

---

# Cubic regularized subspace Newton for non-convex optimization

---

Jim Zhao<sup>1</sup>  
University of Basel  
jim.zhao@unibas.ch

Aurelien Lucchi  
University of Basel  
aurelien.lucchi@unibas.ch

Nikita Doikov  
EPFL  
nikita.doikov@epfl.ch

## Abstract

This paper addresses the optimization problem of minimizing non-convex continuous functions, a problem highly relevant in high-dimensional machine learning scenarios, particularly those involving over-parameterization. We analyze a randomized coordinate second-order method named SSCN, which can be interpreted as applying the cubic regularization of Newton’s method in random subspaces. This approach effectively reduces the computational complexity associated with utilizing second-order information, making it applicable in higher-dimensional scenarios. Theoretically, we establish strong global convergence guarantees for non-convex functions to a stationary point, with interpolating rates for arbitrary subspace sizes and allowing inexact curvature estimation, starting from an arbitrary initialization. When increasing the subspace size, our complexity matches the  $\mathcal{O}(\epsilon^{-3/2})$  rate of the full Newton’s method with cubic regularization. Additionally, we propose an adaptive sampling scheme ensuring the exact convergence rate of  $\mathcal{O}(\epsilon^{-3/2}, \epsilon^{-3})$  to a second-order stationary point, without requiring to sample all coordinates. Experimental results demonstrate substantial speed-ups achieved by SSCN compared to conventional first-order methods and other second-order subspace methods.

## 1 INTRODUCTION

In this paper, we address the problem of minimizing an objective function of the form  $\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ , where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a two-times differentiable, possibly non-convex function, having Lipschitz continuous gradient and Hessian. Our focus lies on scenarios where the dimension  $n$  has the potential to be significantly large, a context that holds relevance for numerous machine learning applications. This pertains particularly to situations in which models tend to exhibit a large number of parameters, commonly referred to as over-parametrization. One method of choice to optimize the objective functions associated with such models is to use (randomized) coordinate descent (CD) methods [Nesterov, 2012] or more generally subspace descent methods [Kozak et al., 2019]. The latter class of methods relies on iterative updates of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{S}_k^\top \mathbf{S}_k \nabla f(\mathbf{x}_k), \quad (1)$$

where  $\eta_k > 0$  is a step-size and  $\mathbf{S}_k \in \mathbb{R}^{\tau \times n}$  is a thin matrix where  $\tau \ll n$  is a subspace size. This update corresponds to moving in the (negative) direction of the gradient in the subspace spanned by the columns of  $\mathbf{S}_k$ . While various rules exist to choose the matrix  $\mathbf{S}_k$  [Hanzely et al., 2020], we will here focus on the case where  $\mathbf{S}_k$  is randomly sampled according to an arbitrary but fixed distribution of coordinate subsets  $\mathbf{S}_k$  in  $\mathbb{R}^n$ . We choose to focus on coordinate subsets because, in this scenario, there is no additional cost for applying a projection onto the subspace.

While first-order methods such as Eq. (1) are simple and relatively well-studied, their convergence is notably slow. In contrast, second-order optimization methods excel in terms of convergence speed as they possess the capability to capture more information about the optimization landscape. By incorporating such information about the curvature of the objective function, second-order methods such as Newton’s method with cubic regularization [Nesterov and Polyak, 2006] can navigate complex landscapes with greater efficiency. This often results in faster convergence rates (in terms of iterations) and enhanced accuracy in finding optimal

---

<sup>1</sup>Reverse alphabetical ordering, all authors contributed equally.

solutions. While first-order methods such as gradient descent are computationally cheaper per iteration, second-order methods can offer significant advantages in scenarios where function landscapes are intricate or when the convergence speed is crucial. However, their per-iteration cost makes them expensive in high-dimensional spaces.

Based on this observation, we study efficient subspace second-order methods, of the following form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{S}_k^\top [\mathbf{H}_k + \alpha_k \mathbf{I}]^{-1} \mathbf{S}_k \nabla f(\mathbf{x}_k), \quad (2)$$

where  $\mathbf{H}_k \in \mathbb{R}^{\tau \times \tau}$  is a fixed curvature matrix,  $\mathbf{I} \in \mathbb{R}^{\tau \times \tau}$  is the identity matrix, and  $\alpha_k > 0$  is a regularization constant. Note that in (2) we need to invert only the matrix of size  $\tau \times \tau$ , which is computationally cheap for small  $\tau \ll n$ . Clearly, substituting  $\mathbf{H}_k := \mathbf{0}$  into (2), we recover the pure CD method (1). However, to capture the second-order information about the objective function, we make the following natural choice

$$\mathbf{H}_k = \mathbf{S}_k \nabla^2 f(\mathbf{x}_k) \mathbf{S}_k^\top, \quad (3)$$

where  $\nabla^2 f(\mathbf{x}_k) \in \mathbb{R}^{n \times n}$  is the Hessian matrix. Therefore, for a specific coordinate subset, we need to use the second-order information only along the chosen coordinates, making our method scalable. It remains only to choose a parameter  $\alpha_k > 0$  in (2), which we select by using the cubic regularization technique Nesterov and Polyak [2006]. We use the name *stochastic subspace cubic Newton* (SSCN) for this algorithm, as it was introduced recently by Hanzely et al. [2020].

We see in our experiments (Fig. 1), that the combination of second-order information in a coordinate method, as in SSCN, indeed leads to a significant improvement in performance, as compared to the first-order CD. However, while Hanzely et al. [2020] focus on convex objective functions, it has been challenging to show a strict benefit from employing second-order information in coordinate descent methods in the *non-convex case*, which is important for modern applications in machine learning. In this work, we demonstrate that one can obtain strong theoretical convergence guarantees for general, possibly non-convex functions. Toward this goal, we make the following contributions:

- **Convergence guarantees:** we prove that for non-convex functions, SSCN converges to a stationary point starting from an arbitrary initialization (global convergence), at least with the rate of first-order CD when using an *approximation* of the curvature matrix, and with *strictly better rates* when using the true Hessian (3). The method converges for an arbitrary selection of  $\tau$ , and the rate becomes better for larger  $\tau$ , matching the convergence of the full Cubic Newton for  $\tau = n$ .

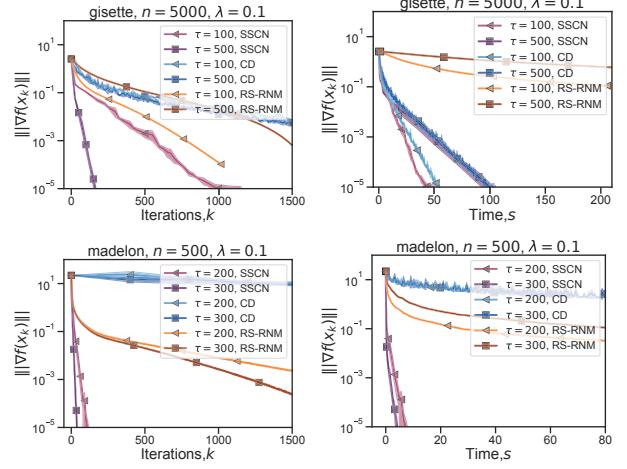


Figure 1: Comparison of CD, SSCN and RS-RNM [Fuji et al., 2022] for different constant coordinate schedules, where  $\tau$  denotes the dimension of the subspace in the SSCN method. Performance is measured w.r.t. iterations (first column) and time (second column) averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for the datasets *gisette* (first row) and *madelon* (second row). Experiment details are described in Section 5 and additional plots for the *duke* dataset can be found in Figure 6 in Appendix E.

- **Sampling schemes:** along with the constant sampling (fixed  $\tau$ ), we propose an *adaptive* theoretical sampling scheme ( $\tau_k$  grows with  $k$ ) that allows us to prove stronger convergence to a *second-order stationary* point for SSCN at the full Cubic Newton rate. Thus, this sampling scheme avoids being stuck at saddle points. In practice, we demonstrate that adaptive sampling does not require to fully sample all coordinates, while preserving the fast rate of convergence for the algorithm.
- **Experimentally:** Finally, we provide experimental results that verify our theoretical analysis and demonstrate significant speed-ups compared to the classical coordinate descent.

## 2 RELATED WORK

### Newton's method and cubic regularization (CR).

Newton's method is the classical optimization algorithm that employs second-order information (the Hessian matrix) about the objective function [Jorge and Stephen, 2006, Nesterov, 2018]. Even though the traditional Newton's method with a unit step size is effective at handling ill-conditioned problems, it might

Table 1: Comparison to previous work for non-convex optimization.  $\tau_k := \tau(S_k)$  refers to the adaptive sampling scheme of the main result in Theorem 4.6. And  $p'$  is defined as  $p' = \frac{\tau^2}{n^2}$ .

Method	# Coord.	Convergence rate to $\ \nabla f(\mathbf{x})\  \leq \epsilon$	2nd order stationary pt?	Iter. cost (exact step)
Gradient Descent [Nesterov, 2013]	$n$	$\mathcal{O}(k^{-1/2})$	✗	$\mathcal{O}(n)$
Cubic Newton (CR) [Nesterov and Polyak, 2006, Cartis et al., 2011]	$n$	$\mathcal{O}(k^{-2/3})$	✓	$\mathcal{O}(n^3)$
Stochastic CR [Tripuraneni et al., 2018, Kohler and Lucchi, 2017]	$n$	$\mathcal{O}(k^{-2/3})$	✓	$\mathcal{O}(n^3)$
Coordinate Descent [Richtárik and Takáč, 2014]	$\tau \leq n$	$\mathcal{O}(k^{-1/2})$	✗	$\mathcal{O}(\tau)$
Random subspace regularized Newton [Fuji et al., 2022]	$\tau \leq n$	$\mathcal{O}(k^{-1/2})$	✗	$\mathcal{O}(\tau^3)$
<b>This work:</b> Theorem 4.5	$\tau \leq n$	$\mathcal{O}(k^{-2/3} + (n(1-p'))^{1/4}k^{-1/2})$	✗	$\mathcal{O}(\tau^3)$
<b>This work:</b> Theorem 4.6	$\tau_k \leq n$	$\mathcal{O}(k^{-2/3})$	✓	$\mathcal{O}(\tau_k^3)$

not achieve global convergence when starting from arbitrary initializations. Therefore, various regularization techniques have been proposed to improve the convergence properties of the Newton’s method and to ensure the global rates (see Polyak [2007] for a detailed historical overview).

Among these regularization techniques is the well-established cubic regularization of Newton’s method, which achieves global complexity bounds that are provably better than those of the gradient descent [Nesterov and Polyak, 2006]. It uses a cubic over-estimator of the objective function as a regularization technique for the computation of a step to minimize the objective function. One limitation of this approach lies in its dependence on calculating the exact minimum of the cubic model. In an alternative approach, Cartis et al. [2011] introduced the ARC method, which alleviates this demand by allowing for an approximation of the minimizer. Other methods have been proposed to reduce the computational complexity of CR, which we discuss below.

**Subsampled/stochastic Newton and CR.** For objective functions that have a finite-sum structure, a popular method is to use sub-sampling techniques to approximate the Hessian matrix, such as in Byrd et al. [2011]. These techniques have also been adapted to cubic regularization [Kohler and Lucchi, 2017, Tripuraneni et al., 2018] followed by various improvements such as more practical sampling conditions [Wang et al., 2018] or variance reduction [Zhou et al., 2020]. We emphasize that there is a notable distinction between our work and these prior works. The latter addresses scenarios involving the sampling of data points, whereas our focus lies in the sampling of coordinates. These approaches are therefore orthogonal and could be combined with each other.

**Coordinate descent.** Coordinate descent methods are particularly useful when dealing with high-dimensional optimization problems, as they allow for efficient optimization by only updating one coordinate at a time, which can be computationally less expensive than updating the entire vector. There is a wide

literature focusing on the case where  $\tau = 1$  with precise convergence rates derived for instance by Nesterov [2012], Richtárik and Takáč [2014]. A generalization of CD known as subspace descent [Kozak et al., 2019] projects the gradient onto a random subspace at each iteration. The rates of convergence of coordinate descent for non-convex objectives were studied by Patrascu and Necoara [2015].

**Subspace Newton.** The subspace idea discussed above has also been extended to Newton’s method by Gower et al. [2019] resulting in the update rule  $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{S}_k (\nabla_S^2 f(\mathbf{x}_k))^{-1} \nabla_S f(\mathbf{x}_k)$ , where the gradient and Hessian are computed over a subset of selected coordinates  $S$ . Finally, Hanzely et al. [2020] extends this idea to cubic regularization, deriving rates of convergence in the more generic case where the objective function  $f$  is convex. In contrast, we consider the case where the function  $f$  is not necessarily convex. Recently, Fuji et al. [2022] analyzed a randomized subspace version of a (differently regularized) Newton’s method discussed in Ueda and Yamashita [2010] and obtained convergence to a first-order critical point with the same iteration complexity as CD. An overview of previous works for non-convex optimization can be found in Table 1.

## 3 ALGORITHM

### 3.1 Notation and setting

Our goal is to optimize a bounded below function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . We denote  $f^* := \inf_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$ . By  $\|\cdot\|$  we denote the standard Euclidean norm for vectors and the spectral norm for matrices.

**Assumption 3.1.** We assume that  $f$  has Lipschitz continuous Hessians with constant  $L_2 \geq 0$ , i.e.  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , it holds  $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L_2 \|\mathbf{x} - \mathbf{y}\|$ .

Consequently, we have a global bound for second-order

Taylor approximation of  $f$ , for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ :

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{1}{2} \langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle| \leq \frac{L_2}{6} \|\mathbf{y} - \mathbf{x}\|^3. \quad (4)$$

For a given subset of coordinates  $S \subset [n] := \{1, \dots, n\}$  and any vector  $\mathbf{x} \in \mathbb{R}^n$  we denote by  $\mathbf{x}_{[S]} \in \mathbb{R}^n$  a vector with nonzero elements whose indices are in  $S$  and by  $\mathbf{A}_{[S]} \in \mathbb{R}^{n \times n}$  the matrix with nonzero elements whose both rows and columns are in  $S$

$$(\mathbf{x}_{[S]})_i \stackrel{\text{def}}{=} \begin{cases} x_i & \text{if } i \in S \\ 0 & \text{else,} \end{cases} \quad (\mathbf{A}_{[S]})_{ij} \stackrel{\text{def}}{=} \begin{cases} A_{ij} & \text{if } i, j \in S \\ 0 & \text{else.} \end{cases}$$

We also denote the cardinality of set  $S$  by  $\tau(S) := |S|$ . Furthermore, we denote by  $\mathbf{x}|_S \in \mathbb{R}^{\tau(S)}$  the vector which only contains the entries in  $\mathbf{x}$ , which are in  $S$ . Similarly  $\mathbf{A}|_S \in \mathbb{R}^{\tau(S) \times \tau(S)}$  contains only the entries  $\mathbf{A}_{ij}$  with  $i \in S$  and  $j \in S$ . Note that  $\mathbf{A}_{[S]}|_S \equiv \mathbf{A}|_S$ .

### 3.2 Stochastic subspace cubic newton

Inequality (4) motivates us to introduce the following cubic regularized model, for a given  $\mathbf{x} \in \mathbb{R}^n$ , symmetric matrix  $\mathbf{Q} = \mathbf{Q}^\top \in \mathbb{R}^{n \times n}$ , coordinate subset  $S \subset [n]$ , and regularization parameter  $M > 0$ ,

$$\begin{aligned} \bar{m}_{\mathbf{x}, \mathbf{Q}, S, M}(\mathbf{h}) &:= \\ f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{h}_{[S]} \rangle + \frac{1}{2} \langle \mathbf{Q} \mathbf{h}_{[S]}, \mathbf{h}_{[S]} \rangle + \frac{M}{6} \|\mathbf{h}_{[S]}\|^3 \\ &\equiv f(\mathbf{x}) + \langle \nabla f(\mathbf{x})|_S, \mathbf{h} \rangle + \frac{1}{2} \langle \mathbf{Q}|_S \mathbf{h}, \mathbf{h} \rangle + \frac{M}{6} \|\mathbf{h}|_S\|^3, \end{aligned}$$

where  $\mathbf{h} \in \mathbb{R}^n$ . For simplicity and when it is clear from the context, we can omit extra indices, denoting our model simply by  $\bar{m}(\mathbf{h}) = \bar{m}_{\mathbf{x}, \mathbf{Q}, S, M}(\mathbf{h}) : \mathbb{R}^n \rightarrow \mathbb{R}$ . Then, the next iterate of our method is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \arg \min_{\mathbf{h} \in \mathbb{R}^n} \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M}(\mathbf{h}), \quad k \geq 0, \quad (5)$$

where  $S_k \subset [n]$  are random subspaces of a fixed size  $\tau \equiv \tau(S_k)$ ,  $\tau \in [n]$ , so we update the  $\tau$  coordinates of  $\mathbf{x}_k$  which are in  $S_k$  in each iteration. Note that for  $\tau = n$  we obtain the full cubic Newton step [Nesterov and Polyak, 2006]. However, we are interested in choosing  $\tau \ll n$ , such that the corresponding optimization subproblem (5) can be solved efficiently when  $n$  is large.

*Remark 3.2.* We note that Eq. (5) is equivalent to the following update rule:  $\mathbf{x}_{k+1}|_{S_k} = \mathbf{x}_k|_{S_k} + \arg \min_{\mathbf{h} \in \mathbb{R}^\tau} m_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M}(\mathbf{h})$  with a model  $m : \mathbb{R}^\tau \rightarrow \mathbb{R}$  defined as

$$\begin{aligned} m_{\mathbf{x}, \mathbf{Q}, S, M}(\mathbf{h}) &:= \\ f(\mathbf{x}) + \langle \nabla f(\mathbf{x})|_S, \mathbf{h} \rangle + \frac{1}{2} \langle \mathbf{Q}|_S \mathbf{h}, \mathbf{h} \rangle + \frac{M}{6} \|\mathbf{h}\|^3, \quad (6) \end{aligned}$$

where  $\mathbf{h} \in \mathbb{R}^\tau$ . This update rule implies that in practice we only need to solve a cubic subproblem of dimension  $\tau \ll n$ . At the same time, it is more convenient to work with the initial  $\bar{m}$  in the theoretical analysis.

