots, \bar{\mathbf{u}}_k \in C_k} \mathcal{X}(\bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_k).$$

So far, all the steps have been identical to the proof in [45]. We conclude the proof with one last step that is a simple adaptation of the proof in [45], combined with the result of Lemma 21.

Since the ϵ-covering number $|C_k|$ can be bounded by $\epsilon/2$-packing number, which can be bounded by $(2/\epsilon)^{d_k}$, using the union bound. Therefore,

$$\begin{aligned} P(\,\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) &\leq \sum_{\bar{\mathbf{u}}_1 \in C_1, \dots, \bar{\mathbf{u}}_k \in C_k} P\left(\mathcal{X}(\bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_k) - \mathbb{E}[\mathcal{X}(\bar{\mathbf{u}}_1, \dots, \bar{\mathbf{u}}_k)] \geq \frac{t}{2}\right) \\ &\leq k_0^{\sum_{i=1}^k d_i} \cdot 2\exp\left(-\frac{t^2 n^2}{2\sigma^2(n+1)(1-n/N)}\right). \end{aligned}$$

□

Lemma 23 (Lemma 4.7 restated)**.** *Consider the sub-sampled gradient, Hessian and third-order tensor defined in Eq.* (5)*. Under Assumption 1, the sampling conditions in Eqs.* (6)*,* (7) *and* (8) *are satisfied with probability $1 - \delta$, $\delta \in (0, 1)$ for the following choice of the size of the sample sets $\mathcal{S}^g, \mathcal{S}^b$ and $\mathcal{S}^t$:*

$$\begin{aligned} n_g &= \tilde{\mathcal{O}}\left(\kappa_g^2 \epsilon^2 \,/ L_f^2 + 1\,/N\right)^{-1}, \\ n_b &= \tilde{\mathcal{O}}\left(\kappa_b^2 \epsilon^{4/3} \,\Big/ L_g^2 + 1\,/N\right)^{-1}, \\ n_t &= \tilde{\mathcal{O}}\left(\kappa_t^2 \epsilon^{2/3} \,\Big/ L_b^2 + 1\,/N\right)^{-1}, \end{aligned}$$

where $\tilde{\mathcal{O}}$ hides poly-logarithmic factors and a polynomial dependency to d.

Proof. **Gradient**

Let $n_g := |\mathcal{S}^g|$ and note that by the triangle inequality as well as Lipschitz continuity of ∇f (Assumption 1) we have

$$\|\mathbf{g}(\mathbf{x})\| = \frac{1}{n_g} \sum_{i \in \mathcal{S}^g} \|\nabla f_i(\mathbf{x})\| \leq L_g := \sigma_g \tag{68}$$

We then apply Theorem 19 on the gradient vector and require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0, 1]$.

$$P\left(\|\mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})\| > t\right) \leq 2d \exp\left(-\frac{n_g t^2}{8\sigma_g^2(1 + 1/n_g)(1 - n_g/N)}\right) \overset{!}{\leq} \delta \tag{69}$$

Taking the log on both side, we get

$$-\frac{n_g t^2}{8\sigma_g^2(1 + 1/n_g)(1 - n_g/N)} \overset{!}{\leq} \log \frac{\delta}{2d}, \tag{70}$$

which implies

$$\begin{aligned} n_g t^2 &\overset{!}{\geq} \log \frac{2d}{\delta} \left(8\sigma_g^2(1 + 1/n_g)(1 - n_g/N)\right) \\ &= \log \frac{2d}{\delta} \left(8\sigma_g^2 \left(1 + 1/n_g - n_g/N - 1/N\right)\right) \end{aligned} \tag{71}$$

Since $\frac{1}{n_g} - \frac{1}{N} < 1$, we instead require the following simpler condition,

$$\begin{aligned} & n_g t^2 \geq \log \frac{2d}{\delta} \left(8\sigma_g^2 \left(2 - n_g/N\right)\right) \\ \implies & n_g \cdot \left(t^2 + \log \frac{2d}{\delta} \frac{8\sigma_g^2}{N}\right) \geq \log \frac{2d}{\delta} \left(16\sigma_g^2\right) \\ \implies & n_g \geq \frac{16\sigma_g^2 \log \frac{2d}{\delta}}{\left(t^2 + \frac{8\sigma_g^2}{N} \log \frac{2d}{\delta}\right)} \end{aligned} \tag{72}$$

Finally, we can simply choose $t = \kappa_g \epsilon$ in order to satisfy Eq. (6).

