First- and Second-Order High Probability Complexity Bounds for Trust-Region Methods with Noisy Oracles

L. Cao∗§ A. S. Berahas† K. Scheinberg‡

January 31, 2023

Abstract

In this paper, we present convergence guarantees for a modified trust-region method designed for minimizing objective functions whose value and gradient and Hessian estimates are computed with noise. These estimates are produced by generic stochastic oracles, which are not assumed to be unbiased or consistent. We introduce these oracles and show that they are more general and have more relaxed assumptions than the stochastic oracles used in prior literature on stochastic trust-region methods. The method utilizes a relaxed step acceptance criterion and a cautious trust-region radius updating strategy which allows us to derive exponentially decaying tail bounds on the iteration complexity for convergence to points that satisfy approximate first- and second-order optimality conditions. Finally, we present two sets of numerical results. We first explore the tightness of our theoretical results on an example with adversarial zeroth- and first-order oracles. We then investigate the performance of the modified trust-region algorithm on standard noisy derivative-free optimization problems.

1 Introduction

The trust-region (TR) methods form a well-established class of iterative numerical methods for optimizing nonlinear continuous functions. In each iteration, TR methods minimize an approximation model of the objective function, often a quadratic model, within a trust-region. The book [11] contains an exhaustive coverage of these methods up to the time of its publication, and [23] offers a more recent survey. In this article, we consider the behavior of TR methods on unconstrained continuous optimization problems

$$\min_{x \in \mathbb{R}^n} \phi(x),$$

(1.1)

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is (twice) differentiable with Lipschitz continuous derivatives, but neither the objective function nor its associated derivatives are assumed to be computable accurately.

Under the condition that the function $\phi$ and its derivatives can be evaluated accurately, the complexity analysis of TR methods (for solving problem (1.1)) is well-established [11]. However, this condition is not always satisfied in the real-world problems. For example, in derivative-free optimization (DFO), the function $\phi$ is often a mapping between the input and the output of a computer program, such as a simulation of the

∗Beijing International Center for Mathematical Research, Peking University, Beijing, China; caoliyuan@bicmr.pku.edu.cn
†Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA; albertberahas@gmail.com
‡School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA; katyas@cornell.edu
§Corresponding author
physical world or the training of a machine learning model. As a result, there can be noise in the evaluation of $\phi$ due to limited numerical precision or randomness. Furthermore, the derivatives of $\phi$ cannot be computed directly and, when needed, can only be estimated using zeroth-order information. Another example is empirical risk minimization (ERM), where the objective function is the average of many functions, i.e., $\phi(x) = \frac{1}{N} \sum_{i=1}^{N} l(x, d_i)$. In settings where the number of functions $N$ is large (or even infinite, in which case the average becomes an expectation), it is typical to use the average of a subset of the functions, i.e., $\phi_B(x) = \frac{1}{|B|} \sum_{d_i \in B} l(x, d_i)$, where $B \subseteq \{d_1, \ldots, d_N\}$, instead of the true objective function $\phi$ to reduce the computational effort, at the cost of introducing errors in the evaluations.

When dealing with such problems, algorithms utilize various, usually stochastic, approximations of the objective function and derivative information ($\phi(x)$, $\nabla \phi(x)$ and $\nabla^2 \phi(x)$) in lieu of their exact counterparts. These approximations vary in terms of quality and reliability, and a variety of algorithms, not just TR methods, have been proposed and analyzed under different assumptions on the approximations employed. In order to give an overview of existing works and to clearly describe the contributions of this paper, we find it convenient to first propose and define general oracles that compute approximations of these approximations vary in terms of quality and reliability, and a variety of algorithms, not just TR methods, have been proposed and analyzed under different assumptions on the approximations employed. In order to give an overview of existing works and to clearly describe the contributions of this paper, we find it convenient to first propose and define general oracles that compute approximations of $\phi(x)$, $\nabla \phi(x)$ and $\nabla^2 \phi(x)$, and then discuss the different assumptions on these oracles made in prior literature as compared to those made in this paper. Definition 1.1 presents the general form of the stochastic oracles considered.

**Definition 1.1 (Stochastic jth-order oracle over a set $S_j$).** We say that the jth-order oracle $\varphi_j$ is implementable over a set $S_j \subseteq [0, +\infty) \times [0, 1]$ if it is capable of producing an estimate of $\nabla^j \phi$ that satisfies

$$\mathbb{P}_{\xi^{(j)}} \left\{ \|\varphi_j(x, \xi^{(j)}) - \nabla^j \phi(x)\| \leq A_j \right\} \geq p_j$$

(1.2)

for each $(A_j, p_j) \in S_j$ and any $x \in \mathbb{R}^n$, where $\xi^{(j)}$ is a random variable defined on some probability space whose distribution depends on $(A_j, p_j)$ and $x$, and $\mathbb{P}_{\xi^{(j)}}$ denotes probability with respect to that distribution.