---

#### Algorithm 1 SSCN: Stochastic Subspace Cubic Newton

---

```

1: Initialization:  $\mathbf{x}_0 \in \mathbb{R}^n$ , distribution  $\mathcal{D}$  of random subsets  $S \subset [n]$  of size  $\tau \equiv \tau(S)$ 
2: for  $k = 0, 1, \dots$  do
3:   Sample  $S_k$  from distribution  $\mathcal{D}$ 
4:   Estimate matrix  $\mathbf{Q}_k \approx \nabla^2 f(\mathbf{x}_k)|_{S_k}$ 
5:   Set  $\mathbf{x}_{k+1}|_{S_k} = \mathbf{x}_k|_{S_k} + \arg \min_{\mathbf{h} \in \mathbb{R}^\tau} m_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h})$ ,
     for some  $M_k > 0$ 
6: end for

```

---

The resulting optimization method is stated in Algorithm 1, where we first sample  $S_k$  from a chosen distribution  $\mathcal{D}$ , and then perform an update by minimizing our model. Additional details concerning this minimization subproblem will be provided shortly.

In the remaining part of this work, we will assume  $\mathcal{D}$  to be the uniform distribution. In Algorithm 1, we have the freedom of choosing the matrix  $\mathbf{Q}_k$  in every iteration. Even though we mainly focus on employing the true Hessian  $\nabla^2 f(\mathbf{x}_k)$  for a selected subset of coordinates, there are several interesting possibilities, that can be also considered within our framework.

- *Full Hessian matrix:*  $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_k)|_{S_k}$ . Then, our algorithm recovers the SSCN method from Hanzely et al. [2020]. This is the most powerful version, which we equip with new strong convergence guarantees, valid for general non-convex objective functions.
- *No second-order information:*  $\mathbf{Q}_k = \mathbf{0}$ . In this case, Algorithm 1 and our analysis recovers the rate of the coordinate descent (CD) method, even though our algorithm is slightly different due to the cubic regularization, which affects the step-size selection: one step becomes  $\mathbf{h}_k^* = -\eta_k \nabla f(\mathbf{x}_k)|_{S_k} \in \mathbb{R}^\tau$ , with  $\eta_k = \sqrt{2/(M_k \|\nabla f(\mathbf{x}_k)|_{S_k}\|)}$ . The ability to tackle this extreme case demonstrates the robustness of our iterations: we show that Algorithm 1 works even if the approximation  $\mathbf{Q}_k \approx \nabla^2 f(\mathbf{x}_k)|_{S_k}$  is not exact.
- *Lazy Hessian updates:*  $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_t)|_{S_k}$ , where  $\mathbf{x}_t$  is some point from the past,  $0 \leq t \leq k$ . In this case, we use the same Hessian for several steps, which improves the arithmetic complexity of the method [Doikov et al., 2023].
- *Quasi-Newton updates*, such as DFP, BFGS, and L-BFGS [Dennis and Moré, 1977, Nocedal and

Wright, 1999]. Some recent works combine quasi-Newton methods with the cubic regularization technique, including Kamzolov et al. [2023], Scieur [2024].

- *Finite-difference approximation.* When second-order information is not directly available, we can approximate the Hessian using only the gradients, as follows:  $(\mathbf{Q}_k)_{ij} = \frac{1}{\delta}(\nabla f(\mathbf{x}_k + \delta \mathbf{e}_i) - \nabla f(\mathbf{x}_k))_j$ , for  $i, j \in S_k$ , with a possible symmetrization later on. Here,  $\mathbf{e}_i$  is the basis vector, and  $\delta > 0$  is a parameter. Choosing  $\delta$  sufficiently small we ensure the rate of the cubic Newton [Cartis et al., 2012, Grapiglia et al., 2022, Doikov and Grapiglia, 2023].

**Complexity of solving the cubic subproblem.** Note that for non-convex functions the model  $\bar{m}(\mathbf{h})$  is in general non-convex. However, its *global minimum* is always well-defined and can be found by standard techniques from linear algebra. One step of our method can be rewritten in the standard form, as in (2):  $(\mathbf{x}_{k+1} - \mathbf{x}_k)|_{S_k} = -(\mathbf{Q}_k|_{S_k} + \alpha_k \mathbf{I})^{-1} \nabla f(\mathbf{x}_k)|_{S_k}$ , and the regularization constant  $\alpha_k$  can be found as a solution to the following *univariate concave maximization*,

$$\max_{\alpha > 0} \left[ -\frac{1}{2} ((\mathbf{Q}_k|_{S_k} + \alpha \mathbf{I})^{-1} \nabla f(\mathbf{x}_k)|_{S_k}, \nabla f(\mathbf{x}_k)|_{S_k}) - \frac{2^4 \alpha^3}{3 M_k^2} : \mathbf{Q}_k|_{S_k} + \alpha \mathbf{I} \succ \mathbf{0} \right]. \quad (7)$$

It can be done efficiently by means of any one-dimensional procedure (e.g. the binary search or univariate Newton's method, see also Chapter 7 of Conn et al. [2000] and Section 5 of Nesterov and Polyak [2006]). Then, the complexity of one step is  $\mathcal{O}(\tau^3)$ , as for the standard matrix inversion. Cartis et al. [2011] show that one can retain the fast rate with an *inexact model minimizer* which solves  $m(\mathbf{h})$  on a Krylov subspace. The subproblem can also be solved using gradient descent, as shown by Carmon and Duchi [2019], which means that only Hessian-vector products are required.

## 4 CONVERGENCE ANALYSIS

### 4.1 General convergence rate

First, let us establish a general convergence result for Algorithm 1 when using an arbitrary symmetric matrix  $\mathbf{Q}_k = \mathbf{Q}_k^\top \in \mathbb{R}^{n \times n}$ . To quantify the approximation error, we introduce parameter  $\sigma \geq 0$  such that

$$\|\nabla^2 f(\mathbf{x}_k)_{[S_k]} - \mathbf{Q}_k\| \leq \sigma, \quad \forall k. \quad (8)$$

Note that  $\sigma$  bounds the distance to the already subsampled Hessian  $\nabla^2 f(\mathbf{x}_k)_{[S_k]}$ , **not** to the full Hessian  $\nabla^2 f(\mathbf{x}_k)$ . Thus, in the simplest case one could choose

$\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_k)_{[S_k]}$  (and we have  $\sigma = 0$ ), but our approach also accommodates inaccuracies in the Hessian estimation. For instance, if we use lazy Hessian updates ( $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_t)_{[S_k]}$ , where  $\mathbf{x}_t$  is some point from the past,  $0 \leq t \leq k$ ), we can choose  $\sigma := L_2 \|\mathbf{x}_k - \mathbf{x}_t\|_2$ , where  $L_2$  is the Lipschitz constant of the Hessian (Assumption 3.1). Similar bounds can be derived if we use finite-difference approximation of the second derivatives. Then,  $\sigma$  corresponds to the approximation error. On the other extreme, when  $\mathbf{Q}_k = \mathbf{0}$  our method should recover the standard rate of the coordinate descent (CD). For that, we have to use an additional assumption.

**Assumption 4.1.** Let  $f$  have Lipschitz continuous gradient with constant  $L_1 > 0$ , i.e.  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , it holds  $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L_1 \|\mathbf{x} - \mathbf{y}\|$ .

In this case the error  $\sigma$  is upper bounded by  $\sigma \leq L_1$ . By choosing the regularization parameter  $M_k$  in our cubic model sufficiently large, we can ensure that the following progress condition is satisfied, at each iteration  $k \geq 0$ :

$$f(\mathbf{x}_k + \mathbf{h}_k^*) \leq \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h}_k^*). \quad (9)$$

This inequality justifies that our method is monotone (i.e.  $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ ), and it is crucial for establishing that SSCN can achieve *any* precision in terms of the gradient norm for *any* size of a stochastic subspace, and for arbitrary selection of  $\mathbf{Q}_k$ . All missing proofs are provided in the appendix.

**Theorem 4.2.** Let the sequence  $\{\mathbf{x}_i\}$  be generated by Algorithm 1 with arbitrary  $\mathbf{Q}_k$  satisfying (8), and any fixed  $\tau \equiv \tau(S_k) \in [n]$ . Let the regularization parameter at iteration  $k \geq 0$  be chosen as

$$M_k = 2L_2 + \frac{7^2(\sigma + L_1)^2}{2\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2}. \quad (10)$$

For a given accuracy level  $\varepsilon > 0$ , assume that  $\|\nabla f(\mathbf{x}_i)\| \geq \varepsilon$ , for all  $0 \leq i \leq K$ . Then, it holds

$$K \leq \frac{n}{\tau} \left[ \frac{(2 + \frac{7^2}{3})(\sigma + L_1)(f(\mathbf{x}_0) - f^*)}{\varepsilon^2} + \frac{4L_2(f(\mathbf{x}_0) - f^*)}{3(\sigma + L_1)\varepsilon} \right]. \quad (11)$$

### 4.2 The power of second-order information

In this and the following sections, we assume that we use the exact second-order information,  $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_k)_{[S_k]}$ . In this case, we are able to ensure a faster rate of convergence, thus showing the power of utilizing the second-order information, for general possibly non-convex problems.

We would like to highlight that the analysis we employ is fundamentally different than in Hanzely et al. [2020], which means that our work can not be seen as a straightforward extension but it instead requires

novel ideas to derive the proofs. Indeed, the non-convex analysis of the deterministic full-space cubic Newton method for non-convex objectives is based on the following progress for every iteration (see, e.g. the proof of Theorem 1 in Nesterov and Polyak [2006]),

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{1}{12\sqrt{L}} \|\nabla f(\mathbf{x}_{k+1})\|^{3/2},$$

where  $L$  is the Lipschitz constant of the Hessian. Note that contrary to the first-order gradient methods, we have the gradient norm at the *new point*  $\|\nabla f(\mathbf{x}_{k+1})\|$  in the right-hand side of the last inequality. Therefore, when analysing a stochastic subspace version of the cubic Newton, it becomes challenging to generalize this proof, since  $\mathbf{x}_{k+1}$  depends on the choice of the random subspace. In contrast, our new analysis is based on studying the global properties of the stochastic cubic model of the objective function and allows inexactness in the progress inequality (see sections B and C in the appendix for the full proofs). To the best of our knowledge, our analysis is novel.

Using the model's optimality (see Section A.1 in the appendix), we can establish that we consistently decrease the objective function at each step, with the following progress.

**Lemma 4.3.** *For any  $\mathbf{x}_k \in \mathbb{R}^n$  and arbitrary  $S_k \subset [n]$ , let  $\mathbf{h}_k^* = \arg \min_{\mathbf{h}} \bar{m}_{\mathbf{x}_k, \nabla^2 f(\mathbf{x}_k), S_k, M}(\mathbf{h})$ . Then we have, for any  $M \geq L_2$ ,*

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{M}{12} \|\mathbf{h}_k^*\|^3. \quad (12)$$

For our refined analysis, we must characterize the distance between the coordinate-sampled gradient  $\nabla f(\mathbf{x})_{[S]}$  and the full gradient  $\nabla f(\mathbf{x})$ , as well as the corresponding distance for the Hessian.

**Lemma 4.4.** *For any  $\mathbf{x} \in \mathbb{R}^n$  and any subset  $S \subset [n]$ , we have with probability at least  $1 - \delta$ :*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \leq \delta^{-1} \sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\| \quad (13)$$

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\| \leq \delta^{-1} \sqrt{1 - p'} \|\nabla^2 f(\mathbf{x})\|_F \quad (14)$$

where  $p' := (\frac{\tau(S)}{n})^2$  and  $\|\cdot\|_F$  is the Frobenius norm of a matrix.

As expected, Lemma 4.4 shows that increasing the sampling size  $\tau(S)$  yields more accurate sampled gradients and Hessians. This effect will later be verified experimentally in Section 5 where we will test different coordinate sampling schedules. Note that we can prove analogous bounds in expectation, as in Eqs. (13)-(14) (see Lemma C.1, C.2). We denote by  $\mathbb{E}[\cdot]$  the full expectation w.r.t. all randomness in the algorithm. We are ready to state our main convergence rate for a fixed coordinate sample size.

**Theorem 4.5.** *Let the sequence  $\{\mathbf{x}_i\}$  be generated by Algorithm 1 with  $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_k)_{[S_k]}$ , and any fixed  $\tau \equiv \tau(S_k) \in [n]$ . For a given  $\varepsilon > 0$ , assume that the first  $K$  gradients are such that  $\mathbb{E}\|\nabla f(\mathbf{x}_i)\| \geq \varepsilon, 0 \leq i \leq K$ . Then for a sufficiently large  $M \geq L_2$ , it holds*

$$K = \mathcal{O}\left(\left[\frac{n}{\tau}\right]^{3/2} \frac{\sqrt{L_2}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}} + n^{1/2}(1 - p')^{1/2} \left[\frac{n}{\tau}\right]^2 \frac{L_1(f(\mathbf{x}_0) - f^*)}{\varepsilon^2}\right). \quad (15)$$

We see that according to this result, SSCN achieves any desirable accuracy for the gradient norm after a finite number of iterations, starting from an arbitrary initialization  $\mathbf{x}_0$ . The size of the stochastic subspaces  $\tau = \tau(S_k)$  can be *arbitrary*. Note that  $p' = (\frac{\tau}{n})^2$  increases with increasing  $\tau$  and we recover the rate of the full Cubic Newton for  $\tau = n$ . If the cubic subproblem is solved exactly in each iteration (with the cost of  $\mathcal{O}(\tau^3)$ ), then the total computational complexity is  $\mathcal{O}\left((\tau n)^{3/2} \frac{\sqrt{L_2}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}} + (1 - p')^{1/2} \tau n^{5/2} \frac{L_1(f(\mathbf{x}_0) - f^*)}{\varepsilon^2}\right)$ . Note that the global computational complexity for CR is  $\mathcal{O}\left(n^3 \frac{\sqrt{L_2}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}}\right)$ . Therefore, we see that SSCN is already *strictly better* than CR for  $\tau(1 - p')^{1/2} < (\varepsilon n L_2)^{1/2}/L_1$ . Note that in practice we can already observe speed-ups of SSCN for smaller problem dimensions. We also see that the second term in (15) matches the complexity of CD, up to the factor  $n^{3/2}(1 - p')^{1/2}/\tau$ , which tends to 0 when  $\tau \rightarrow n$ . Lastly, we highlight that the final complexity of SSCN with constant sampling can be taken as the minimum of both (11) and (15), thus achieving the best of these two bounds.

### 4.3 Adaptive sampling scheme

In the following, we present a scheme to sample the number of coordinates at each iteration that yields even faster convergence to a *second-order stationary point* up to an arbitrary precision. This scheme is *adaptive* in the sense that it depends on the gradient and Hessian measured at each iteration. We measure optimality using the standard first and second-order criticality measures  $\|\nabla f(\mathbf{x})\|$  and  $\lambda_{\min}(\nabla^2 f(\mathbf{x}))$  (the minimum eigenvalue of the Hessian matrix). To do so, we introduce the following quantity [Nesterov and Polyak, 2006]:

$$\mu(\mathbf{x}) = \max \{\|\nabla f(\mathbf{x})\|^{3/2}, [-\lambda_{\min}(\nabla^2 f(\mathbf{x}))]^3\}.$$

Given  $\epsilon_1 > 0$  and  $\epsilon_2 > 0$ , we would like to have with probability at least  $1 - \delta$

$$\begin{aligned} \|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)_{[S_k]}\| &\leq \delta^{-1} \epsilon_1, \\ \|\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)_{[S_k]}\| &\leq \delta^{-1} \sqrt{\epsilon_2}. \end{aligned} \quad (16)$$

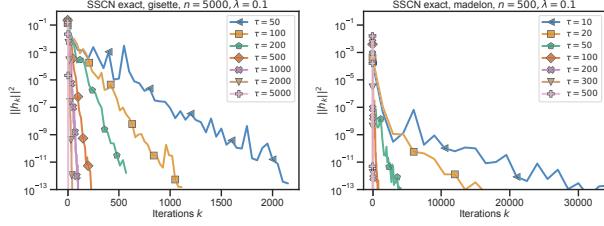


Figure 2: Squared norm of the step  $\mathbf{h}_k$  for different constant coordinate schedules for logistic regression with non-convex regularization with  $\lambda = 0.1$  for two different datasets (left: *gisette*, right: *madelon*). For the same plot for the *duke* dataset, see Figure 10. Note that the y-axis is plotted in log-scale.

Taking into account Lemma 4.4, to ensure (16), it is enough to choose  $\tau(S_k)$  such that

$$\sqrt{1 - \frac{\tau(S_k)}{n}} = \frac{\delta^{-1} \epsilon_1}{\|\nabla f(\mathbf{x}_k)\|} \quad \text{and} \quad \sqrt{1 - p'} = \frac{\delta^{-1} \sqrt{\epsilon_2}}{\|\nabla^2 f(\mathbf{x}_k)\|_F}.$$

Putting everything together, we obtain the following condition for our adaptive scheme at iteration  $k \geq 0$ :

$$\frac{\tau(S_k)}{n} \geq \max \left\{ 1 - \frac{\delta^{-2} \epsilon_1^2}{\|\nabla f(\mathbf{x}_k)\|^2}, \sqrt{1 - \frac{\delta^{-2} \epsilon_2}{\|\nabla^2 f(\mathbf{x}_k)\|_F^2}} \right\}. \quad (17)$$

Next, we will demonstrate that choosing  $\epsilon_1 = \epsilon_2 = c_{k-1} \cdot \|\mathbf{h}_{k-1}^*\|^2$ , for a given sequence  $(c_k)$ , allows us to recover the convergence rate of cubic regularization. Importantly, Theorem 4.6 permits the choice of an arbitrary sequence  $(c_k)$  enabling the adjustment of the number of coordinates as a function of the iteration  $k$ . Section 5 will illustrate that this flexibility leads to an effective strategy, resulting in substantial speed-ups.