Hessian

Again, let $n_b := |\mathcal{S}^b|$ and note that by the triangle inequality as well as Lipschitz continuity of $\nabla^2 f$ (Assumption 1) we have

$$\|\mathbf{B}(\mathbf{x})\| \leq \frac{1}{n_b} \sum_{i \in \mathcal{S}^b} \|\nabla^2 f_i(\mathbf{x})\| \leq L_g := \sigma_b$$

Now we apply the matrix Hoeffding's inequality stated in Theorem 19 on the Hessian and require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0, 1]$.

$$P\left(\left\|\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x})\right\| > t\right) \leq 2d \exp\left(-\frac{n_b t^2}{8\sigma_b^2(1 + 1/n_b)(1 - n_b/N)}\right) \overset{!}{\leq} \delta \tag{73}$$

Taking the log on both side, we get

$$-\frac{n_b t^2}{8\sigma_b^2(1 + 1/n_b)(1 - n_b/N)} \leq \log \frac{\delta}{2d}, \tag{74}$$

which implies

$$\begin{aligned} n_b t^2 &\geq \log \frac{2d}{\delta}(8\sigma_b^2(1+1/n_b)(1-n_b/N)) \\ &= \log \frac{2d}{\delta}\left(8\sigma_b^2\left(1+1/n_b - n_b/N - 1/N\right)\right) \end{aligned} \tag{75}$$

Since $\frac{1}{n_b} - \frac{1}{N} < 1$, we instead require the following simpler condition,

$$\begin{aligned} & n_b t^2 \geq \log \frac{2d}{\delta}\left(8\sigma_b^2\left(2 - n_b/N\right)\right) \\ \implies & n_b \cdot \left(t^2 + \log \frac{2d}{\delta}\frac{8\sigma_b^2}{N}\right) \geq \log \frac{2d}{\delta}\left(16\sigma_b^2\right) \\ \implies & n_b \geq \frac{16\sigma_b^2 \log \frac{2d}{\delta}}{\left(t^2 + \frac{8\sigma_b^2}{N}\log \frac{2d}{\delta}\right)} \end{aligned} \tag{76}$$

Finally, we can simply choose $t = \kappa_b \epsilon^{2/3}$ in order to satisfy Eq. (7) since $\forall \mathbf{s} \in \mathbb{R}^d$,

$$\|(\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x}))\mathbf{s}\| \leq \|(\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x}))\| \cdot \|\mathbf{s}\| \leq \kappa_b \epsilon^{2/3}\|\mathbf{s}\|, \tag{77}$$

Third-order derivative Let $n_t := |\mathcal{S}^t|$ and assume that $a \leq \nabla^3 f_i(\mathbf{u}_1, \ldots, \mathbf{u}_k) \leq b$ for all $i \in \{1, \ldots, n\}$ and $(\mathbf{u}_1, \ldots, \mathbf{u}_k) \in \mathbb{R}^{d_1 \times \cdots \times d_k}$. Define $\sigma_t = (b-a)$.

We apply Theorem 7 and require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0, 1]$.

$$P\left(\left\|\mathbf{T}(\mathbf{x}) - \nabla^3 f(\mathbf{x})\right\| > t\right) \leq k_0^{3d} \cdot 2\exp\left(-\frac{n_t^2 t^2}{2\sigma_t^2(n_t+1)(1-n_t/N)}\right) \overset{!}{\leq} \delta \tag{78}$$

Taking the log on both side, we get

$$-\frac{n_t^2 t^2}{2\sigma_t^2(n_t+1)(1-n_t/N)} \leq \log \frac{\delta}{2k_0^{3d}}, \tag{79}$$

which implies

$$\begin{aligned} n_t^2 t^2 &\geq \log \frac{2k_0^{3d}}{\delta}(2\sigma_t^2(n_t+1)(1-n_t/N)) \\ &= \log \frac{2k_0^{3d}}{\delta}\left(2\sigma_t^2\left(n_t + 1 - n_t^2/N - n_t/N\right)\right) \\ \implies n_t t^2 &\geq \log \frac{2k_0^{3d}}{\delta}\left(2\sigma_t^2\left(1 + 1/n_t - n_t/N - 1/N\right)\right) \end{aligned} \tag{80}$$

Since $\frac{1}{n_t} - \frac{1}{N} < 1$, we instead require the following simpler condition,

$$\begin{aligned} & n_t t^2 \geq \log \frac{2k_0^{3d}}{\delta}\left(2\sigma_t^2\left(2 - n_t/N\right)\right) \\ \implies & n_t \cdot \left(t^2 + \log \frac{2k_0^{3d}}{\delta}\frac{2\sigma_t^2}{N}\right) \geq \log \frac{2k_0^{3d}}{\delta} 4\sigma_t^2 \\ \implies & n_t \geq \frac{4\sigma_t^2 \log \frac{2k_0^{3d}}{\delta}}{\left(t^2 + \frac{2\sigma_t^2}{N}\log \frac{2k_0^{3d}}{\delta}\right)}. \end{aligned} \tag{81}$$