**Theorem 4.6.** Consider the sequence  $\{\mathbf{x}_k\}_{k=0}^K$  generated by  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k^*$  where  $\tau$  satisfies Eq. (17) with  $\epsilon_1 = \epsilon_2 = c_{k-1} \cdot \|\mathbf{h}_{k-1}^*\|^2$  for some  $c_{k-1} > 0$ ,  $\forall k = 0, \dots, K$ . Let  $M \geq L_2$ . Let  $\Delta_0 = f(\mathbf{x}_0) - f^*$ , and define the following constants (dependent on  $M$ ):

$$C_M = \left( (2M+1)^{3/2} + (4\delta^{-3/4} + \sqrt{2}\delta^{-3/2}) \max_i c_i^{3/2} \right)^{-1},$$

$$D_M = \left( \frac{27M^3}{2} + 4\delta^{-3/2} \max_i c_i^{3/2} \right)^{-1}.$$

Then with probability at least  $1 - \delta$ , we have

$$\min_{1 \leq j \leq K} \mu(\mathbf{x}_j) \leq \frac{1}{K} \left( \frac{6}{M} \max(C_M^{-1}, D_M^{-1}) \Delta_0 + 4\delta^{-3/2} c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right),$$

where  $\mathbf{h}_{-1}^*$  is such that

$$\mathbb{E}_{[S]} \|\nabla^2 f(\mathbf{x}_0) - \nabla^2 f(\mathbf{x}_0)_{[S]}\| \leq \sqrt{c_{-1}} \|\mathbf{h}_{-1}^*\|.$$

Theorem 4.6 states that Algorithm 1 with an adaptive sampling scheme converges to an  $\epsilon$ -second-order stationary point at a rate of  $\mathcal{O}(\epsilon^{-3/2}, \epsilon^{-3})$ .

**A practical scaling rule.** The theoretical result of Theorem 4.6 relies on using Eq. (17) which requires access to the gradient and Hessian norms. We note that one can use the estimates computed over a subset of coordinates  $S \subset [n]$ . We give further details regarding the validity of this scheme in the Appendix. Alternatively, we have discovered that a less complex schedule produces comparable results. Our starting point is the observation that  $\|\mathbf{h}_{k-1}^*\|^2$  exhibits exponential growth on most datasets, as illustrated in Figure 2. We posit that this exponential increase allows us to employ a coordinate sampling schedule that also follows an exponential trend. By choosing a slowly growing constant for the exponential, we obtain significant speed-ups in practice. We substantiate the effectiveness of this straightforward approach in Appendix E.2.

Table 2: Overview of the datasets used in the experimental section with non-convex regularizer  $\lambda$ .

	Type	n	#samples	$\lambda$
<i>gisette</i>	Classification	5.000	6.000	0.1
<i>duke</i>	Classification	7.129	44	0.1
<i>madelon</i>	Classification	500	2.000	0.1
<i>realsim</i>	Classification	20.958	72.309	0.1

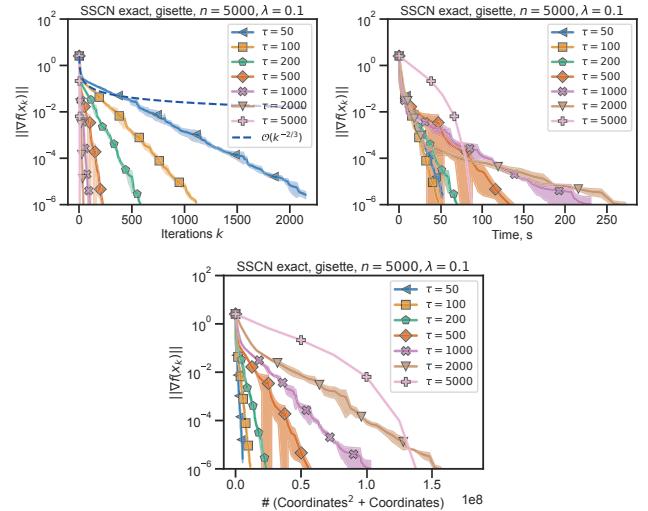


Figure 3: Convergence of different constant coordinate schedules measured w.r.t. iterations (first column), time (second column) and  $\#(\text{Coordinates}^2 + \text{Coordinates})$  evaluated (third column) averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for *gisette* dataset. Similar plots for the *duke* and *madelon* datasets can be found in Fig. 4.

## 5 EXPERIMENTS

We now verify our theoretical results numerically. Due to space limitations, we can only present a fraction of the experiments and refer the reader to the Appendix for the remaining experiments. We ran experiments with a logistic regression loss with a *non-convex regularizer*  $\lambda \cdot \sum_{i=1}^n \mathbf{x}_i^2 / (1 + \mathbf{x}_i^2)$  [Kohler and Lucchi, 2017] where  $\lambda > 0$  controls the strength of the regularizer. An overview of the three datasets used in our experiments can be found in Table 2. All runs were initialized in the origin at  $\mathbf{x}_0 = \mathbf{0}$  and all the plots shown in this section are averaged over three runs. The shaded region corresponds to one standard deviation. In the inner loop of Algorithm 1, we solve the subproblem exactly up to a pre-specified tolerance of  $1e^{-5}$ . We use the exact same subsolver, which is discussed in section 3.3 in Kohler and Lucchi [2017]<sup>1</sup>. The experiments were run on an Apple MacBook Pro with an Apple M1 Pro Chip and 16 GB RAM.

**Constant coordinate schedule.** In Figure 3 and 4 we plot the convergence of different constant coordinate schedules w.r.t. iterations, time and # (coordinates<sup>2</sup> + coordinates). We chose the last measure to approximate the efficiency of the coordinate schedule, since the Hessian matrix scales quadratically and the gradient scales linearly with the number of sampled coordinates. As expected, full cubic Newton is the fastest in terms of number of iterations and the fewer coordinates are sampled, the more iterations are required to reach the same gradient norm. However, in terms of average run time and number of evaluated coordinates, we observe that **smaller coordinate schedules are faster and more efficient up to some gradient norm** for the *gisette* and *duke* dataset, where the benefit is more pronounced if the problem is higher dimensional. We note that, even for the smaller *madelon* dataset, smaller coordinate schedules are more efficient w.r.t # coordinate evaluations until  $\|\nabla f(\mathbf{x}_k)\| \approx 1e^{-1}$ . This clearly underscores the potential for substantial computational savings through the utilization of a straightforward approach that samples a fixed number of coordinates. Next, we will compare our method to other first and second order subspace methods.

**Comparison to other methods.** We compare SSCN to the random subspace regularized Newton method (RS-RNM) proposed by [Fuji et al., 2022] for different constant coordinate schedules, which we believe is the fairest comparison to our method. Other inexact or stochastic Newton methods either study convex setups Gower et al. [2019], Jiang et al. [2024] or focus on set-

tings, where the stochasticity arises from subsampling data points Yao et al. [2018], Xu et al. [2020], while the dimensionality of the problem remains unchanged. In comparison, [Fuji et al., 2022] and our work both study subspace Newton methods in a smooth, nonconvex setting. We also compare SSCN to a *vanilla* randomized coordinate descent (CD), where the step size  $\eta$  was chosen via a line search procedure that guarantees that  $f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \geq \frac{\eta}{2} \|\nabla f(\mathbf{x}_k)\|_2^2$ , which is the so-called Armijo condition [Nocedal and Wright, 1999].

The results are shown in Figure 1 and 6. It can be seen that SSCN is much faster than RS-RNM on both datasets. We attribute it to two different aspects: 1.) SSCN is more efficient as it does not need to build the full Hessian matrix before projecting it onto a random subspace. Consequently, the speed-up of RS-RNM compared to full space RNM is only w.r.t. the matrix inversion, which is performed in a lower dimension. This is particularly apparent on the *gisette* dataset, which has 5000 coordinates. In contrast, SSCN only constructs the Hessian submatrix of size  $\tau \times \tau$ . This is why the runtime is much longer for RS-RNM, while the number of iterations for both methods is comparable. 2.) SSCN performs well on ill-conditioned problems, which is the case for the *madelon* dataset (which has a condition number of the order of  $1e7 \sim 1e8$ ), while RS-RNM seems to be affected by the condition number. For this reason, RS-RNM also needs more iterations to find a first-order stationary point.

For the *duke* dataset, we observe that CD converges extremely fast in few iterations. Since the per-iteration cost of CD is much lower compared to SSCN, this results in much faster convergence for CD. However, CD converges very slowly to what seems to be a suboptimal solution for the *madelon* dataset, while for the *gisette* dataset the fastest schedule is clearly SSCN which only samples 2% of the coordinates. This set of experiments highlights some interesting trade-offs between CD and SSCN. In the case of simpler objective functions, CD demonstrates a clear advantage. However, this advantage can diminish rapidly **when dealing with more intricate objective functions where SSCN tends to be more efficient**. We believe that these experiments highlight that our proposed method is a competitive subspace Newton method in practice.

## 6 LIMITATIONS

In terms of theory, we emphasize that the adaptive scheme studied in Section 4.3 is developed assuming we have access to the partial derivatives of the expected objective function. Thus, if the dataset contains a large number of data points, one might need to resort to a stochastic approximation, which would require adapting the analysis to work with high probability.

<sup>1</sup>We used the implementation provided at [https://github.com/jonaskohler/subsampled\\_cubic\\_regularization/](https://github.com/jonaskohler/subsampled_cubic_regularization/) under the Apache-2.0 license.

This could be an interesting venue for future work where one can use both sampling of coordinates and datapoints in the same algorithm.

Based on experimental observations (refer to, for example, Figure 1 and Figure 6), we have found that the comparative advantage of SSCN over first-order CD is heavily contingent upon the complexity of the objective function. When dealing with a well-conditioned loss function, the lower per-iteration cost of CD results in notably faster convergence in terms of wall clock time. However, in scenarios where the loss function is ill-conditioned, SSCN manages to converge while CD struggles to do so within a reasonable timeframe. Thus, there appears to be a significant interest in studying the intrinsic ill-conditioning aspect of contemporary machine learning models. This would allow us to better understand the applicability of coordinate methods for such models.

**Extension to arbitrary subspaces** While this work focuses on coordinate subspaces, we believe that our overall analysis should also work for arbitrary subspaces  $S_k$ . The key step is to have analogous inequalities to Lemma 4.4 when  $S_k$  is an arbitrary subspace. In this case, we just need to substitute  $\nabla f(x_k)|_{S_k} \in \mathbb{R}^\tau$  with the vector  $\nabla f_{S_k}(x_k) := S_k^\top \nabla f(x_k)$ , and replace  $\nabla^2 f(x_k)|_{S_k} \in \mathbb{R}^{\tau \times \tau}$  with the matrix  $\nabla^2 f(x_k)_{S_k} := S_k^\top \nabla^2 f(x_k) S_k$ , where the 'thin' matrix  $S_k \in \mathbb{R}^{n \times \tau}$  describes the subspace. In the case when  $S_k$  is a submatrix of the identity matrix, it corresponds to the coordinate subspace and our current method. Then, we can use the analogous inequality to Lemma 4.4 as the definition of 'effective subspace size'  $\tau$ , and our results will remain valid. However, we would like to emphasize that arbitrary subspaces are not that important from a practical perspective, and coordinate selection is typically the only viable choice in practice. This is why we focused mainly on this case in the paper. For all other choices, it is also not clear how the gradients and Hessian should be estimated effectively. This still remains an open question.

## 7 CONCLUSION

We analyzed the convergence rate of SSCN for the class of twice differentiable non-convex functions. From a theoretical perspective, to the best of our knowledge, we are the first to prove the global convergence guarantees of the subspace Newton method for a wide family of non-convex problems, ensuring that

1. for an arbitrary fixed  $\tau$  our global rates interpolate between the rate of coordinate descent and the cubic Newton rate and recover the cubic Newton rate for full coordinate sampling and

2. by using our novel adaptive sampling scheme for  $\tau$ , SSCN achieves rates that are provably better than previously known.

Furthermore, we have observed empirically that a more straightforward exponential schedule produces favorable results. Overall, our experiments demonstrated that one can sample a fraction of the coordinates while observing fast convergence. This results in significant computational gains compared to the vanilla cubic regularization algorithm. There are various interesting extensions to consider such as the use of importance sampling and combining coordinate sampling with datapoint sampling in the case of finite-sum objective functions.

## Acknowledgments

ND is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22.00133.

## References

- Artem Agafonov, Dmitry Kamzolov, Pavel Dvurechensky, Alexander Gasnikov, and Martin Takáč. Inexact tensor methods and their application to stochastic convex optimization. *Optimization Methods and Software*, 39(1):42–83, 2024.
- 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.
- Yair Carmon and John Duchi. Gradient descent finds the cubic-regularized nonconvex newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.
- 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.
- Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012.
- Coralia Cartis, Jaroslav Fowkes, and Zhen Shao. Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares. *arXiv preprint arXiv:2211.09873*, 2022.
- Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.
- John E Dennis, Jr and Jorge J Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.
- Michał Dereziński and Elizaveta Rebrova. Sharp analysis of sketch-and-project methods via a connection to randomized singular value decomposition. *SIAM Journal on Mathematics of Data Science*, 6(1):127–153, 2024.
- Nikita Doikov and Geovani Nunes Grapiglia. First and zeroth-order implementations of the regularized Newton method with lazy approximated Hessians. *arXiv preprint arXiv:2309.02412*, 2023.
- Nikita Doikov and Yurii E Nesterov. Inexact tensor methods with dynamic accuracies. In *ICML*, pages 2577–2586, 2020.
- Nikita Doikov, El Mahdi Chayti, and Martin Jaggi. Second-order optimization with lazy Hessians. In *International Conference on Machine Learning*. PMLR, 2023.
- Terunari Fuji, Pierre-Louis Poirion, and Akiko Takeda. Randomized subspace regularized newton method for unconstrained non-convex optimization. *arXiv preprint arXiv:2209.04170*, 2022.
- Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtárik. Rsn: randomized subspace Newton. *Advances in Neural Information Processing Systems*, 32, 2019.
- Geovani Nunes Grapiglia, Max LN Gonçalves, and GN Silva. A cubic regularization of Newton’s method with finite difference Hessian approximations. *Numerical Algorithms*, pages 1–24, 2022.
- Filip Hanzely, Nikita Doikov, Yurii Nesterov, and Peter Richtarik. Stochastic subspace cubic newton method. In *International Conference on Machine Learning*, pages 4027–4038. PMLR, 2020.
- Slavomír Hanzely. Sketch-and-project meets newton method: Global  $\mathcal{O}(k^{-2})$  convergence with low-rank updates. *arXiv preprint arXiv:2305.13082*, 2023.
- Ruichen Jiang, Parameswaran Raman, Shoham Sabach, Aryan Mokhtari, Mingyi Hong, and Volkan Cevher. Krylov cubic regularized newton: A subspace second-order method with dimension-free convergence rate. In *International Conference on Artificial Intelligence and Statistics*, pages 4411–4419. PMLR, 2024.
- Nocedal Jorge and J Wright Stephen. *Numerical optimization*. Springer, 2006.
- Dmitry Kamzolov, Klea Ziu, Artem Agafonov, and Martin Takáč. Cubic regularized quasi-Newton methods. *arXiv preprint arXiv:2302.04987*, 2023.
- Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning*, pages 1895–1904. PMLR, 2017.
- David Kozak, Stephen Becker, Alireza Doostan, and Luis Tenorio. Stochastic subspace descent. *arXiv preprint arXiv:1904.01145*, 2019.
- Jonathan Lacotte, Yifei Wang, and Mert Pilancı. Adaptive newton sketch: Linear-time optimization with quadratic convergence and effective hessian dimensionality. In *International Conference on Machine Learning*, pages 5926–5936. PMLR, 2021.
- Aurelien Lucchi and Jonas Kohler. A sub-sampled tensor method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 43(5):2856–2891, 2023.
- Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

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

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Andrei Patrascu and Ion Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61(1):19–46, 2015.

Boris T Polyak. Newton’s method and its use in optimization. *European Journal of Operational Research*, 181(3):1086–1096, 2007.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

Damien Scieur. Adaptive quasi-Newton and Anderson acceleration framework with explicit global (accelerated) convergence rates. In *International Conference on Artificial Intelligence and Statistics*, pages 883–891. PMLR, 2024.

Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Kenji Ueda and Nobuo Yamashita. Convergence properties of the regularized newton method for the unconstrained nonconvex optimization. *Applied Mathematics and Optimization*, 62:27–46, 2010.

Zhe Wang, Yi Zhou, Yingbin Liang, and Guanghui Lan. A note on inexact condition for cubic regularized newton’s method. *arXiv preprint arXiv:1808.07384*, 2018.

Peng Xu, Fred Roosta, and Michael W Mahoney. Newton-type methods for non-convex optimization under inexact hessian information. *Mathematical Programming*, 184(1):35–70, 2020.

Zhewei Yao, Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Inexact non-convex newton-type methods. *arXiv preprint arXiv:1802.06925*, 2018.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. *The Journal of Machine Learning Research*, 21(1):4130–4192, 2020.

## Checklist

1. For all models and algorithms presented, check if you include:
  - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
  - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**
  - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**
2. For any theoretical claim, check if you include:
  - (a) Statements of the full set of assumptions of all theoretical results. **Yes**
  - (b) Complete proofs of all theoretical results. **Yes**
  - (c) Clear explanations of any assumptions. **Yes**
3. For all figures and tables that present empirical results, check if you include:
  - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
  - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
  - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
  - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
  - (a) Citations of the creator If your work uses existing assets. **Yes**
  - (b) The license information of the assets, if applicable. **Yes**
  - (c) New assets either in the supplemental material or as a URL, if applicable. **Yes**
  - (d) Information about consent from data providers/curators. **Yes**
  - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**
5. If you used crowdsourcing or conducted research with human subjects, check if you include:

- (a) The full text of instructions given to participants and screenshots. **Not applicable**
- (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
- (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

## APPENDIX

### Contents

<b>1 INTRODUCTION</b>	<b>1</b>
<b>2 RELATED WORK</b>	<b>2</b>
<b>3 ALGORITHM</b>	<b>3</b>
3.1 Notation and setting . . . . .	3
3.2 Stochastic subspace cubic newton . . . . .	4
<b>4 CONVERGENCE ANALYSIS</b>	<b>5</b>
4.1 General convergence rate . . . . .	5
4.2 The power of second-order information . . . . .	5
4.3 Adaptive sampling scheme . . . . .	6
<b>5 EXPERIMENTS</b>	<b>8</b>
<b>6 LIMITATIONS</b>	<b>8</b>
<b>7 CONCLUSION</b>	<b>9</b>
<b>A PROOF OF AUXILIARY LEMMAS</b>	<b>14</b>
A.1 Model optimality . . . . .	14
<b>B CONVERGENCE OF A FIXED SAMPLING SCHEME</b>	<b>16</b>
B.1 General convergence rate . . . . .	16
B.2 The power of second-order information . . . . .	18
<b>C CONVERGENCE ADAPTIVE SCHEME - HIGH-PROBABILITY VERSION</b>	<b>21</b>
C.1 General second-order convergence . . . . .	21
C.2 Adaptive sampling . . . . .	26
<b>D EXTENDED RELATED WORK</b>	<b>30</b>
<b>E ADDITIONAL EXPERIMENTS</b>	<b>31</b>
E.1 Constant coordinate schedule on additional datasets . . . . .	31
E.2 Constant vs. exponential schedule . . . . .	32
E.3 Comparison to other methods for more datasets . . . . .	32
E.4 Convergence to an $\epsilon$ -ball . . . . .	33
E.5 Adaptive schedule on <i>gisette</i> dataset . . . . .	34
E.6 Squared norm of the step $\mathbf{h}_k$ for <i>duke</i> dataset . . . . .	35

## A PROOF OF AUXILIARY LEMMAS

### A.1 Model optimality

As mentioned earlier, in our analysis, we will assume that each iteration is performed by computing  $\mathbf{h}_k^* = \arg \min_{\mathbf{h} \in \mathbb{R}^n} [\bar{m}(\mathbf{h}) := \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M}(\mathbf{h})]$  exactly, and then we set  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k^*$ . Of course, in practice the algorithm needs to solve the subproblem only along coordinates in  $S_k$ , which has dimensionality  $\tau$ . Note that in general (for non-convex objective functions), the model  $\bar{m}(\mathbf{h})$  can also be non-convex. However, its *global minimum* is always well-defined and can be found efficiently by employing standard techniques from linear algebra (see discussion in the end of the previous section). Prior work [Cartis et al., 2011] has shown that it is possible to retain the remarkable properties of the cubic regularization algorithm with an inexact model minimizer. For our purpose, we will simply rely on first-order and second-order optimality, i.e.  $\nabla \bar{m}(\mathbf{h}_k^*) = 0$  and  $\nabla^2 \bar{m}(\mathbf{h}_k^*) \succeq 0$ , which respectively imply:

$$\begin{aligned} \nabla f(\mathbf{x}_k)_{[S_k]} + \mathbf{Q}_{k[S_k]} \mathbf{h}_k^* + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{h}_k^* &= 0, \\ \mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} + \frac{M}{2\|\mathbf{h}_k^*\|} \mathbf{h}_k^* (\mathbf{h}_k^*)^\top &\succeq 0. \end{aligned} \quad (18)$$

Based on these optimality properties, we can establish the following lemmas, which will serve as the foundational bases for our main convergence theorem.

**Proposition A.1.** *For all global minimizers  $\mathbf{h}_k^*$  of  $\bar{m}(\mathbf{h}) = \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M}(\mathbf{h})$  over  $\mathbb{R}^n$  it holds that*

$$\mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \succeq 0, \quad (19)$$

where  $\mathbf{I}$  denotes the identity matrix.

*Proof.* This proof follows closely the proof of Theorem 3.1 in Cartis et al. [2011]. From the second-order necessary optimality conditions at  $\mathbf{h}_k^*$  we have

$$\left\langle \left( \mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} + \frac{M}{2\|\mathbf{h}_k^*\|} \mathbf{h}_k^* (\mathbf{h}_k^*)^\top \right) \mathbf{w}, \mathbf{w} \right\rangle \geq 0$$

for all vectors  $\mathbf{w} \in \mathbb{R}^n$ .

If  $\mathbf{h}_k^* = 0$ , we immediately have the result. Thus, we only need to consider  $\mathbf{h}_k^* \neq 0$ . There are two cases to analyse. Firstly, suppose that  $\langle \mathbf{w}, \mathbf{h}_k^* \rangle = 0$ . Then it immediately follows

$$\left\langle \left( \mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \right) \mathbf{w}, \mathbf{w} \right\rangle \geq 0 \quad \text{for all } \mathbf{w} \in \mathbb{R}^n \text{ s.t. } \langle \mathbf{w}, \mathbf{h}_k^* \rangle = 0. \quad (20)$$

It remains to consider vectors  $\mathbf{w}$  for which  $\langle \mathbf{w}, \mathbf{h}_k^* \rangle \neq 0$ . Since  $\mathbf{w}$  and  $\mathbf{h}_k^*$  are not orthogonal, the line  $\mathbf{h}_k^* + \alpha \mathbf{w}$  intersects the ball of radius  $\|\mathbf{h}_k^*\|$  at two points,  $\mathbf{h}_k^*$  and  $\mathbf{u}_k^* \neq \mathbf{h}_k^*$ , and thus

$$\|\mathbf{u}_k^*\| = \|\mathbf{h}_k^*\|. \quad (21)$$

Let  $\mathbf{w}_k^* = \mathbf{u}_k^* - \mathbf{h}_k^*$ , and note that  $\mathbf{w}_k^*$  is parallel to  $\mathbf{w}$ , thus  $\mathbf{w} = \beta \mathbf{w}_k^*$  for some  $\beta \neq 0$ . Since  $\mathbf{h}_k^*$  is a global minimizer of  $\bar{m}(\mathbf{h})$ , we have that

$$\begin{aligned} 0 &\leq \bar{m}(\mathbf{u}_k^*) - \bar{m}(\mathbf{h}_k^*) \\ &= \langle \nabla f(\mathbf{x}_k)_{[S_k]}, (\mathbf{u}_k^* - \mathbf{h}_k^*) \rangle + \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{u}_k^*, \mathbf{u}_k^* \rangle - \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle \\ &\quad + \frac{M}{6} (\|\mathbf{u}_k^*\|^3 - \|\mathbf{h}_k^*\|^3) \\ &\stackrel{(21)}{=} \langle \nabla f(\mathbf{x}_k)_{[S_k]}, (\mathbf{u}_k^* - \mathbf{h}_k^*) \rangle + \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{u}_k^*, \mathbf{u}_k^* \rangle - \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle. \end{aligned} \quad (22)$$

But (18) gives that

$$\langle \nabla f(\mathbf{x}_k)_{[S_k]}, (\mathbf{u}_k^* - \mathbf{h}_k^*) \rangle = \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, (\mathbf{h}_k^* - \mathbf{u}_k^*) \rangle + \frac{M}{2} \|\mathbf{h}_k^*\| \langle \mathbf{h}_k^* - \mathbf{u}_k^*, \mathbf{h}_k^* \rangle. \quad (23)$$

Further note that from (21) it follows that

$$\langle \mathbf{h}_k^* - \mathbf{u}_k^*, \mathbf{h}_k^* \rangle = \frac{1}{2} \langle \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{1}{2} \langle \mathbf{u}_k^*, \mathbf{u}_k^* \rangle - \langle \mathbf{u}_k^*, \mathbf{h}_k^* \rangle = \frac{1}{2} \langle \mathbf{w}_k^*, \mathbf{w}_k^* \rangle. \quad (24)$$

Now plugging (24) in (23), and then plugging it in (22) we get

$$\begin{aligned} 0 &\leq \frac{M}{4} \|\mathbf{h}_k^*\| \langle \mathbf{w}_k^*, \mathbf{w}_k^* \rangle + \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle - \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{u}_k^* \rangle \\ &\quad + \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} \mathbf{u}_k^*, \mathbf{u}_k^* \rangle \\ &= \frac{M}{4} \|\mathbf{h}_k^*\| \langle \mathbf{w}_k^*, \mathbf{w}_k^* \rangle + \frac{1}{2} \langle \mathbf{Q}_{k[S_k]} (\mathbf{u}_k^* - \mathbf{h}_k^*), (\mathbf{u}_k^* - \mathbf{h}_k^*) \rangle \\ &\stackrel{\mathbf{w}_k^* = \mathbf{u}_k^* - \mathbf{h}_k^*}{=} \frac{1}{2} \left\langle \left( \mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \right) (\mathbf{u}_k^* - \mathbf{h}_k^*), (\mathbf{u}_k^* - \mathbf{h}_k^*) \right\rangle \\ &= \frac{1}{2\beta} \left\langle \left( \mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \right) \mathbf{w}, \mathbf{w} \right\rangle. \end{aligned} \quad (25)$$

Therefore, it holds for any  $\mathbf{w} \in \mathbb{R}^n$  such that  $\langle \mathbf{w}, \mathbf{h}_k^* \rangle \neq 0$ . Finally we can conclude from (20) and (25) that

$$\mathbf{Q}_{k[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \succeq 0,$$

which completes the proof.  $\square$

**Lemma A.2.** *For all global minimizers  $\mathbf{h}_k^*$  of  $\bar{m}(\mathbf{h}) = \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M}(\mathbf{h})$  over  $\mathbb{R}^n$  it holds that*

$$\langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle \leq 0. \quad (26)$$

*Proof.* Multiplying (19) twice with  $\mathbf{h}_k^*$  and multiplying (18) once with  $\mathbf{h}_k^*$  we get

$$\begin{aligned} \langle \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{M}{2} \|\mathbf{h}_k^*\|^3 &\geq 0, \\ \langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle + \mathbf{Q}_{k[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{M}{2} \|\mathbf{h}_k^*\|^3 &= 0. \end{aligned}$$

$\square$

Note that (19) is a stronger version of the standard second-order optimality condition (18), which takes additionally into account that  $\mathbf{h}^*$  is a *global minimum*.

**Lemma 4.3.** *For any  $\mathbf{x}_k \in \mathbb{R}^n$  and arbitrary  $S_k \subset [n]$ , let  $\mathbf{h}_k^* = \arg \min_{\mathbf{h}} \bar{m}_{\mathbf{x}_k, \nabla^2 f(\mathbf{x}_k), S_k, M}(\mathbf{h})$ . Then we have, for any  $M \geq L_2$ ,*

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{M}{12} \|\mathbf{h}_k^*\|^3. \quad (12)$$

*Proof Lemma 4.3.* By Assumption 3.1, we have for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ :

$$|f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle - \frac{1}{2} \langle \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}), \mathbf{y} - \mathbf{x} \rangle| \leq \frac{L_2}{6} \|\mathbf{y} - \mathbf{x}\|^3.$$

This implies that,

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{h}_k^* \rangle + \frac{1}{2} \langle \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{L_2}{6} \|\mathbf{h}_k^*\|^3 \\ &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{h}_k^* \rangle + \frac{1}{2} \langle \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{M}{6} \|\mathbf{h}_k^*\|^3 \\ &= f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle + \frac{1}{2} \langle \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{M}{6} \|\mathbf{h}_k^*\|^3 \\ &= \bar{m}_{\mathbf{x}_k, \nabla^2 f(\mathbf{x}_k), S_k, M}(\mathbf{h}_k^*), \end{aligned} \quad (27)$$

where we used  $M \geq L_2$  in the second inequality.

By first-order optimality of the cubic model, recall that

$$\nabla f(\mathbf{x}_k)_{[S_k]} + \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^* + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{h}_k^* = 0.$$

Taking the inner product with  $\mathbf{h}_k^*$ , we obtain

$$\langle \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle = -\langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle - \frac{M}{2} \|\mathbf{h}_k^*\|^3. \quad (28)$$

Combining Eq. (27) and (28), we obtain

$$\begin{aligned} f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) &\leq \frac{1}{2} \langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle - \frac{M}{12} \|\mathbf{h}_k^*\|^3 \\ &\leq -\frac{M}{12} \|\mathbf{h}_k^*\|^3, \end{aligned}$$

where the last inequality holds since  $\langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{h}_k^* \rangle \leq 0$  by Eq. (26).  $\square$

## B CONVERGENCE OF A FIXED SAMPLING SCHEME

In this section, we assume that in each iteration we sample a coordinate subspace  $S_k$  of the same fixed sample size  $\tau(S_k) \equiv \tau \in [n]$ . We establish the global convergence rates for our method, for an arbitrary initialization  $\mathbf{x}_0$ .

### B.1 General convergence rate

Let us prove our general result that allows for an approximate matrix  $\mathbf{Q} \approx \nabla^2 f(\mathbf{x}_k)_{[S_k]}$ .

**Theorem 4.2.** *Let the sequence  $\{\mathbf{x}_i\}$  be generated by Algorithm 1 with arbitrary  $\mathbf{Q}_k$  satisfying (8), and any fixed  $\tau \equiv \tau(S_k) \in [n]$ . Let the regularization parameter at iteration  $k \geq 0$  be chosen as*

$$M_k = 2L_2 + \frac{7^2(\sigma+L_1)^2}{2\|\nabla f(\mathbf{x}_k)_{[S_k]}\|}. \quad (10)$$

For a given accuracy level  $\varepsilon > 0$ , assume that  $\|\nabla f(\mathbf{x}_i)\| \geq \varepsilon$ , for all  $0 \leq i \leq K$ . Then, it holds

$$K \leq \frac{n}{\tau} \left[ \frac{(2+\frac{7^2}{3})(\sigma+L_1)(f(\mathbf{x}_0)-f^*)}{\varepsilon^2} + \frac{4L_2(f(\mathbf{x}_0)-f^*)}{3(\sigma+L_1)\varepsilon} \right]. \quad (11)$$

*Proof.* First, we want to choose  $M_k > 0$  such that the following progress condition is satisfied:

$$f(\mathbf{x}_k + \mathbf{h}_k^*) \leq \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h}_k^*). \quad (29)$$

We can ensure that this condition holds for a sufficiently large value of  $M_k$ . Indeed, by Lipschitzness of the Hessian, we have

$$\begin{aligned} f(\mathbf{x}_k + \mathbf{h}_k^*) &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^* \rangle + \frac{1}{2} \langle \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{L_2}{6} \|\mathbf{h}_k^*\|^3 \\ &= \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h}_k^*) + \frac{1}{2} \langle (\nabla^2 f(\mathbf{x}_k)_{[S_k]} - \mathbf{Q}_k) \mathbf{h}_k^*, \mathbf{h}_k^* \rangle + \frac{L_2 - M_k}{6} \|\mathbf{h}_k^*\|^3 \\ &\stackrel{(8)}{\leq} \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h}_k^*) + \frac{\sigma}{2} \|\mathbf{h}_k^*\|^2 + \frac{L_2 - M_k}{6} \|\mathbf{h}_k^*\|^3, \end{aligned}$$

and to satisfy (29), it is sufficient to have

$$\frac{M_k}{6} \|\mathbf{h}_k^*\|^3 \geq \frac{\sigma}{2} \|\mathbf{h}_k^*\|^2 + \frac{L_2}{6} \|\mathbf{h}_k^*\|^3. \quad (30)$$

Note that the stationary condition for  $\mathbf{h}_k^*$  gives

$$\begin{aligned} \nabla f(\mathbf{x}_k)_{[S_k]} + \mathbf{Q}_k \mathbf{h}_k^* + \frac{M_k}{2} \|\mathbf{h}_k^*\| \mathbf{h}_k^* &= 0, \\ \mathbf{Q}_k + \frac{M_k}{2} \mathbf{I} &\succeq 0. \end{aligned} \quad (31)$$

Then, observe that

$$\mathbf{Q}_k \stackrel{(8)}{\preceq} \sigma \mathbf{I} + \nabla^2 f(\mathbf{x}_k)_{[S_k]} \preceq (\sigma + L_1) \mathbf{I}, \quad (32)$$

where the second inequality is due to Assumption 4.1.

Denoting  $r := \|\mathbf{h}_k^*\|$ , we get

$$\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 \stackrel{(31)}{=} \langle (\mathbf{Q}_k + \frac{M_k r}{2} \mathbf{I})^2 \mathbf{h}_k^*, \mathbf{h}_k^* \rangle \stackrel{(32)}{\leq} (\sigma + L_1 + \frac{M_k r}{2})^2 r^2.$$

Hence,

$$\frac{M_k}{2} r^2 + (\sigma + L_1) r \geq \|\nabla f(\mathbf{x}_k)_{[S_k]}\|. \quad (33)$$

Resolving the quadratic equation, we conclude from (33) that

$$M_k \|\mathbf{h}_k^*\| = M_k r \stackrel{(33)}{\geq} \sqrt{(\sigma + L_1)^2 + 2M_k \|\nabla f(\mathbf{x}_k)_{[S_k]}\|} - \sigma - L_1. \quad (34)$$

Therefore, we can estimate the left hand side in (30) as follows,

$$\begin{aligned} \frac{M_k}{6} \|\mathbf{h}_k^*\|^3 &= \frac{M_k}{12} \|\mathbf{h}_k^*\|^3 + \frac{M_k}{12} \|\mathbf{h}_k^*\|^3 \\ &\geq \frac{\sqrt{(\sigma + L_1)^2 + 2M_k \|\nabla f(\mathbf{x}_k)_{[S_k]}\|} - \sigma - L_1}{12} \|\mathbf{h}_k^*\|^2 + \frac{M_k}{12} \|\mathbf{h}_k^*\|^3, \end{aligned}$$

and to ensure the inequality in (29), it is sufficient to choose any  $M_k \geq 2L_2 + \frac{(7\sigma + L_1)^2}{2\|\nabla f(\mathbf{x}_k)_{[S_k]}\|}$ . Let us make the following simple choice:

$$M_k := 2L_2 + \frac{7^2(\sigma + L_1)^2}{2\|\nabla f(\mathbf{x}_k)_{[S_k]}\|}. \quad (35)$$

Note that without loss of generality we assume that  $\|\nabla f(\mathbf{x}_k)_{[S_k]}\| > 0$ . Otherwise, the algorithm simply does not move for the current step, since it appears to be already in a local optimum in the sampled subspace.