Finally, we can simply choose $t = \kappa_t \epsilon^{1/3}$ in order to satisfy Eq. (8) since $\forall \mathbf{s} \in \mathbb{R}^d$,

$$\begin{aligned}\|\mathbf{T}[\mathbf{s}]^2 - \nabla^3 f(\mathbf{x})[\mathbf{s}]^2\| &\leq \|\mathbf{T}[s] - \nabla^3 f(\mathbf{x})[\mathbf{s}]\|\|\mathbf{s}\| \\ &\leq \|\mathbf{T} - \nabla^3 f(\mathbf{x})\|\|\mathbf{s}\|^2 \\ &\leq \kappa_t \epsilon^{1/3} \|\mathbf{s}\|^2. \end{aligned} \tag{82}$$

□

B.2 Sampling with replacement

Similarly to the previous case of sampling without replacement, we prove that one can use random sampling with replacement in order to satisfy the three sampling conditions presented in Eqs. (6), (7) and (8).

First, we introduce some known results and then derive a concentration bound for a sum of i.i.d. tensors usually a similar proof technique as in the previous subsection.

B.2.1 Existing results

The following results can for instance be found in [47, 48].

Theorem 24 (Matrix Hoeffding)**.** *Consider a finite sequence $\{\mathbf{X}_k\}$ of independent, random, self-adjoint matrices with dimension d, and let $\{\mathbf{A}_k\}$ be a sequence of fixed self-adjoint matrices. Assume that each random matrix satisfies*

$$\mathbb{E}\mathbf{X}_k = \boldsymbol{0} \quad \textit{and} \quad \mathbf{X}_k^2 \preccurlyeq \mathbf{A}_k^2 \quad \textit{almost surely.}$$

Then, for all $t \geq 0$,

$$P\left(\lambda_{\max}\left(\sum\nolimits_k \mathbf{X}_k\right) \geq t \leq d \cdot e^{-t^2/8\sigma^2}\right) \quad \textit{where} \quad \sigma^2 := \|\sum\nolimits_k \mathbf{A}_k^2\|.$$

Lemma 25 (Hoeffding's lemma)**.** *Let Z be any real-valued bounded random variable such that $a \leq Z \leq b$. Then, for all $s \in \mathbb{R}$,*

$$\mathbb{E}\left[e^{s(Z - \mathbb{E}(Z))}\right] \leq \exp\left(\frac{s^2(b-a)^2}{8}\right). \tag{83}$$

B.2.2 Concentration bound for sum of i.i.d. tensors

In the following, we provide a concentration bound for sampling with replacement for tensors. This result is based on the proof technique introduced in [45] which we adapt for sums of independent random variables.

Proof idea In the following, we first provide a concentration bound for each entry in the tensor $\mathcal{X}$ (Lemma 26). We then use Lemma 26 to obtain a concentration bound for the tensor $\mathcal{X}$ by using a covering argument similar to [45].

Lemma 26. *Let $\mathcal{X}$ be a sum of n i.i.d. tensors $\mathcal{Y}_i \in \mathbb{R}^{d_1 \times \cdots \times d_k}$. Let $\mathbf{u}_1, \ldots \mathbf{u}_k$ be such that $\|\mathbf{u}_i\| = 1$ and assume that for each tensor i, $a \leq \mathcal{Y}_i(\mathbf{u}_1, \ldots, \mathbf{u}_k) \leq b$. Let $\sigma := (b-a)$, then we have*

$$P\left(|\mathcal{X}(\mathbf{u}_1, \ldots, \mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1, \ldots, \mathbf{u}_k)]| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{n\sigma^2}\right).$$

Proof. By Markov's inequality and Hoeffding's lemma, we have

$$\begin{aligned}
&P\big(\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)] \geq t\big) \\
&\quad = P\left(e^{s(\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)-\mathbb{E}[\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)])} \geq e^{st}\right) \\
&\quad \leq e^{-st}\mathbb{E}\left[e^{(s(\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)-\mathbb{E}[\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)])}\right] \\
&\quad = e^{-st}\mathbb{E}\left[e^{(s(\sum_i \mathcal{Y}_i(\mathbf{u}_1,\dots,\mathbf{u}_k)-\mathbb{E}[\sum_i \mathcal{Y}_i(\mathbf{u}_1,\dots,\mathbf{u}_k)])}\right] \\
&\quad \overset{(i)}{=} e^{-st}\prod_{i=1}^{n}\mathbb{E}\left[e^{(s(\mathcal{Y}_i(\mathbf{u}_1,\dots,\mathbf{u}_k)-\mathbb{E}[\mathcal{Y}_i(\mathbf{u}_1,\dots,\mathbf{u}_k)])}\right] \\
&\quad \overset{(ii)}{\leq} \exp\left(-st + \frac{n\sigma^2 s^2}{8}\right),
\end{aligned}$$

where (i) follows by independence of the $\mathcal{Y}_i$'s and (ii) follows from Lemma 25.

After minimizing over s, we obtain

$$P\left(\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k) - \mathbb{E}[\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k)] \geq t\right) \leq e^{-2t^2/(n\sigma^2)}.$$

Similarly one can show that $P(\mathcal{X}(\mathbf{u}_1,\dots,\mathbf{u}_k) \leq -t) \leq e^{-2t^2/(n\sigma^2)}$. We then complete the proof by taking the union of both cases. □

Theorem 27 (Tensor Hoeffding Inequality)**.** *Let $\mathcal{X}$ be a sum of n i.i.d. tensors $\mathcal{Y}_i \in \mathbb{R}^{d_1\times\cdots\times d_k}$. Let $\mathbf{u}_1,\dots\mathbf{u}_k$ be such that $\|\mathbf{u}_i\| = 1$ and assume that for each tensor i, $a \leq \mathcal{Y}_i(\mathbf{u}_1,\dots,\mathbf{u}_k) \leq b$. Let $\sigma := (b-a)$, then we have*

$$P(\,\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) \leq k_0^{(\sum_{i=1}^k d_i)} \cdot 2\exp\left(-\frac{t^2}{2n\sigma^2}\right),$$

where $k_0 = \left(\frac{2k}{\log(3/2)}\right)$.

Proof. We use the same covering number argument as in [45]. Let $C_1,\dots,C_k$ be ϵ-covers of $S^{d_1-1},\dots,S^{d_k-1}$. Then since $S^{d_1-1}\times\cdots\times S^{d_k-1}$ is compact, there exists a maximizer $(\mathbf{u}_1^*,\dots,\mathbf{u}_k^*)$ of (67). Using the ϵ-covers, we have

$$\|\mathcal{X}\| = \mathcal{X}(\bar{\mathbf{u}}_1+\delta_1,\dots,\bar{\mathbf{u}}_k+\delta_k),$$

where $\bar{\mathbf{u}}_i \in C_i$ and $\|\delta_i\| \leq \epsilon$ for $i = 1,\dots,k$.

Now

$$\|\mathcal{X}\| \leq \mathcal{X}(\bar{\mathbf{u}}_1,\dots,\bar{\mathbf{u}}_k) + \left(\epsilon k + \epsilon^2\binom{k}{2} + \cdots \epsilon^k\binom{k}{k}\right)\|\mathcal{X}\|.$$

Take $\epsilon = \frac{\log(3/2)}{k}$ then the sum inside the parenthesis can be bounded as follows:

$$\epsilon k + \epsilon^2\binom{k}{2} + \cdots \epsilon^k\binom{k}{k} \leq \epsilon k + \frac{(\epsilon k)^2}{2!} + \cdots \frac{(\epsilon k)^k}{k!} \leq e^{\epsilon k} - 1 = \frac{1}{2}.$$

Thus we have

$$\|\mathcal{X}\| \leq 2\max_{\bar{\mathbf{u}}_1\in C_1,\dots,\bar{\mathbf{u}}_k\in C_k} \mathcal{X}(\bar{\mathbf{u}}_1,\dots,\bar{\mathbf{u}}_k).$$

So far, all the steps have been identical to the proof in [45]. We conclude the proof with one last step that is a simple adaptation of the proof in [45], combined with the result of Lemma 26.

Since the ϵ-covering number $|C_k|$ can be bounded by $\epsilon/2$-packing number, which can be bounded by $(2/\epsilon)^{d_k}$, using the union bound. Therefore, by Lemma 26

$$\begin{aligned} P(\,\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) &\leq \sum_{\bar{\mathbf{u}}_1 \in C_1, \ldots, \bar{\mathbf{u}}_k \in C_k} P\left(\mathcal{X}(\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_k) - \mathbb{E}[\mathcal{X}(\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_k]) \geq \frac{t}{2}\right) \\ &\leq k_0^{\sum_{i=1}^k d_i} \cdot 2\exp\left(-\frac{t^2}{2n\sigma^2}\right). \end{aligned}$$

□

Lemma 28. *Consider the sub-sampled gradient, Hessian and third-order tensor defined in Eq.* (5)*. The sampling conditions in Eqs.* (6)*,* (7) *and* (8) *are satisfied with probability* $1-\delta$*,* $\delta \in (0,1)$ *for the following choice of the size of the sample sets* $\mathcal{S}^g, \mathcal{S}^b$ *and* $\mathcal{S}^t$*:*

$$n_g = \tilde{\mathcal{O}}\left(\frac{L_f^2}{\kappa_g^2\epsilon^2}\right), n_b = \tilde{\mathcal{O}}\left(\frac{L_g^2}{\kappa_b^2\epsilon^{4/3}}\right), n_t = \tilde{\mathcal{O}}\left(\frac{L_b^2}{\kappa_t^2\epsilon^{2/3}}\right), \tag{84}$$

where $\tilde{\mathcal{O}}$ *hides poly-logarithmic factors and a polynomial dependency to* d*.*

Proof. The proof consists in using concentration inequalities to prove that there exists a sample size such that the sampled quantity is close enough to the expected value.