Therefore, for this choice of  $M_k$ , we have established, for arbitrary  $\mathbf{y} \in \mathbb{R}^n$ :

$$f(\mathbf{x}_{k+1}) \leq \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{h}_k^*) \leq \bar{m}_{\mathbf{x}_k, \mathbf{Q}_k, S_k, M_k}(\mathbf{y} - \mathbf{x}_k),$$

where the last inequality holds since  $\mathbf{h}_k^*$  is a global minimum of the model. Hence, for any  $\mathbf{y} \in \mathbb{R}^n$ , we have:

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{y} - \mathbf{x}_k \rangle + \frac{1}{2} \langle \mathbf{Q}_k_{[S_k]}(\mathbf{y} - \mathbf{x}_k), \mathbf{y} - \mathbf{x}_k \rangle + \frac{M_k}{6} \|\mathbf{y} - \mathbf{x}_k\|^3 \\ &\stackrel{(32)}{\leq} f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k)_{[S_k]}, \mathbf{y} - \mathbf{x}_k \rangle + \frac{\sigma + L_1}{2} \|\mathbf{y} - \mathbf{x}_k\|^2 + \frac{M_k}{6} \|\mathbf{y} - \mathbf{x}_k\|^3. \end{aligned}$$

Let us take  $\mathbf{y} := \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)_{[S_k]}$  (the coordinated descent step), with some  $\alpha > 0$ . Then,

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \alpha \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 + \frac{\alpha^2(\sigma + L_1)}{2} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 + \frac{\alpha^3 M_k}{6} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^3,$$

or, rearranging the terms,

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \alpha \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 \cdot \left(1 - \frac{\alpha(\sigma + L_1)}{2} - \frac{\alpha^2 M_k}{6} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|\right).$$

Note that  $\alpha > 0$  is an arbitrary stepsize. We choose it in a way to have

$$1 - \frac{\alpha(\sigma + L_1)}{2} - \frac{\alpha^2 M_k}{6} \|\nabla f(\mathbf{x}_k)_{[S_k]}\| = \frac{1}{2},$$

which is achieved by finding the positive root of the quadratic equation. We obtain, denoting  $g := \|\nabla f(\mathbf{x}_k)_{[S_k]}\|$  and  $\beta := L_1 + \sigma$ :

$$\begin{aligned} \alpha &:= \frac{-\beta + \sqrt{\beta^2 + \frac{4}{3} M_k g}}{\frac{2}{3} M_k g} = \frac{3\beta}{2M_k g} \cdot \left( \sqrt{1 + \frac{4M_k}{3\beta^2} g} - 1 \right) \\ &= \frac{3\beta}{2M_k g} \cdot \frac{1 + \frac{4M_k}{3\beta^2} g - 1}{\sqrt{1 + \frac{4M_k}{3\beta^2} g} + 1} = \frac{2}{\beta} \cdot \frac{1}{\sqrt{1 + \frac{4M_k}{3\beta^2} g} + 1} \\ &\geq \frac{2}{\beta} \cdot \frac{1}{2 + \frac{2M_k}{3\beta^2} g}, \end{aligned}$$

where in the last bound we used the inequality  $\sqrt{1+t} \leq 1 + \frac{t}{2}$ , valid for  $t \geq 0$ .

Thus, we obtain the following progress for one step of our method:

$$\begin{aligned}
 f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \frac{1}{\beta} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 \cdot \frac{1}{2 + \frac{2M_k}{3\beta^2} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|} \\
 &= \frac{1}{2\beta} \frac{\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2}{1 + \frac{M_k}{3\beta^2} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|} \\
 &\stackrel{(35)}{=} \frac{1}{2\beta} \frac{\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2}{1 + \frac{7^2}{6} + \frac{2L_2}{3\beta^2} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|} \\
 &\geq \frac{1}{2\beta} \frac{\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2}{1 + \frac{7^2}{6} + \frac{2L_2}{3\beta^2} \|\nabla f(\mathbf{x}_k)\|},
 \end{aligned}$$

where the last inequality holds due to the trivial observation,

$$\|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 = \sum_{i \in S_k} (\nabla f(\mathbf{x}_k))_i^2 \leq \sum_{i=1}^n (\nabla f(\mathbf{x}_k))_i^2 = \|\nabla f(\mathbf{x}_k)\|^2.$$

Taking the expectation w.r.t.  $S_k$ , we get

$$\mathbb{E}_{S_k} [f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] \geq \frac{1}{2\beta} \frac{\mathbb{E}_{S_k} \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2}{1 + \frac{7^2}{6} + \frac{2L_2}{3\beta^2} \|\nabla f(\mathbf{x}_k)\|} = \frac{1}{2\beta} \frac{\left(\frac{\tau}{n}\right) \|\nabla f(\mathbf{x}_k)\|^2}{1 + \frac{7^2}{6} + \frac{2L_2}{3\beta^2} \|\nabla f(\mathbf{x}_k)\|}.$$

The last equality holds because

$$\mathbb{E}_S \|\nabla f(\mathbf{x}_k)_{[S_k]}\|^2 = \sum_{i=1}^n \mathbb{E}_S (\nabla f(\mathbf{x}_k))^{(i)}_{1_{i \in S_k}} = \frac{\tau(S_k)}{n} \|\nabla f(\mathbf{x}_k)\|^2, \quad (36)$$

where we used  $\mathbb{E}_S [(1_{i \in S})^2] = \mathbb{E}_S [(1_{i \in S})] = \frac{\tau(S)}{n}$  if  $S$  is uniformly sampled.

Assuming that  $\|\nabla f(\mathbf{x}_k)\| \geq \varepsilon$  and using monotonicity of the last expression w.r.t  $\|\nabla f(\mathbf{x}_k)\|$ , we obtain

$$\mathbb{E}_{S_k} [f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})] \geq \frac{\left(\frac{\tau}{n}\right) \varepsilon^2}{2\beta \left(1 + \frac{7^2}{6}\right) + \frac{4L_2}{3\beta} \varepsilon}.$$

Taking the full expectation and telescoping this bound for  $K$  iterations, we obtain

$$f(\mathbf{x}_0) - f^* \geq f(\mathbf{x}_0) - \mathbb{E} f(\mathbf{x}_K) \geq \frac{\left(\frac{\tau}{n}\right) \varepsilon^2 K}{2\beta \left(1 + \frac{7^2}{6}\right) + \frac{4L_2}{3\beta} \varepsilon},$$

which leads to the following global complexity:

$$K \leq \frac{n}{\tau} \cdot \frac{2\left(1 + \frac{7^2}{6}\right) \beta (f(\mathbf{x}_0) - f^*)}{\varepsilon^2} + \frac{n}{\tau} \cdot \frac{4L_2 (f(\mathbf{x}_0) - f^*)}{3\beta \varepsilon}.$$

□

## B.2 The power of second-order information

In this section, we use exact second-order information,  $\mathbf{Q}_k = \nabla^2 f(\mathbf{x}_k)_{[S_k]}$ . For better readability, let us introduce the short notation  $\tilde{m}(\mathbf{h}) := \tilde{m}_{\mathbf{x}_k, S_k, M_k}(\mathbf{h}) := \tilde{m}_{\mathbf{x}_k, S_k, \nabla^2 f(\mathbf{x}_k)_{[S_k]}, M_k}(\mathbf{h})$ . We show that in this case, it is possible to prove better convergence rates than interpolate between CD and the full Cubic Newton method.

By Lemma C.1, we have that

$$\mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \leq \sqrt{1 - \frac{\tau}{n}} \|\nabla f(\mathbf{x})\|. \quad (37)$$

And from the proof of Lemma C.2, we get

$$\mathbb{E}_S \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\|_F^2 \leq (1 - p') \|\nabla^2 f(\mathbf{x})\|_F^2, \quad (38)$$

where  $p' = \frac{\tau^2}{n^2}$ .

Let us repeat the analysis of one step of our method, employing bounds (37) and (38) directly. The goal is to refine Lemma C.4. We fix  $\mathbf{x}_k$  and consider one step of the method:

$$\mathbf{h}_k^* = \operatorname{argmin}_{\mathbf{h}} \tilde{m}_{\mathbf{x}_k, S_k, M_k}(\mathbf{h}) \quad (39)$$

for an arbitrary  $S_k \subset [n]$  and  $M \geq L_2$ . Then by first order optimality of  $h_k^*$ , the new gradient norm is bounded as

$$\begin{aligned} \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| &\leq \|\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*\| + \frac{L_2}{2} \|\mathbf{h}_k^*\|^2 \\ &\leq \|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)_{[S_k]}\| + \|(\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)_{[S_k]}) \mathbf{h}_k^*\| \\ &\quad + \frac{L_2}{2} \|\mathbf{h}_k^*\|^2 + \|\nabla f(\mathbf{x}_k)_{[S_k]} + \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*\| \\ &\stackrel{(18)}{=} \|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)_{[S_k]}\| + \|(\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)_{[S_k]}) \mathbf{h}_k^*\| \\ &\quad + \frac{L_2+M}{2} \|\mathbf{h}_k^*\|^2 \\ &\leq \|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)_{[S_k]}\| + \|\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)_{[S_k]}\| \cdot \|\mathbf{h}_k^*\| \\ &\quad + \frac{L_2+M}{2} \|\mathbf{h}_k^*\|^2. \end{aligned}$$

Now, we take (the full) expectation  $\mathbb{E}[\cdot]$  and use the Cauchy-Schwartz inequality for random variables:

$$\begin{aligned} \mathbb{E}\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| &\leq \mathbb{E}\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)_{[S_k]}\| \\ &\quad + \sqrt{\mathbb{E}[\|\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k)_{[S_k]}\|^2] \cdot \mathbb{E}[\|\mathbf{h}_k^*\|^2]} \\ &\quad + \frac{L_2+M}{2} \mathbb{E}[\|\mathbf{h}_k^*\|^2] \\ &\stackrel{(37),(38)}{\leq} \sqrt{1 - \frac{\tau}{n}} \cdot \mathbb{E}\|\nabla f(\mathbf{x}_k)\| \\ &\quad + \sqrt{(1 - p')\mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^2 \cdot \mathbb{E}[\|\mathbf{h}_k^*\|^2]} + \frac{L_2+M}{2} \mathbb{E}[\|\mathbf{h}_k^*\|^2] \\ &\stackrel{(*)}{\leq} \sqrt{1 - \frac{\tau}{n}} \cdot \mathbb{E}\|\nabla f(\mathbf{x}_k)\| + \frac{(1-p')\mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^2}{2M} + \frac{L_2+2M}{2} \mathbb{E}[\|\mathbf{h}_k^*\|^2] \\ &\leq \sqrt{1 - \frac{\tau}{n}} \cdot \mathbb{E}\|\nabla f(\mathbf{x}_k)\| + \frac{(1-p')\mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^2}{2M} + \frac{3M}{2} \mathbb{E}[\|\mathbf{h}_k^*\|^2], \end{aligned}$$

where we used Young's inequality in (\*).

Since the function  $g(x) = x^{3/2}, x \geq 0$  is convex, we have, for an arbitrary  $\lambda \in (0, 1)$ , and  $a, b \geq 0$ :

$$\begin{aligned} (a+b)^{3/2} &= \frac{1}{\lambda^{3/2}} \left( \lambda a + (1-\lambda) \left[ \frac{\lambda}{1-\lambda}(b) \right] \right)^{3/2} \\ &\leq \frac{1}{\lambda^{1/2}} a^{3/2} + \frac{1}{(1-\lambda)^{1/2}} b^{3/2}. \end{aligned} \quad (40)$$

Let us fix some  $\lambda_* \in (0, 1)$ , and use inequality (40) with  $\lambda := \lambda_* \in (0, 1)$ , obtaining:

$$\begin{aligned} &(\mathbb{E}\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\|)^{3/2} \\ &\leq \frac{1}{\lambda_*^{1/2}} \left[ 1 - \frac{\tau}{n} \right]^{3/4} (\mathbb{E}_S \|\nabla f(\mathbf{x}_k)\|)^{3/2} + \left[ \frac{1}{1-\lambda_*} \right]^{1/2} \cdot \left( \frac{(1-p')\mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^2}{2M} + \frac{3M}{2} \mathbb{E}[\|\mathbf{h}_k^*\|^2] \right)^{3/2} \\ &\leq \frac{1}{\lambda_*^{1/2}} \left[ 1 - \frac{\tau}{n} \right]^{3/4} (\mathbb{E}\|\nabla f(\mathbf{x}_k)\|)^{3/2} + \left[ \frac{2}{1-\lambda_*} \right]^{1/2} \cdot \frac{(1-p')^{3/2}}{(2M)^{3/2}} \mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^3 \\ &\quad + \left[ \frac{2}{1-\lambda_*} \right]^{1/2} \left[ \frac{3M}{2} \right]^{3/2} \mathbb{E}[\|\mathbf{h}_k^*\|^3] \end{aligned}$$

where in the last inequality we used (40) again, but with  $\lambda := \frac{1}{2}$ , as well as Jensen's inequality for the expectation. Hence, we obtain that

$$\begin{aligned} \mathbb{E}[\|\mathbf{h}_k^*\|^3] &\geq \left[\frac{1-\lambda_*}{2}\right]^{1/2} \left[\frac{2}{3M}\right]^{3/2} \left( (\mathbb{E}\|\nabla f(\mathbf{x}_{k+1})\|)^{3/2} - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} (\mathbb{E}\|\nabla f(\mathbf{x}_k)\|)^{3/2} \right) \\ &\quad - \frac{(1-p')^{3/2}}{3^{3/2}M^3} \mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^3. \end{aligned}$$

It remains to combine it with the progress of one step in terms of the functional residual (Lemma 4.3):

$$\mathbb{E}f(\mathbf{x}_k) - \mathbb{E}f(\mathbf{x}_{k+1}) \geq \frac{M}{12} \mathbb{E}[\|\mathbf{h}_k^*\|^3]$$

We have proved the following inequality.

**Lemma B.1.** *For one step of our method, with an arbitrary  $S_k \subset [n]$  of size  $\tau_k = |S_k|$ ,  $1 \leq \tau_k \leq n$ , and  $M \geq L_2$ , it holds, for any  $\lambda_* \in (0, 1)$ :*

$$\begin{aligned} &\mathbb{E}f(\mathbf{x}_k) - \mathbb{E}f(\mathbf{x}_{k+1}) \\ &\geq \frac{1}{12} \left[\frac{1-\lambda_*}{2M}\right]^{1/2} \left[\frac{2}{3}\right]^{3/2} \left( (\mathbb{E}\|\nabla f(\mathbf{x}_{k+1})\|)^{3/2} - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau_k}{n}\right]^{3/4} (\mathbb{E}\|\nabla f(\mathbf{x}_k)\|)^{3/2} \right) \\ &\quad - \frac{(1-p_{2,k})^{3/2}}{12 \cdot 3^{3/2} M^2} \mathbb{E}\|\nabla^2 f(\mathbf{x}_k)\|_F^3, \end{aligned} \tag{41}$$

where  $p_{2,k} = \frac{\tau_k(\tau_k-1)}{n(n-1)}$ .

Let us denote by  $L_1$  the Lipschitz constant of the gradient, which provides us a uniform bound for the Hessian in Frobenius norm:  $\|\nabla^2 f(\mathbf{x})\|_F \leq \sqrt{n}L_1, \forall \mathbf{x}$ .

Then, using the same subset size  $\tau_k \equiv \tau$  for all iterations, we can telescope (41) as follows:

$$\begin{aligned} f(\mathbf{x}_0) - f^* &\geq \frac{1}{12} \left[\frac{1-\lambda_*}{2M}\right]^{1/2} \left[\frac{2}{3}\right]^{3/2} \sum_{i=1}^k \left[ (\mathbb{E}\|\nabla f(\mathbf{x}_i)\|)^{3/2} - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} (\mathbb{E}\|\nabla f(\mathbf{x}_{i-1})\|)^{3/2} \right] \\ &\quad - \frac{(1-p_2)^{3/2} n^{3/2} L_1^3}{12 \cdot 3^{3/2} M^2} k \\ &= \frac{1}{12} \left[\frac{1-\lambda_*}{2M}\right]^{1/2} \left[\frac{2}{3}\right]^{3/2} \left( 1 - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} \right) \sum_{i=1}^{k-1} (\mathbb{E}\|\nabla f(\mathbf{x}_i)\|)^{3/2} \\ &\quad + \frac{1}{12} \left[\frac{1-\lambda_*}{2M}\right]^{1/2} \left[\frac{2}{3}\right]^{3/2} (\mathbb{E}\|\nabla f(\mathbf{x}_k)\|)^{3/2} \\ &\quad - \frac{1}{12} \left[\frac{1-\lambda_*}{2M\lambda_*}\right]^{1/2} \left[\frac{2}{3}\right]^{3/2} \left[1 - \frac{\tau}{n}\right]^{3/4} \|\nabla f(\mathbf{x}_0)\|^{3/2} - \frac{(1-p_2)^{3/2} n^{3/2} L_1^3}{12 \cdot 3^{3/2} M^2} k. \end{aligned}$$

Therefore, we obtain the following convergence result.

**Theorem B.2.** *For any  $M \geq L_2$ ,  $\lambda_* \in (0, 1)$ , and  $1 \leq \tau \leq n$ , it holds*

$$\begin{aligned} &\min_{1 \leq i \leq k} (\mathbb{E}\|\nabla f(\mathbf{x}_i)\|)^{3/2} \\ &\leq 12 \left[\frac{3}{2}\right]^{3/2} \left[\frac{2}{1-\lambda_*}\right]^{1/2} \left( 1 - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} \right)^{-1} \cdot \left[ \frac{\sqrt{M}(f(\mathbf{x}_0) - f^*)}{k} + \frac{(1-p')^{3/2} n^{3/2} L_1^3}{12 \cdot 3^{3/2} M^{3/2}} \right] + \frac{R_0}{k}, \end{aligned}$$

where

$$R_0 := \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} \cdot \left( 1 - \frac{1}{\lambda_*^{1/2}} \left[1 - \frac{\tau}{n}\right]^{3/4} \right)^{-1} \|\nabla f(\mathbf{x}_0)\|^{3/2}.$$

The constant  $R_0$  is a technical term which is not important. Let us ignore it for simplicity.

Note that we have a freedom in choosing parameter  $M \geq L_2$  and  $\lambda_* \in (0, 1)$ . Let us assume that all gradients are sufficiently large, for some given tolerance  $\varepsilon > 0$ :

$$\mathbb{E}\|\nabla f(\mathbf{x}_i)\| \geq \varepsilon, \quad \forall 0 \leq i \leq k.$$

We can choose  $M \geq L_2$  such that the constant term with  $L_1$  be sufficiently small, namely

$$12\left[\frac{3}{2}\right]^{3/2}\left[\frac{2}{1-\lambda_*}\right]^{1/2}\left(1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4}\right)^{-1} \cdot \frac{(1-p')^{3/2}n^{3/2}}{12 \cdot 3^{3/2}M^{3/2}}L_1^3 \leq \left(\frac{\varepsilon}{2}\right)^{3/2}.$$

E.g., this condition will be satisfied for the choice

$$\boxed{M := L_2 + \left[\frac{2}{1-\lambda_*}\right]^{1/3}\left(1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4}\right)^{-2/3}\frac{(1-p')n}{\varepsilon}L_1^2.} \quad (42)$$

In this case, the number of iterations  $k$  required to reach  $\varepsilon$  accuracy is:

$$\begin{aligned} k &= \mathcal{O}\left(\left[\frac{1}{1-\lambda_*}\right]^{1/2}\left(1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4}\right)^{-1}\frac{\sqrt{M}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}} + \frac{R_0}{\varepsilon^{3/2}}\right) \\ &\stackrel{(42)}{=} \mathcal{O}\left(\left[\frac{1}{1-\lambda_*}\right]^{1/2}\left(1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4}\right)^{-1}\cdot\frac{\sqrt{L_2}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}} \right. \\ &\quad \left. + \left[\frac{1}{1-\lambda_*}\right]^{2/3}\left(1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4}\right)^{-4/3} \cdot n^{1/2}(1-p')^{1/2}\frac{L_1(f(\mathbf{x}_0) - f^*)}{\varepsilon^2} + \frac{R_0}{\varepsilon^{3/2}}\right). \end{aligned} \quad (43)$$

It remains to choose  $\lambda_* \in (0, 1)$ .