Gradient

Let $n_g := |\mathcal{S}^g|$ and note that by the triangle inequality as well as Lipschitz continuity of ∇f (Assumption 1) we have

$$\|\mathbf{g}(\mathbf{x})\| = \frac{1}{n_g}\sum_{i\in\mathcal{S}^g}\|\nabla f_i(\mathbf{x})\| \leq L_g := \sigma_g \tag{85}$$

We then apply Theorem 24 on the gradient vector and require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0,1]$.

$$P\left(\,\|\mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})\| > t\right) \leq d\exp\left(\frac{-t^2}{8\sigma_g^2/n_g}\right) \overset{!}{\leq} \delta \tag{86}$$

Taking the log on both side, we get

$$\frac{-t^2}{8\sigma_g^2/n_g} = \frac{-t^2\cdot n_g}{8\sigma_g^2} \leq \log\frac{\delta}{d}, \tag{87}$$

which implies

$$\begin{aligned} t^2\cdot n_g &\geq (8\sigma_g^2)\log\frac{d}{\delta} \\ \implies n_g &\geq \frac{8\sigma_g^2}{t^2}\log\frac{d}{\delta}. \end{aligned} \tag{88}$$

Finally, we can simply choose $t = \kappa_g\epsilon$ in order to satisfy Eq. (6).

Hessian

Again, let $n_b := |\mathcal{S}^b|$ and note that by the triangle inequality as well as Lipschitz continuity of $\nabla^2 f$ (Assumption 1) we have

$$\|\mathbf{B}(\mathbf{x})\| \leq \frac{1}{n_b} \sum_{i \in \mathcal{S}^b} \|\nabla^2 f_i(\mathbf{x})\| \leq L_g := \sigma_b$$

Now we apply Theorem 24 on the Hessian and require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0, 1]$.

$$P\left(\left\|\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x})\right\| > t\right) \leq d \exp\left(\frac{-t^2}{8\sigma_b^2/n_b}\right) \overset{!}{\leq} \delta \tag{89}$$

Taking the log on both side, we get

$$\frac{-t^2}{8\sigma_b^2/n_b} = \frac{-t^2 \cdot n_b}{8\sigma_b^2} \leq \log \frac{\delta}{d}, \tag{90}$$

which implies

$$\begin{aligned} t^2 \cdot n_b &\geq (8\sigma_b^2) \log \frac{d}{\delta} \\ \implies n_b &\geq \frac{8\sigma_b^2}{t^2} \log \frac{d}{\delta}. \end{aligned} \tag{91}$$

Finally, we can simply choose $t = \kappa_b \epsilon^{2/3}$ in order to satisfy Eq. (7) since $\forall \mathbf{s} \in \mathbb{R}^d$,

$$\|(\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x}))\mathbf{s}\| \leq \|(\mathbf{B}(\mathbf{x}) - \nabla^2 f(\mathbf{x}))\| \cdot \|\mathbf{s}\| \leq \kappa_b \epsilon^{2/3} \|\mathbf{s}\|, \tag{92}$$

Third-order derivative

We apply Theorem 27 on the normalized [4] third-order derivative. Let $n_t := |\mathcal{S}^t|$ and define

$$\mathcal{Z} = \frac{1}{n_t} \sum_{i \in \mathcal{S}^t} \nabla^3 f_i(\mathbf{x}) - \nabla^3 f(\mathbf{x}) = \mathbf{T}(\mathbf{x}) - \nabla^3 f(\mathbf{x}). \tag{93}$$

We then require the probability of a deviation larger or equal to t to be lower than some $\delta \in (0, 1]$.

$$P\left(\left\|\mathbf{T}(\mathbf{x}) - \nabla^3 f(\mathbf{x})\right\| > t\right) \leq k_0^{3d} \cdot 2 \exp\left(-\frac{t^2 n_t}{2\sigma_t^2}\right) \overset{!}{\leq} \delta \tag{94}$$

Taking the log on both side, we get

$$-\frac{t^2 n_t}{2\sigma_t^2} \leq \log \frac{\delta}{2k_0^{3d}}, \tag{95}$$

which implies

$$n_t \geq \frac{2\sigma_t^2}{t^2} \log \frac{2k_0^{3d}}{\delta}. \tag{96}$$

[4] Note that we here consider the normalized sum. The reader can verify that the bound of Theorem 27 becomes $P(\|\mathcal{X} - \mathbb{E}\mathcal{X}\| \geq t) \leq k_0^{(\sum_{i=1}^k d_i)} \cdot 2 \exp\left(-\frac{t^2 n}{2\sigma^2}\right)$.