Let us consider the following simple choice:  $\lambda_* := 1 - \frac{\tau}{n}$ . Then,

$$\begin{aligned} 1 - \frac{1}{\lambda_*^{1/2}}\left[1 - \frac{\tau}{n}\right]^{3/4} &= 1 - \left[1 - \frac{\tau}{n}\right]^{1/4} = 1 - \frac{(n-\tau)^{1/4}}{n^{1/4}} = \frac{n^{1/4} - (n-\tau)^{1/4}}{n^{1/4}} \\ &\geq \frac{[n-n+\tau]}{4n^{3/4}}\frac{1}{n^{1/4}} = \frac{\tau}{4n}, \end{aligned}$$

where we used concavity of the function  $\phi(x) := x^{1/4}, x \geq 0$ , which implies that, for any  $x, y \geq 0$ :

$$y^{1/4} = \phi(y) \leq \phi(x) + \phi'(x)(y-x) = x^{1/4} + \frac{y-x}{4x^{3/4}} \Leftrightarrow x^{1/4} - y^{1/4} \geq \frac{x-y}{4x^{3/4}}.$$

At the same time,

$$\frac{1}{1-\lambda_*} = \frac{n}{\tau}.$$

Therefore, for this choice of  $\lambda_* := 1 - \frac{\tau}{n}$  we obtain the following interpolating complexity, valid for any  $1 \leq \tau \leq n$ :

$$\boxed{k = \mathcal{O}\left(\left[\frac{n}{\tau}\right]^{3/2}\frac{\sqrt{L_2}(f(\mathbf{x}_0) - f^*)}{\varepsilon^{3/2}} + n^{1/2}(1-p')^{1/2}\left[\frac{n}{\tau}\right]^2\frac{L_1(f(\mathbf{x}_0) - f^*)}{\varepsilon^2}\right)}. \quad (44)$$

## C CONVERGENCE ADAPTIVE SCHEME - HIGH-PROBABILITY VERSION

### C.1 General second-order convergence

For instructional purposes, we first conduct an analysis under some general conditions for the quality of the approximation of the sampled coordinate gradients. In section C.2, we will demonstrate that one can remove this condition using an adaptive sampling scheme.

*Condition 1* (Condition only used in Theorem 4.2). For  $\epsilon_1, \epsilon_2 \geq 0$ , for all  $\mathbf{x} \in \mathbb{R}^n$  and any subset  $S \subset [n]$  from  $\mathcal{D}$  with probability at least  $1 - \delta$ ,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \leq \delta^{-\frac{1}{2}}\epsilon_1 \text{ and } \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\| \leq \delta^{-\frac{1}{2}}\sqrt{\epsilon_2}. \quad (45)$$

This condition is established in Lemma 4.4, with its proof provided in the subsequent two lemmas.

**Lemma C.1.** For any  $\mathbf{x} \in \mathbb{R}^n$  and any subset  $S \subset [n]$ , we have under an uniform sampling scheme

$$\mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \leq \sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\| =: \epsilon_1. \quad (46)$$

This implies a high-probability bound

$$\Pr \left( \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \geq \delta^{-\frac{1}{2}} \underbrace{\sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\|}_{=\epsilon_1} \right) \leq \delta. \quad (47)$$

*Proof.* For any  $\mathbf{x} \in \mathbb{R}^n$ , we have

$$\begin{aligned} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\|^2 &= \sum_{i=1}^n (\nabla f(\mathbf{x})^{(i)} - \nabla f(\mathbf{x})^{(i)} 1_{i \in S})^2 \\ &= \sum_{i=1}^n (\nabla f(\mathbf{x})^{(i)})^2 + (\nabla f(\mathbf{x})^{(i)} 1_{i \in S})^2 - 2(\nabla f(\mathbf{x})^{(i)})^2 1_{i \in S}. \end{aligned}$$

Taking expectation over  $S$ , we get

$$\begin{aligned} \mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\|^2 &= \sum_{i=1}^n \mathbb{E}_S (\nabla f(\mathbf{x})^{(i)} - \nabla f(\mathbf{x})^{(i)} 1_{i \in S})^2 \\ &= \sum_{i=1}^n (\nabla f(\mathbf{x})^{(i)})^2 + \mathbb{E}_S (\nabla f(\mathbf{x})^{(i)} 1_{i \in S})^2 - 2\mathbb{E}_S (\nabla f(\mathbf{x})^{(i)})^2 1_{i \in S} \\ &\stackrel{(i)}{=} \sum_{i=1}^n (\nabla f(\mathbf{x})^{(i)})^2 (1 + \frac{\tau(S)}{n} - 2\frac{\tau(S)}{n}) \\ &= (1 - \frac{\tau(S)}{n}) \|\nabla f(\mathbf{x})\|^2, \end{aligned}$$

where (i) used  $\mathbb{E}_S[(1_{i \in S})^2] = \mathbb{E}_S[(1_{i \in S})] = \frac{\tau(S)}{n}$ .

We conclude the expectation bound using Jensen's inequality,

$$\mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \leq (\mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\|^2)^{1/2} \leq \sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\|. \quad (48)$$

Finally, we can obtain a high-probability bound using Markov's inequality as follows:

$$\Pr(\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \geq \mu) = \Pr(\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\|^2 \geq \mu^2) \leq \mu^{-2} \mathbb{E}_S \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\|^2. \quad (49)$$

Setting  $\mu = \delta^{-\frac{1}{2}} \sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\|$  for  $\delta \in (0, 1)$  yields:

$$\Pr \left( \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x})_{[S]}\| \geq \delta^{-\frac{1}{2}} \underbrace{\sqrt{1 - \frac{\tau(S)}{n}} \|\nabla f(\mathbf{x})\|}_{=\epsilon_1} \right) \leq \delta. \quad (50)$$

□

**Lemma C.2.** For any  $\mathbf{x} \in \mathbb{R}^n$  and any subset  $S \subset [n]$ , we have under an uniform sampling scheme

$$\mathbb{E}_S \|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_2 \leq \sqrt{1 - p'} \|\nabla^2 f(\mathbf{x})\|_F =: \sqrt{\epsilon_2}, \quad (51)$$

where  $p' := \left(\frac{\tau(S)}{n}\right)^2$ . This implies a high-probability bound

$$\Pr\left(\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\|_2 \geq \delta^{-\frac{1}{2}} \underbrace{\sqrt{1-p'} \|\nabla^2 f(\mathbf{x})\|_F}_{=\sqrt{\epsilon_2}}\right) \leq \delta. \quad (52)$$

*Proof.*

$$\mathbb{E}_S \|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_F^2 = \mathbb{E}_S \sum_{i,j} ((\nabla^2 f(\mathbf{x}))_{ij}^2 1_{i,j \in S} + (\nabla^2 f(\mathbf{x}))_{ij}^2 - 2(\nabla^2 f(\mathbf{x}))_{ij}^2 1_{i,j \in S}) \quad (53)$$

$$= \sum_{i,j} (\nabla^2 f(\mathbf{x}))_{ij}^2 \mathbb{E}_S [1_{i,j \in S}^2 + 1 - 2 \cdot 1_{i,j \in S}] \quad (54)$$

$$= (1-p') \sum_{i,j} (\nabla^2 f(\mathbf{x}))_{ij}^2 \quad (55)$$

$$= (1-p') \|\nabla^2 f(\mathbf{x})\|_F^2, \quad (56)$$

where  $p' := \mathbb{E}_S [1_{i,j \in S}] = p^2 = \left(\frac{\tau(S)}{n}\right)^2$ .

Since  $\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F$  for any matrix  $\mathbf{A}$ , we get the desired result, again using Jensen's inequality:

$$\mathbb{E}_S \|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_2 \leq \sqrt{1-p'} \|\nabla^2 f(\mathbf{x})\|_F.$$

We can derive a high probability bound by Markov's inequality since

$$\begin{aligned} \Pr(\|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_2 \geq \mu) &= \Pr(\|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_2^2 \geq \mu^2) \leq \frac{\mathbb{E}_S \|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_2^2}{\mu^2} \\ &\leq \frac{\mathbb{E}_S \|\nabla^2 f(\mathbf{x})_{[S]} - \nabla^2 f(\mathbf{x})\|_F^2}{\mu^2} \end{aligned} \quad (57)$$

Setting  $\mu = \delta^{-\frac{1}{2}} \sqrt{1 - \left(\frac{\tau(S)}{n}\right)^2} \|\nabla^2 f(\mathbf{x})\|_F$  for  $\delta \in (0, 1)$  yields:

$$\Pr\left(\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\|_2 \geq \delta^{-\frac{1}{2}} \underbrace{\sqrt{1-p'} \|\nabla^2 f(\mathbf{x})\|_F}_{=\sqrt{\epsilon_2}}\right) \leq \delta. \quad (58)$$

□

We are now ready to state the first main result of this section.

**Theorem C.3.** Let the sequence  $\{\mathbf{x}_i\}$  be generated by  $\mathbf{x}_{k+1} = \mathbf{x}_k + \arg \min_{\mathbf{h}} \tilde{m}_{\mathbf{x}_k, S_k, M_k}(\mathbf{h})$  and let  $M \geq L_2$ . Assume that the objective function  $f(\mathbf{x})$  is bounded below:

$$f(\mathbf{x}) \geq f^* \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

Let  $\Delta_0 = f(\mathbf{x}_0) - f^*$ , and define the following constants (dependent on  $M$ ):

$$C = \frac{12\sqrt{3}}{M} \left(\frac{2M+1}{2}\right)^{3/2}, D = 162M^2.$$

Then, under Condition 1 we have with probability at least  $1 - \delta$ ,

$$\min_{1 \leq j \leq k} \mu(\mathbf{x}_j) \leq \max(C, D) \frac{\Delta_0}{k} + \sqrt{3} \delta^{-3/4} \epsilon_1^{3/2} + 4\delta^{-3/2} \epsilon_2^{3/2}.$$

Theorem C.3 states that Algorithm 1 converges to an  $\epsilon$ -second-order stationary point at a rate of  $\mathcal{O}(\epsilon^{-3/2}, \epsilon^{-3})$ , up to a ball whose radius is determined by  $\epsilon_1$  and  $\epsilon_2$ . We also expect that the constants in the bound could be made tighter even if it results in a somewhat less readable proof. However, the interesting aspect of this theorem is that it shows that we obtain the same convergence as cubic regularization up to a ball. Next, we turn our attention to characterizing how  $\epsilon_1$  and  $\epsilon_2$  depend on the number of sampled coordinates.

In order to prove the theorem, we will first prove two lemmas, Lemma C.4 and C.5, that relate the gradient and the Hessian of the objective function  $f$  with the norm of the step  $\mathbf{h}_k^*$ .

**Lemma C.4.** *For any  $\mathbf{x}_k \in \mathbb{R}^n$ , let  $\mathbf{h}_k^* = \arg \min_{\mathbf{h}} m_3(\mathbf{h}; \mathbf{x}_k, S_k)$  for an arbitrary  $S_k \subset [n]$  and let  $M \geq L_2$ . Then, the full gradient norm  $\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\|$  at the new iterate can be bounded with probability at least  $1 - \delta$  as*

$$\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| \leq \frac{2M+1}{2} \|\mathbf{h}_k^*\|^2 + \delta^{-1/2} \epsilon_1 + \frac{1}{2} \delta^{-1} \epsilon_2.$$

*Proof.* By the triangle inequality, we have:

$$\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| \leq \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*) - \nabla \phi(\mathbf{h}_k^*)\| + \|\nabla \phi(\mathbf{h}_k^*)\|, \quad (59)$$

where  $\nabla \phi(\mathbf{h}_k^*) = \nabla f(\mathbf{x}_k)_{[S_k]} + \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*$ .

From the first-order optimality condition in (18) it follows that  $\|\nabla \phi(\mathbf{h}_k^*)\| = \frac{M}{2} \|\mathbf{h}_k^*\|^2$ . Therefore we focus on the first term, for which

$$\begin{aligned} & \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*) - \nabla f(\mathbf{x}_k)_{[S_k]} - \nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^*\| \\ & \leq \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*) - \nabla f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*\| \\ & \quad + \|\nabla f(\mathbf{x}_k)_{[S_k]} - \nabla f(\mathbf{x}_k)\| + \|\nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^* - \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*\| \\ & \stackrel{\text{Assumption 3.1}}{\leq} \frac{L_2}{2} \|\mathbf{h}_k^*\|^2 + \|\nabla f(\mathbf{x}_k)_{[S_k]} - \nabla f(\mathbf{x}_k)\| + \|\nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^* - \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*\|. \end{aligned} \quad (60)$$

From Eq. (59) and Eq. (60) we have that

$$\begin{aligned} \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| & \leq \frac{M+L_2}{2} \|\mathbf{h}_k^*\|^2 + \|\nabla f(\mathbf{x}_k)_{[S_k]} - \nabla f(\mathbf{x}_k)\| + \|\nabla^2 f(\mathbf{x}_k)_{[S_k]} \mathbf{h}_k^* - \nabla^2 f(\mathbf{x}_k) \mathbf{h}_k^*\| \\ & \leq \frac{M+L_2}{2} \|\mathbf{h}_k^*\|^2 + \|\nabla f(\mathbf{x}_k)_{[S_k]} - \nabla f(\mathbf{x}_k)\| + \|\nabla^2 f(\mathbf{x}_k)_{[S_k]} - \nabla^2 f(\mathbf{x}_k)\| \|\mathbf{h}_k^*\| \\ & \stackrel{(i)}{\leq} \frac{M+L_2}{2} \|\mathbf{h}_k^*\|^2 + \delta^{-1/2} \epsilon_1 + \sqrt{\delta^{-1} \epsilon_2} \|\mathbf{h}_k^*\| \\ & \stackrel{(ii)}{\leq} \frac{M+L_2}{2} \|\mathbf{h}_k^*\|^2 + \delta^{-1/2} \epsilon_1 + \frac{1}{2} \delta^{-1} \epsilon_2 + \frac{1}{2} \|\mathbf{h}_k^*\|^2 \\ & \leq \frac{2M+1}{2} \|\mathbf{h}_k^*\|^2 + \delta^{-1/2} \epsilon_1 + \frac{1}{2} \delta^{-1} \epsilon_2, \end{aligned}$$

where (i) uses Eq. (47) and (52) and (ii) uses Young's inequality for products.  $\square$

**Lemma C.5.** *For any  $\mathbf{x}_k \in \mathbb{R}^n$ , let  $\mathbf{h}_k^* = \arg \min_{\mathbf{h}} m_3(\mathbf{h}; \mathbf{x}_k, S_k)$  for an arbitrary  $S_k \subset [n]$ . Then, the smallest eigenvalue at the new iterate can be bounded with probability at least  $1 - \delta$  as*

$$-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)) \leq \frac{3M}{2} \|\mathbf{h}_k^*\| + \delta^{-1} \sqrt{\epsilon_2}. \quad (61)$$

*Proof.* Recall that

$$\|(\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y}))\| \leq L_2 \|\mathbf{x} - \mathbf{y}\|, \quad (62)$$

By Eq. (19),  $\nabla^2 f(\mathbf{x}_k)_{[S_k]} + \frac{M}{2} \|\mathbf{h}_k^*\| \cdot \mathbf{I} \succeq 0$ . Therefore

$$\begin{aligned} \nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*) &\stackrel{(62)}{\succeq} \nabla^2 f(\mathbf{x}_k) - L_2 \|\mathbf{h}_k^*\| \mathbf{I} \\ &\stackrel{(52)}{\succeq} \nabla^2 f(\mathbf{x}_k)_{[S_k]} - \sqrt{\delta^{-1}\epsilon_2} \mathbf{I} - L_2 \|\mathbf{h}_k^*\| \mathbf{I} \\ &\stackrel{\text{Prop. A.1}}{\succeq} -\left(\frac{1}{2}M + L_2\right) \|\mathbf{h}_k^*\| \mathbf{I} - \sqrt{\delta^{-1}\epsilon_2} \mathbf{I} \\ &\succeq -\frac{3M}{2} \|\mathbf{h}_k^*\| \mathbf{I} - \sqrt{\delta^{-1}\epsilon_2} \mathbf{I}, \end{aligned}$$

where we used  $L_2 \leq M$ . Therefore, we have

$$\|\mathbf{h}_k^*\| \geq \frac{2}{3M} \left( -\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)) - \sqrt{\delta^{-1}\epsilon_2} \right), \quad (63)$$

which implies

$$-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)) \leq \frac{3M}{2} \|\mathbf{h}_k^*\| + \sqrt{\delta^{-1}\epsilon_2}. \quad (64)$$

□

Now we are ready to prove Theorem C.3.