Finally, we can simply choose $t = \kappa_t \epsilon^{1/3}$ in order to satisfy Eq. (8) since $\forall \mathbf{s} \in \mathbb{R}^d$,

$$\begin{aligned}\|\mathbf{T}[\mathbf{s}]^2 - \nabla^3 f(\mathbf{x})[\mathbf{s}]^2\| &\leq \|\mathbf{T}[s] - \nabla^3 f(\mathbf{x})[\mathbf{s}]\|\|\mathbf{s}\| \\ &\leq \|\mathbf{T} - \nabla^3 f(\mathbf{x})\|\|\mathbf{s}\|^2 \\ &\leq \kappa_t \epsilon^{1/3}\|\mathbf{s}\|^2. \end{aligned} \tag{97}$$

□

C Total worst-case complexity

C.1 First-order guarantees

In this section, we analyze the total worst-case complexity of Algorithm 1 to obtain an ϵ-first-order critical point.

Theorem 29 (Theorem 4.8 restated)**.** *Let f_{low} be a lower bound on f. Denote by $\mathcal{K}_{outer}(\epsilon)$ the number of outer iterations defined in Eq.* (21) *and by $\mathcal{K}(\epsilon)$ the complexity of the model subsolver, both specialized to the case of first-order criticality. Assume Condition 1 holds. Then, given $\epsilon > 0$, Algorithm 1 needs at most*

$$\mathcal{K}_{outer}(\epsilon) \cdot (n_g + n_b \mathcal{K}(\epsilon) + n_t \mathcal{K}(\epsilon))$$

(stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$.

Proof. According to Theorem 6, the total number of outer iterations to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$ is $\mathcal{K}_{outer}(\epsilon)$ which is equal to

$$\left\lceil \left(\frac{8\sqrt[3]{2}\kappa_s^{4/3}(f(\mathbf{x}_0) - f_{low})}{\eta_1 \sigma_{min}} \epsilon^{-4/3} \right) \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2} \right) + \frac{1}{\log \gamma_2} \log \left(\frac{\sigma_{max}}{\sigma_0} \right) \right\rceil. \tag{98}$$

We denote the complexity of the model subsolver by $\mathcal{K}(\epsilon)$. Since the model sub-solver is a gradient-based approach,it only requires computing the gradient at $\mathbf{x}_k$ once, but it needs to recompute the Hessian-vector products and Tensor-vector products at each sub-iteration. Therefore, the gradient complexity is

$$\mathcal{K}_g(\epsilon) \leq n_g \cdot \mathcal{K}_{outer}(\epsilon). \tag{99}$$

The complexity of the Hessian-vector products is

$$\mathcal{K}_b(\epsilon) \leq n_b \cdot \mathcal{K}_{outer}(\epsilon) \cdot \mathcal{K}(\epsilon) \tag{100}$$

while the complexity of the Tensor-vector products is

$$\mathcal{K}_t(\epsilon) \leq n_t \cdot \mathcal{K}_{outer}(\epsilon) \cdot \mathcal{K}(\epsilon) \tag{101}$$

We conclude by combining Eqs. (99), (100) and (101).

□

Corollary 30 (Corollary 4.9 restated)**.** *Under the same assumptions as in Theorem 9, Algorithm 1 with the non-convex AGD variant presented in [13] as subsolver needs at most $\tilde{\mathcal{O}}(N\epsilon^{-3})$ (stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$ and $\epsilon \ll \frac{1}{N}$.*

Proof. In the first-order case, the termination criterion in Condition 2 is

$$\|\nabla m_k(\mathbf{s}_k)\| \leq \theta \|\mathbf{s}_k\|^3.$$

Employing the non-convex AGD variant presented in [13] as subproblem solver, we reach $\|\nabla m_k(\mathbf{s}_k)\| \leq \theta\|\mathbf{s}_k\|^3$ in $\mathcal{O}((\theta\|\mathbf{s}_k\|^3)^{-5/3})$ iterations. Now, given the lower bound on the stepnorm developed in Lemma 3 ($\|\mathbf{s}_k\| \geq \mathcal{O}(\epsilon^{1/3})$), we get that the required complexity for solving the subproblem is $\mathcal{K}(\epsilon) = \mathcal{O}(\epsilon^{-5/3})$.

By Theorem 9 and Lemma 8, we therefore get that Algorithm 1 needs at most

$$\begin{aligned}\mathcal{K}_{total}(\epsilon) &= \mathcal{K}_{outer}(\epsilon) \cdot (n_g + n_b\mathcal{K}(\epsilon) + n_t\mathcal{K}(\epsilon)) \\ &= \mathcal{K}_{outer}(\epsilon) \cdot \left(\tilde{\mathcal{O}}\left(\kappa_g^2\epsilon^2 \left/ L_f^2 + 1/N\right.\right)^{-1} + \tilde{\mathcal{O}}\left(\kappa_b^2\epsilon^{4/3} \left/ L_g^2 + 1/N\right.\right)^{-1}\mathcal{K}(\epsilon)\right. \\ &\quad \left. + \tilde{\mathcal{O}}\left(\kappa_t^2\epsilon^{2/3} \left/ L_b^2 + 1/N\right.\right)^{-1}\mathcal{K}(\epsilon)\right)\end{aligned}$$

oracle calls to reach an iterate $\mathbf{x}^*$ such that $\|\nabla f(\mathbf{x}^*)\| \leq \epsilon$.

Since we assumed $\epsilon \ll \frac{1}{N}$, the term $1/N$ dominates in each $\tilde{\mathcal{O}}$ in the equation above and thus:

$$\begin{aligned}\mathcal{K}_{total}(\epsilon) &= \mathcal{K}_{outer}(\epsilon) \cdot \left(\tilde{\mathcal{O}}(N) + \tilde{\mathcal{O}}(N)\mathcal{K}(\epsilon) + \tilde{\mathcal{O}}(N)\mathcal{K}(\epsilon)\right) \\ &= \mathcal{K}_{outer}(\epsilon) \cdot \left(\tilde{\mathcal{O}}(N) + 2\tilde{\mathcal{O}}(N\epsilon^{-5/3})\right).\end{aligned}$$

Finally, we obtain the total complexity by using the result established in Theorem 6 stating that $\mathcal{K}_{outer}(\epsilon_1) = \mathcal{O}(\epsilon_1^{-4/3})$.

□

C.2 Second-order guarantees

We now analyze the total worst-case complexity of Algorithm 1 to obtain an approximate second-order critical point $\mathbf{x}^*$. To compute each update step, we solve the current model m_k with the algorithm presented in [15] which is an accelerated gradient method for non-convex optimization that requires $\tilde{\mathcal{O}}(\epsilon_2^{-7/4})$ to find a point $\mathbf{x}^*$ such that $\chi_{m,2}(\mathbf{x}^*) \leq \sqrt{\epsilon_2}$.

Corollary 31 (Corollary 4.10 restated)**.** *Under the same assumptions as in Theorem 9, Algorithm 1 with the non-convex AGD variant presented in [15] as subsolver needs at most $\tilde{\mathcal{O}}(N\epsilon^{-15/4})$ (stochastic) oracle calls to reach an iterate $\mathbf{x}^*$ such that $\chi_{f,2}(\mathbf{x}^*) \leq \epsilon_2$ and $\epsilon_2 \ll \frac{1}{N}$.*

Proof. We start from the statement of Theorem 9 where $\mathcal{K}_{outer}(\epsilon_2)$ and $\mathcal{K}(\epsilon_2)$ are now specialized to the case of second-order criticality. Then, given $\epsilon_2 > 0$, Algorithm 1 needs at most

$$\begin{aligned}\mathcal{K}_{total}(\epsilon_2) = \mathcal{K}_{outer}(\epsilon_2) \cdot \Big(&\tilde{\mathcal{O}}\left(\frac{\kappa_g^2\epsilon_2^2}{L_f^2} + \frac{1}{N}\right)^{-1} + \tilde{\mathcal{O}}\left(\frac{\kappa_b^2\epsilon_2^{4/3}}{L_g^2} + \frac{1}{N}\right)^{-1}\mathcal{K}(\epsilon_2) \\ &+ \tilde{\mathcal{O}}\left(\frac{\kappa_t^2\epsilon_2^{2/3}}{L_b^2} + \frac{1}{N}\right)^{-1}\mathcal{K}(\epsilon_2)\Big)\end{aligned}$$

oracle calls to reach an iterate $\mathbf{x}^*$ such that $\chi_{f,2}(\mathbf{x}^*) \leq \epsilon_2$.