*Proof of Theorem C.3.* By the convexity of the function  $g(x) = x^{3/2}, x \geq 0$ , Jensen's inequality yields  $(\sum_{i=1}^d x_i)^{3/2} \leq \sqrt{d} \sum_{i=1}^d (x_i)^{3/2}$ . Applied to the result of Lemma C.4, we obtain

$$\begin{aligned} \|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\|^{3/2} &\leq \left( \frac{M+L_2+1}{2} \|\mathbf{h}_k^*\|^2 + \delta^{-1/2}\epsilon_1 + \delta^{-1}\epsilon_2 \right)^{3/2} \\ &\leq \sqrt{3} \left( \frac{M+L_2+1}{2} \right)^{3/2} \|\mathbf{h}_k^*\|^3 + \sqrt{3}\delta^{-3/4}\epsilon_1^{3/2} + \sqrt{3}\delta^{-3/2}\epsilon_2^{3/2} \\ &\leq \sqrt{3} \left( \frac{2M+1}{2} \right)^{3/2} \|\mathbf{h}_k^*\|^3 + \sqrt{3}\delta^{-3/4}\epsilon_1^{3/2} + \sqrt{3}\delta^{-3/2}\epsilon_2^{3/2}. \end{aligned} \quad (65)$$

Let  $C_M = \frac{M}{12\sqrt{3}} \left( \frac{2}{2M+1} \right)^{3/2}$ . Using a telescoping argument, we have

$$\begin{aligned} f(\mathbf{x}_0) - f^* &\geq \sum_{i=0}^{k-1} (f(\mathbf{x}_i) - f(\mathbf{x}_{i+1})) \\ &\stackrel{\text{Lem. 4.3}}{\geq} \sum_{i=0}^{k-1} \frac{M}{12} \|\mathbf{h}_i^*\|^3 \\ &\stackrel{\text{Lem. C.4}}{\geq} C_M \sum_{i=0}^{k-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} - \sqrt{3}k\delta^{-3/4}C_M\epsilon_1^{3/2} - \sqrt{3}k\delta^{-3/2}C_M\epsilon_2^{3/2} \\ &\geq C_M k \min_{0 \leq i \leq k-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} - \sqrt{3}k\delta^{-3/4}C_M\epsilon_1^{3/2} - \sqrt{3}k\delta^{-3/2}C_M\epsilon_2^{3/2} \\ &\geq C_M k \min_{1 \leq i \leq k} \|\nabla f(\mathbf{x}_i)\|^{3/2} - \sqrt{3}k\delta^{-3/4}C_M\epsilon_1^{3/2} - \sqrt{3}k\delta^{-3/2}C_M\epsilon_2^{3/2}. \end{aligned}$$

Now rearranging for  $\|\nabla f(\mathbf{x}_j)\|$  we get

$$\min_{1 \leq i \leq k} \|\nabla f(\mathbf{x}_i)\|^{3/2} \leq \frac{1}{C_M k} (f(\mathbf{x}_0) - f^*) + \sqrt{3}\delta^{-3/4}\epsilon_1^{3/2} + \sqrt{3}\delta^{-3/2}\epsilon_2^{3/2}. \quad (66)$$

We can also obtain a guarantee in terms of second-order optimality as follows. By Lemma C.5, we have  $-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)) \leq \frac{3M}{2} \|\mathbf{h}_k^*\| + \sqrt{\delta^{-1}} \epsilon_2$ . By the convexity of the function  $g(x) = x^3$  on  $\mathbb{R}^+$ , Jensen's inequality yields  $(\sum_{i=1}^d x_i)^3 \leq d^2 \sum_{i=1}^d (x_i)^3$ , therefore

$$\begin{aligned} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)))^3 &\leq \frac{27M^3}{2} \|\mathbf{h}_k^*\|^3 + 4\delta^{-3/2} \epsilon_2^{3/2} \\ \implies \|\mathbf{h}_k^*\|^3 &\geq \frac{2}{27M^3} \left( (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)))^3 - 4\delta^{-3/2} \epsilon_2^{3/2} \right). \end{aligned} \quad (67)$$

Using a telescoping argument, we have

$$\begin{aligned} f(\mathbf{x}_0) - f^* &\geq \sum_{i=0}^{k-1} f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \\ &\stackrel{\text{Lem. 4.3}}{\geq} \sum_{i=0}^{k-1} \frac{M}{12} \|\mathbf{h}_i^*\|^3 \\ &\stackrel{(67)}{\geq} \frac{1}{162M^2} \sum_{i=0}^{k-1} \left( (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 - 4\delta^{-3/2} \epsilon_2^{3/2} \right) \\ &\geq \frac{1}{162M^2} \sum_{i=0}^{k-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 - \frac{2}{81M^2} k \delta^{-3/2} \epsilon_2^{3/2} \\ &\geq \frac{1}{162M^2} \min_{1 \leq i \leq k} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 - \frac{2}{81M^2} k \delta^{-3/2} \epsilon_2^{3/2}. \end{aligned}$$

By rearranging, we get

$$\min_{1 \leq i \leq k} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)))^3 \leq \frac{162M^2}{k} (f(\mathbf{x}_0) - f^*) + 4\delta^{-3/2} \epsilon_2^{3/2}. \quad (68)$$

□

## C.2 Adaptive sampling

We now present our main convergence result when using the adaptive sampling scheme described in Section 4.3.

*Remark C.6.* Note that the number of sampled coordinates,  $\tau(S)$ , does not need to be the same for every iteration. Since we analyze the progress of each iteration  $k \geq 0$  separately, we allow for different probability distributions when selecting the coordinates at each iteration. In Lemma 4.4, we consider the distribution of selecting a particular coordinate subset, conditioning on the previous iterate. Then, in the final bound we consider the randomness over the sigma-algebra of all the iterates  $x_k$ . In practice, one can choose the distribution for  $\tau_k$  to be uniform for each  $k \geq 0$ , but we emphasize once more that  $\tau_k$  can vary for different  $k$  (for instance we can sample fewer coordinates at the beginning of the optimization process).

**Theorem 4.6.** Consider the sequence  $\{\mathbf{x}_k\}_{k=0}^K$  generated by  $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k^*$  where  $\tau$  satisfies Eq. (17) with  $\epsilon_1 = \epsilon_2 = c_{k-1} \cdot \|\mathbf{h}_{k-1}^*\|^2$  for some  $c_{k-1} > 0$ ,  $\forall k = 0, \dots, K$ . Let  $M \geq L_2$ . Let  $\Delta_0 = f(\mathbf{x}_0) - f^*$ , and define the following constants (dependent on  $M$ ):

$$\begin{aligned} C_M &= \left( (2M+1)^{3/2} + (4\delta^{-3/4} + \sqrt{2}\delta^{-3/2}) \max_i c_i^{3/2} \right)^{-1}, \\ D_M &= \left( \frac{27M^3}{2} + 4\delta^{-3/2} \max_i c_i^{3/2} \right)^{-1}. \end{aligned}$$

Then with probability at least  $1 - \delta$ , we have

$$\begin{aligned} \min_{1 \leq j \leq K} \mu(\mathbf{x}_j) &\leq \frac{1}{K} \left( \frac{6}{M} \max(C_M^{-1}, D_M^{-1}) \Delta_0 \right. \\ &\quad \left. + 4\delta^{-3/2} c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right), \end{aligned}$$

where  $\mathbf{h}_{-1}^*$  is such that  
 $\mathbb{E}_{[S]} \|\nabla^2 f(\mathbf{x}_0) - \nabla^2 f(\mathbf{x}_0)_{[S]}\| \leq \sqrt{c_{-1}} \|\mathbf{h}_{-1}^*\|.$

*Proof.* By Lemma C.4,

$$\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\| \leq \frac{2M+1}{2} \|\mathbf{h}_k^*\|^2 + \frac{(2\delta^{-1/2} + \delta^{-1})c_{k-1}}{2} \|\mathbf{h}_{k-1}^*\|^2. \quad (69)$$

By the convexity of the function  $g(x) = x^{3/2}$ , Jensen's inequality yields  $(\frac{1}{d} \sum_{i=1}^d \mathbf{x}_i)^{3/2} \leq \frac{1}{d} \sum_{i=1}^d (\mathbf{x}_i)^{3/2}$ . Therefore

$$\|\nabla f(\mathbf{x}_k + \mathbf{h}_k^*)\|^{3/2} \leq \frac{1}{2} ((2M+1)^{3/2} \|\mathbf{h}_k^*\|^3 + ((2\delta^{-1/2} + \delta^{-1})c_{k-1})^{3/2} \|\mathbf{h}_{k-1}^*\|^3). \quad (70)$$

Let  $\mathbf{h}_{-1}^*$  be such that  $\mathbb{E}_{[S]} \|\nabla^2 f(\mathbf{x}_0) - \nabla^2 f(\mathbf{x}_0)_{[S]}\| \leq \sqrt{c_{-1}} \|\mathbf{h}_{-1}^*\|$ . Summing Eq. (70) over  $K$ , we get

$$\begin{aligned} \sum_{i=0}^{K-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} &\leq \frac{1}{2} \left( (2M+1)^{3/2} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 + \sum_{i=0}^{K-1} ((2\delta^{-1/2} + \delta^{-1})c_{i-1})^{3/2} \|\mathbf{h}_{i-1}^*\|^3 \right) \\ &\leq \frac{1}{2} \left( (2M+1)^{3/2} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 + \sum_{i=0}^{K-1} ((2\delta^{-1/2} + \delta^{-1})c_i)^{3/2} \|\mathbf{h}_i^*\|^3 \right. \\ &\quad \left. + ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right) \\ &\leq \frac{1}{2} \left( (2M+1)^{3/2} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 + (2\delta^{-1/2} + \delta^{-1})^{3/2} \max c_i^{3/2} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \right. \\ &\quad \left. + ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right) \\ &\leq \frac{1}{2} \left( ((2M+1)^{3/2} + (4\delta^{-3/4} + \sqrt{2}\delta^{-3/2}) \max c_i^{3/2}) \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \right. \\ &\quad \left. + ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right) \end{aligned}$$

which implies

$$\sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \geq 2C_M \sum_{i=0}^{K-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} - ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} C_M \|\mathbf{h}_{-1}^*\|^3. \quad (71)$$

where  $C_M := ((2M+1)^{3/2} + (4\delta^{-3/4} + \sqrt{2}\delta^{-3/2}) \max c_i^{3/2})^{-1}$ .

Using a telescoping argument, we have

$$\begin{aligned} f(\mathbf{x}_0) - f^* &\geq \sum_{i=0}^{K-1} f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \\ &\stackrel{\text{Eq. (12)}}{\geq} \frac{M}{12} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \\ &\stackrel{\text{Eq. (71)}}{\geq} \frac{MC_M}{6} \sum_{i=0}^{K-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} - \frac{M}{12} ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} C_M \|\mathbf{h}_{-1}^*\|^3 \end{aligned}$$

Rearranging terms, we obtain

$$\begin{aligned} (f(\mathbf{x}_0) - f^*) + \frac{M}{12} ((2\delta^{-1/2} + \delta^{-1})c_{-1})^{3/2} C_M \|\mathbf{h}_{-1}^*\|^3 &\geq \frac{MC_M}{6} \sum_{i=0}^{K-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} \\ &\geq \frac{MC_M}{6} K \min_{0 \leq i \leq K-1} \|\nabla f(\mathbf{x}_i + \mathbf{h}_i^*)\|^{3/2} \\ &= \frac{MC_M}{6} K \min_{1 \leq i \leq K} \|\nabla f(\mathbf{x}_i)\|^{3/2}. \end{aligned} \quad (72)$$

Now rearranging for  $\|\nabla f(\mathbf{x}_i)\|$  we get

$$\min_{1 \leq i \leq K} \|\nabla f(\mathbf{x}_i)\|^{3/2} \leq \frac{1}{K} \left( \frac{6}{MC_M} (f(\mathbf{x}_0) - f^*) + \frac{1}{2} ((2\delta^{-1/2} + \delta^{-1}) c_{-1})^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right), \quad (73)$$

which implies  $\min_{1 \leq j \leq K} \|\nabla f(\mathbf{x}_j)\| = \mathcal{O}(K^{-2/3})$ .

We can also obtain a guarantee in terms of second-order optimality as follows. By Lemma C.5,

$$-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)) \leq \frac{3M}{2} \|\mathbf{h}_k^*\| + \sqrt{\delta^{-1} \epsilon_2}. \quad (74)$$

By the convexity of the function  $g(x) = x^3$  on  $\mathbb{R}^+$ , Jensen's inequality yields  $(\frac{1}{d} \sum_{i=1}^d \mathbf{x}_i)^3 \leq \frac{1}{d} \sum_{i=1}^d (\mathbf{x}_i)^3$ , therefore

$$\begin{aligned} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_k + \mathbf{h}_k^*)))^3 &\stackrel{\text{monotonicity of } x^3}{\leq} \left( \frac{1}{2} \left( 3M \|\mathbf{h}_k^*\| + 2\sqrt{\delta^{-1} \epsilon_2} \right) \right)^3 \\ &\stackrel{\text{Jensen}}{\leq} \frac{1}{2} \left( 27M^3 \|\mathbf{h}_k^*\|^3 + 8\delta^{-3/2} \epsilon_2^{3/2} \right) \\ &= \frac{1}{2} \left( 27M^3 \|\mathbf{h}_k^*\|^3 + 8\delta^{-3/2} c_{k-1}^{3/2} \|\mathbf{h}_{k-1}^*\|^3 \right). \end{aligned} \quad (75)$$

Let  $\mathbf{h}_{-1}^*$  be such that  $\mathbb{E}_S \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x})_{[S]}\| \leq \sqrt{c_{-1}} \|\mathbf{h}_{-1}^*\|$ . Summing Eq. (75) over  $K$ , we get

$$\begin{aligned} \sum_{i=0}^{K-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 &\leq \frac{27M^3}{2} \sum_{i=0}^{k-1} \|\mathbf{h}_i^*\|^3 + 4 \sum_{i=0}^{k-1} \delta^{-3/2} c_{i-1}^{3/2} \|\mathbf{h}_{i-1}^*\|^3 \\ &\leq \frac{27M^3}{2} \sum_{i=0}^{k-1} \|\mathbf{h}_i^*\|^3 + 4 \left( \sum_{i=0}^{k-1} \delta^{-3/2} c_i^{3/2} \|\mathbf{h}_i^*\|^3 + \delta^{-3/2} c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3 \right) \\ &\leq \left( \frac{27M^3}{2} + 4\delta^{-3/2} \max_i c_i^{3/2} \right) \sum_{i=0}^{k-1} \|\mathbf{h}_i^*\|^3 + 4\delta^{-3/2} c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3, \end{aligned}$$

which implies

$$\sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \geq D_M \sum_{i=0}^{K-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 - 4\delta^{-3/2} D_M c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3, \quad (76)$$

where  $D_M = \left( \frac{27M^3}{2} + 4\delta^{-3/2} \max_i c_i^{3/2} \right)^{-1}$ .

Using a telescoping argument, we have

$$\begin{aligned} f(\mathbf{x}_0) - f^* &\geq \sum_{i=0}^{K-1} f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \\ &\stackrel{\text{Eq. (12)}}{\geq} \frac{M}{12} \sum_{i=0}^{K-1} \|\mathbf{h}_i^*\|^3 \\ &\stackrel{\text{Eq. (76)}}{\geq} \frac{M}{12} D_M \sum_{i=0}^{K-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 - \frac{M}{3} \delta^{-3/2} D_M c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3. \end{aligned}$$

By rearranging terms, we obtain

$$\begin{aligned} (f(\mathbf{x}_0) - f^*) + \frac{M}{3} \delta^{-3/2} D_M c_{-1}^{3/2} \|\mathbf{h}_{-1}^*\|^3 &\geq \frac{M}{12} D_M \sum_{i=0}^{K-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 \\ &\geq \frac{M}{12} D_M K \min_{0 \leq i \leq K-1} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i + \mathbf{h}_i^*)))^3 \\ &\geq \frac{M}{12} D_M K \min_{1 \leq i \leq K} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i)))^3. \end{aligned}$$

Thus, we get

$$\min_{1 \leq i \leq K} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_i)))^3 \leq \frac{1}{K} \left( \frac{12}{MD_M} (f(\mathbf{x}_0) - f^*) + 4\delta^{-3/2} c_{-1}^{3/2} \|\mathbf{h}_*^*\|^3 \right),$$

which implies  $\min_{1 \leq j \leq K} (-\lambda_{\min}(\nabla^2 f(\mathbf{x}_j))) = \mathcal{O}(K^{-1/3})$ .

□

## D EXTENDED RELATED WORK

In the literature random subspace methods are also widely known as sketch-and-project methods. In the convex setting, the sketched Newton method proposed in Hanzely [2023] is shown to converge at a global rate of  $\mathcal{O}(k^{-2})$ . More recently, Dereziński and Rebrova [2024] derive sharp convergence rates for sketch-and-project methods to iteratively solve linear systems via a connection to randomized singular value decomposition. They extend their setting to the minimization of convex function with stochastic Newton methods. In Lacotte et al. [2021], the authors make a connection between the sketch size and the efficient Hessian dimensionality and show quadratic convergence for self-concordant, strongly convex functions.

Extensions of stochastic Newton methods to inexact tensor methods for convex objectives were proposed in Lucchi and Kohler [2023], Agafonov et al. [2024], Doikov and Nesterov [2020]. In the context of non-convex optimization, Cartis et al. [2022] derive a high-probability bound for convergence to a first-order stationary bound for randomised subspace methods, which are safeguarded by trust region or quadratic regularization with a complexity of  $\mathcal{O}(k^{-1/2})$ , which is the same rate as for gradient-based methods [Nesterov, 2018].

## E ADDITIONAL EXPERIMENTS

**Used datasets and license** The datasets used in the experiments are all taken from LibSVM [Chang and Lin, 2011], which are provided under a modified BSD license.

### E.1 Constant coordinate schedule on additional datasets

We verified our theoretical results also on three other datasets *duke* ( $n = 7129, d = 44$ ), *madelon* ( $n = 500, d = 2000$ ) and *realsim* ( $n = 20.958, d = 72.309$ ). The convergence results can be found in Figure 4.