Since we assumed $\epsilon \ll \frac{1}{N}$, the term $1/N$ dominates in each $\tilde{\mathcal{O}}$ in the equation above and thus:

$$\mathcal{K}_{total}(\epsilon_2) = \mathcal{K}_{outer}(\epsilon_2) \cdot \left(\tilde{\mathcal{O}}(N) + \tilde{\mathcal{O}}(N)\mathcal{K}(\epsilon_2) + \tilde{\mathcal{O}}(N)\mathcal{K}(\epsilon_2)\right) \tag{102}$$

In the second-order case, the termination criterion in Condition 2 is

$$\chi_{m,2}(\mathbf{x}_k, \mathbf{s}_k) \leq \theta \|\mathbf{s}_k\|^2. \tag{103}$$

Employing the non-convex AGD variant presented in [15] as subproblem solver, we reach $\|\chi_{m,2}(\mathbf{x}_k, \mathbf{s}_k) m_k(\mathbf{s}_k)\| \leq \theta \|\mathbf{s}_k\|^2$ in $\mathcal{O}((\theta\|\mathbf{s}_k\|^2)^{-7/4})$ iterations. Now, given the lower bound on the step norm developed in Lemma 4 ($\|\mathbf{s}_k\| \geq \mathcal{O}(\epsilon_2^{1/2})$), we get that the required complexity for solving the subproblem is $\mathcal{K}(\epsilon_2) = \mathcal{O}(\epsilon_2^{-7/4})$. Plugging this in Eq. (102), we get

$$\mathcal{K}_{outer}(\epsilon_2) \cdot \left[\tilde{\mathcal{O}}(N) + 2\tilde{\mathcal{O}}(N\epsilon_2^{7/4})\right]. \tag{104}$$

Finally, we obtain the total complexity by using the result established in Theorem 6 stating that $\mathcal{K}_{outer}(\epsilon_2) = \mathcal{O}(\epsilon_2^{-2})$.

□

D Additional Lemmas

D.1 Lipschitz bounds

Lemma 32. *If f has an L-Lipschitz-continuous third-order derivatives, then $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$*

$$\begin{aligned} &\|\nabla^2 f(\mathbf{y}) - \nabla^2 f(\mathbf{x}) - \nabla^3 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\| \leq \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 \\ &\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) - \nabla^3 f(\mathbf{x})[\mathbf{y}-\mathbf{x}]^2\| \leq \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^3 \end{aligned} \tag{105}$$

Proof.

$$\begin{aligned} \nabla^2 f(\mathbf{y}) &= \nabla^2 f(\mathbf{x}) + \int_0^1 \nabla^3 f(\mathbf{x} + \tau(\mathbf{y}-\mathbf{x}))(\mathbf{y}-\mathbf{x}) d\tau \\ &= \nabla^2 f(\mathbf{x}) + \nabla^3 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) + \int_0^1 \left(\nabla^3 f(\mathbf{x} + \tau(\mathbf{y}-\mathbf{x})) - \nabla^3 f(\mathbf{x})\right)(\mathbf{y}-\mathbf{x}) d\tau \end{aligned}$$

Therefore,

$$\begin{aligned} &\|\nabla^2 f(\mathbf{y}) - \nabla^2 f(\mathbf{x}) - \nabla^3 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\| \\ &= \left\| \int_0^1 \left(\nabla^3 f(\mathbf{x} + \tau(\mathbf{y}-\mathbf{x})) - \nabla^3 f(\mathbf{x})\right)(\mathbf{y}-\mathbf{x}) d\tau \right\| \\ &\leq \int_0^1 \left\| \left(\nabla^3 f(\mathbf{x} + \tau(\mathbf{y}-\mathbf{x})) - \nabla^3 f(\mathbf{x})\right)(\mathbf{y}-\mathbf{x}) \right\| d\tau \\ &\leq \|\mathbf{y}-\mathbf{x}\| \int_0^1 \left\| \left(\nabla^3 f(\mathbf{x} + \tau(\mathbf{y}-\mathbf{x})) - \nabla^3 f(\mathbf{x})\right) \right\| d\tau \\ &\leq L\|\mathbf{y}-\mathbf{x}\|^2 \int_0^1 \tau d\tau = \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 \end{aligned} \tag{106}$$

For the second inequality,

$$\begin{aligned}
&\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) - \nabla^3 f(\mathbf{x})[\mathbf{y}-\mathbf{x}]^2\| \\
&= \left\| \int_0^1 \langle \nabla^2 f(\mathbf{x}+\tau(\mathbf{y}-\mathbf{x})) - \nabla^2 f(\mathbf{x}) - \nabla^3 f(\mathbf{x})[\mathbf{y}-\mathbf{x}]^2 \rangle d\tau \right\| \\
&\leq \|\mathbf{y}-\mathbf{x}\| \int_0^1 \left\| \nabla^2 f(\mathbf{x}+\tau(\mathbf{y}-\mathbf{x})) - \nabla^2 f(\mathbf{x}) - \nabla^3 f(\mathbf{x})(\mathbf{y}-\mathbf{x}) \right\| d\tau \\
&\leq \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^3, \qquad (107)
\end{aligned}$$

where the last inequality is simply due to the inequality proven first.

□