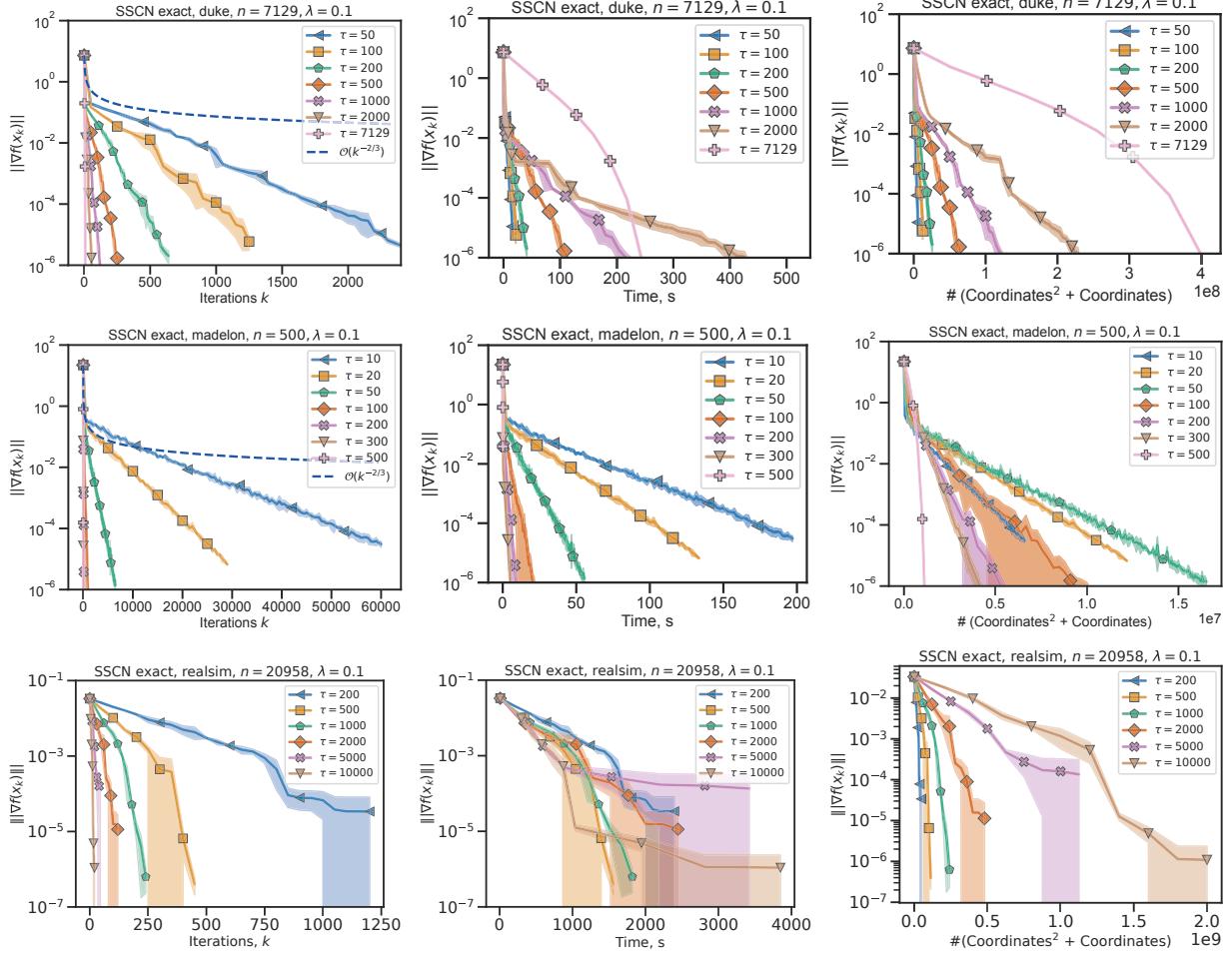


Figure 4: Convergence of different constant coordinate schedules measured w.r.t. iterations (first column), time (second column) and  $\#(\text{Coordinates}^2 + \text{Coordinates})$  evaluated (third column) averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for two datasets. First row: *duke*, second row: *madelon*, third row: *realsim*.

## E.2 Constant vs. exponential schedule

As discussed earlier, the adaptive sampling scheme presented in Section 4.3 suggests using an exponential schedule to sample coordinates. We compare this schedule to a constant schedule in Figure 5. We observe that for the two high-dimensional datasets *gisette* and *duke*, **the best constant schedule and the best exponential schedule perform on par both in terms of time and # coordinates evaluated**. We conclude that in some cases a simple constant schedule can already perform sufficiently well. However, an exponential schedule might still be more appropriate if one needs a high-accuracy solution.

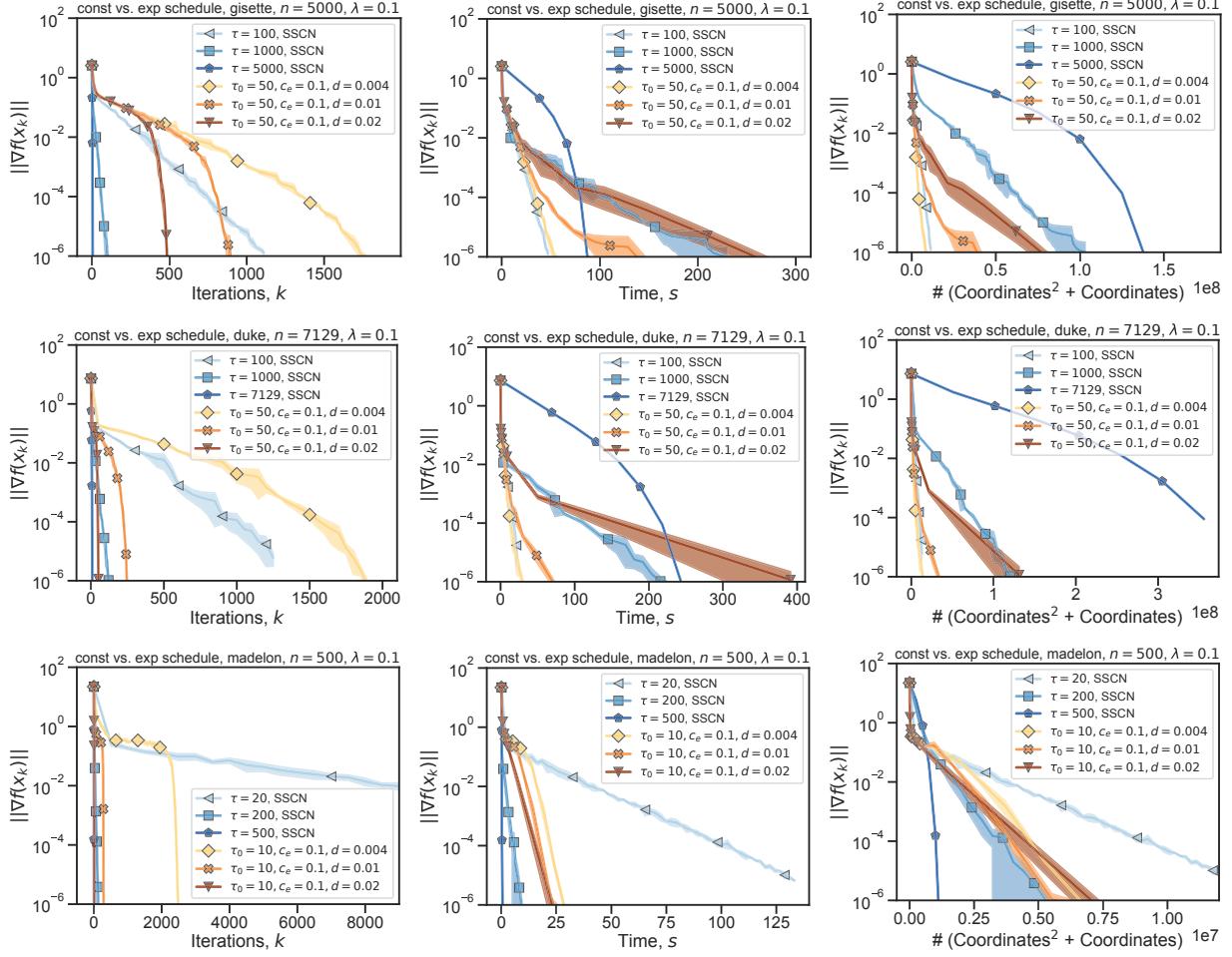


Figure 5: Comparison of constant vs. exponential schedules  $\tau(S_k) = \tau_0 + c_e \exp(dk)$  for different parameters w.r.t. iterations (first column) and time (second column) and  $\# (\text{Coordinates}^2 + \text{Coordinates})$  evaluated (third column) averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for the *gisette*, *duke* and *madelon* dataset.

## E.3 Comparison to other methods for more datasets

We also compared SSCN to CD and RS-RNM [Fuji et al., 2022] on logistic regression with non-convex regularization on two other datasets, namely the *duke* dataset and the *realsim* dataset. The results can be found in Figure 6. We observe that CD converges significantly faster in time to a first-order stationary point for the *duke* dataset and also slightly faster on the *realsim* dataset, while RS-RNM is significantly slower due to the high per-iteration cost. We trace back this differences in performance to the complexity of the loss landscape. In a separate run, we calculated the condition number of the Hessian of the loss function over the full space as an estimate for complexity of the landscape. The results can be found in Figure 7. We note that the loss is extremely ill-conditioned on the *madelon* dataset, which explains while CD fails to converge while SSCN is not affected. In better conditioned

problems such as *duke* or *realsim*, the higher per-iteration cost of SSCN leads to an overall slower convergence. In general, we expect more gains from using SSCN over CD on more complex loss landscape.

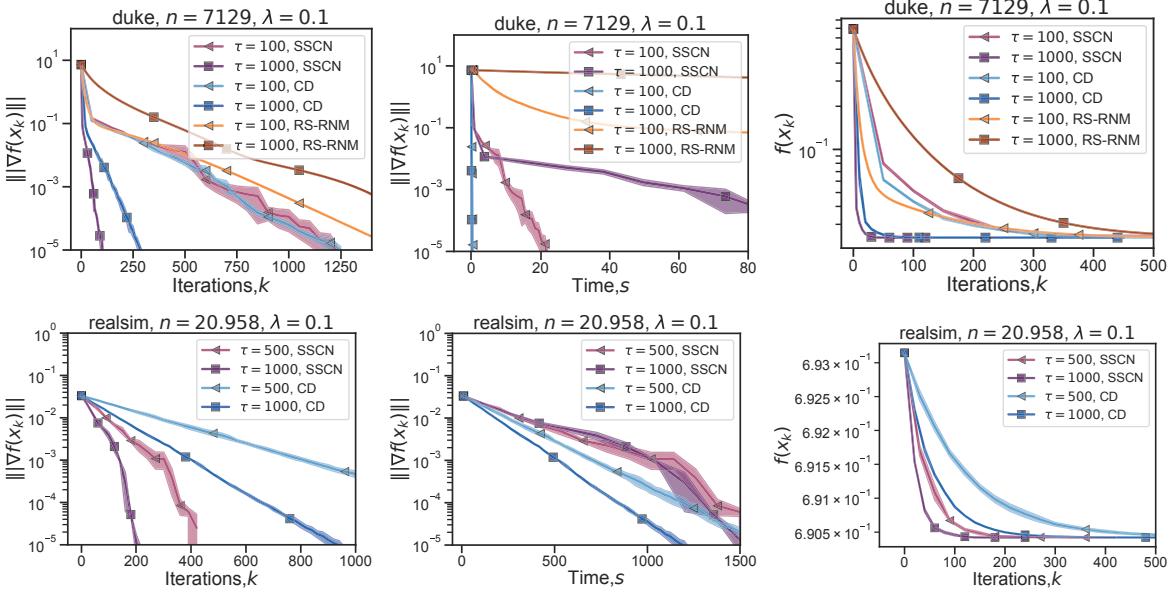


Figure 6: Comparison of CD, SSCN and RS-RNM [Fuji et al., 2022] for different constant coordinate schedules measured the gradient norm w.r.t. iterations (first column) and time (second column) and the loss w.r.t iterations (third column) averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for the *duke* dataset (first row) and *realsim* dataset (second row). Note that RS-RNM exceeded the compute budget of 3000 seconds before converging and is thus not shown.

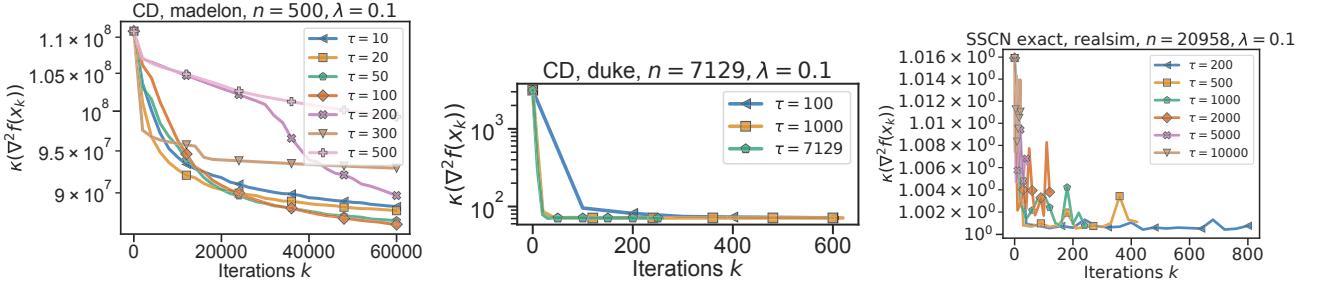


Figure 7: Condition number of full space Hessian of the loss function for the *madelon* (left), *duke* (center) and *realsim* (right) dataset throughout optimization. Note that the condition number on *madelon* is 5 orders of magnitudes larger than the condition number for *duke* and 8 orders larger than the the condition number for the *realsim* dataset.

#### E.4 Convergence to an $\epsilon$ -ball

We validate our prediction from Theorem 6 which guarantees the convergence to a ball whose radius is determined by  $\epsilon_1$  and  $\epsilon_2$ , which in turn depends on the number of sampled coordinates  $\tau(S)$ . The larger  $\tau(S)$ , the smaller the radius of the ball, as stated in Lemma C.1 and Lemma C.2. In Figure 8 we can see that in the setting of binary logistic regression with non-convex regularizer  $\lambda \cdot \sum_{i=1}^n \mathbf{x}_i^2 / (1 + \mathbf{x}_i^2)$  indeed the gradient norm to which each constant coordinate schedule converges to decreases with increasing number of sampled coordinates.

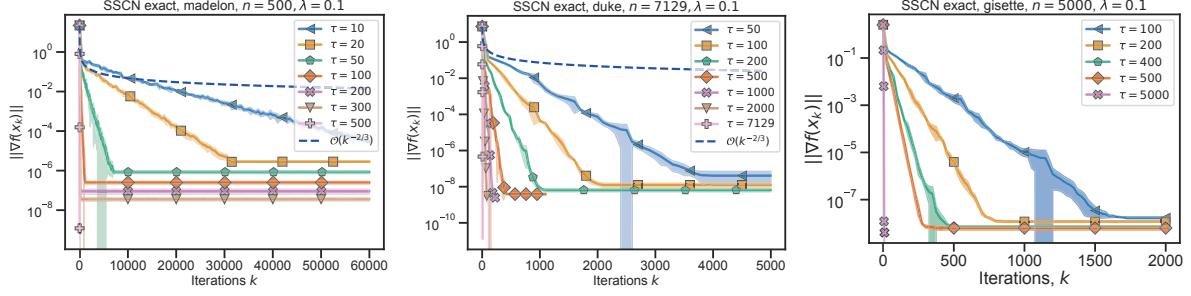


Figure 8: Convergence of different constant coordinate schedules measured w.r.t. iterations averaged over three runs for logistic regression with non-convex regularization with  $\lambda = 0.1$  for three different dataset. Plots correspond to the same setting as Figure 3 where the limits of the y-axis are chosen to be larger.

### E.5 Adaptive schedule on *gisette* dataset

We also verify the proposed adaptive schedule from Eq. (17) on the *gisette* dataset, where we replaced the full gradient norm and Hessian Frobenius norm by estimates  $\|\nabla f(\mathbf{x}_k)_{\text{est}}\|$  and  $\|\nabla^2 f(\mathbf{x}_k)_{\text{est}}\|_2$ , which are estimated as exponential moving averages:

$$\|\nabla f(\mathbf{x}_{k+1})_{\text{est}}\| = \alpha \|\nabla f(\mathbf{x}_{k+1})_{[S]}\| + (1 - \alpha) \|\nabla f(\mathbf{x}_k)_{\text{est}}\| \quad (77)$$

$$\|\nabla^2 f(\mathbf{x}_{k+1})_{\text{est}}\| = \alpha \|\nabla^2 f(\mathbf{x}_{k+1})_{[S]}\| + (1 - \alpha) \|\nabla^2 f(\mathbf{x}_k)_{\text{est}}\|, \quad (78)$$

where the weighting factor was chosen as  $\alpha = 0.2$ . The proposed schedule  $\tau(S_k)_{\text{prop}}$  is further smoothed through an exponential moving average  $\tau(S_{k+1}) = \beta \cdot \tau(S_{k+1})_{\text{prop}} + (1 - \beta) \tau(S_k)$ . As we can see the schedule is indeed close to an exponential schedule.

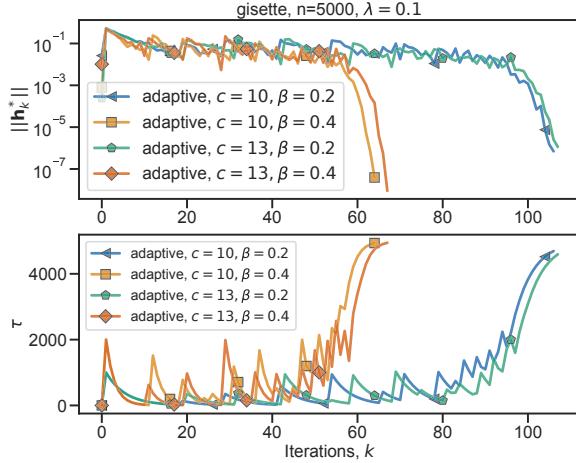


Figure 9: Evolution of  $\|\mathbf{h}_k^*\|$  and  $\tau(S_k)$  for the *gisette* dataset for the adaptive coordinate schedule  $\tau(S_k)$ .

### E.6 Squared norm of the step $\mathbf{h}_k$ for *duke* dataset

We also provide the plot of the squared norm of the step  $\mathbf{h}_k$  for the *duke* dataset in Figure 10 below.

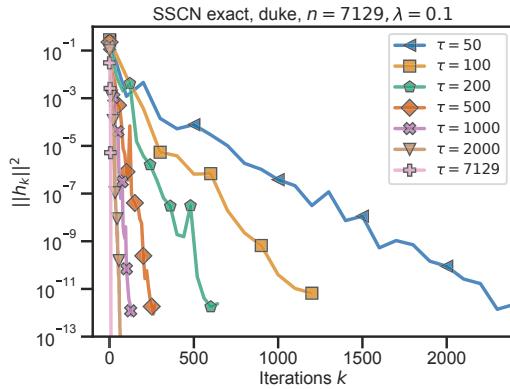


Figure 10: Squared norm of the step  $\mathbf{h}_k$  for different constant coordinate schedules for logistic regression with non-convex regularization with  $\lambda = 0.1$  for the *duke* dataset.