Here $\nabla^j \phi(x)$ denotes the j-th derivative of $\phi(x)$, where $\nabla^0 \phi(x)$ reduces to $\phi(x)$. Clearly, if $(A_j, p_j) \in S_j$ for some $(A_j, p_j)$, then $[A_j, +\infty) \times [0, p_j] \subseteq S_j$. The cost of implementing the oracles depends on the application and will not be considered in this paper. In most applications, the actual cost depends on $(A_j, p_j)$ and monotonically decreases as $A_j$ increases and $p_j$ decreases. We use DFO and ERM as examples to discuss how to implement oracles satisfying (1.2) for a given $(A_j, p_j)$ in Section 2. That being said, the focus of this paper is on the analysis of TR methods under weak assumptions on the oracles in terms of the sets $S_j$. The results will thus apply whenever an oracle is implementable over the corresponding sets.

There are a variety of TR and line search algorithms in the literature that rely on stochastic oracles of this general form, although they are typically posed in a different way, specialized for each paper. For example, [2, 13] analyze the convergence of TR methods under the assumption that the zeroth-order oracle gives exact function values, and, at each iteration $k$, the first-order oracle can produce, with sufficiently high probability, a gradient estimate whose error is no more than the TR radius ($\delta_k$) multiplied by a constant. Since $\delta_k$ is not guaranteed to be bounded away from zero, these works effectively assume that the first-order oracle can produce gradient estimates with arbitrarily high precision, albeit only with a sufficiently high probability. Using our definition of stochastic oracles, [2, 13] assume their zeroth-order oracles are implementable over the (largest possible) set $S_0$ that contains $(0, 1)$, and their first-order oracles are implementable over $(0, \infty) \times [0, \tilde{p}_1]$, where $\tilde{p}_1$ is sufficiently large (e.g., $\tilde{p}_1 > \frac{1}{2}$ in [2]). In contrast, [6, 10] analyze a first-order TR method under the assumption that both zeroth- and first-order oracles are implementable over $(0, \infty) \times [0, \tilde{p}_j], j = 0, 1$, for sufficiently large $\tilde{p}_0$ and $\tilde{p}_1$. In [6], a second-order TR method is also analyzed under the assumption that $S_2 = (0, \infty) \times [0, \tilde{p}_2], j = 0, 1, 2$, for sufficiently large $\tilde{p}_0$, $\tilde{p}_1$, $\tilde{p}_2$, with an additional bound on $\mathbb{E}_{\tilde{S}_0}(\|\varphi_0(x, \xi_0) - \phi(x)\|)$, which requires a larger set $\tilde{S}_0$. Finally, we mention a recent technical report [21] where a first-order TR method with a relaxed step acceptance criterion is analyzed for oracles with deterministically bounded noise, i.e., $S_0 = [\epsilon_f, +\infty) \times [0, 1]$ and $S_1 = [\epsilon_g, +\infty) \times [0, 1]$ for some positive constants $\epsilon_f$ and $\epsilon_g$, which represent the irreducible upper bounds on the noise in the oracles.
Examples of "line search"-like methods based on stochastic oracles include the following. The authors in [9] analyze a stochastic line search method in the same oracle setting as in [2, 13]. In [18], a modified stochastic line search method is proposed and analyzed in a setting similar to that of [6]. A line search method with relaxed Armijo condition is analyzed in [3] under deterministic oracle assumptions, i.e., the sets $S_0$ and $S_1$ contain pairs $(\epsilon_f, 1)$ and $(\epsilon_g, 1)$, respectively, for some positive constants $\epsilon_f$ and $\epsilon_g$. In [5], the analysis of the line search method with relaxed Armijo condition is extended to less restrictive oracle settings where $S_0 = [\epsilon_f, +\infty) \times [0, 1]$ and $S_1 = [0, +\infty) \times [0, \bar{p}_1]$ for some positive $\epsilon_f$ and sufficiently large $\bar{p}_1$.

In a recent paper [14], the same line search as in [3, 5] was analyzed under weaker oracle conditions. In particular, $S_1 = [\epsilon_g, +\infty) \times [0, \bar{p}_1]$, where $\epsilon_g$ (some positive constant) is the best possible accuracy that the first-order oracle can guarantee and $\bar{p}_1$ is some constant larger than $1/2$. The zeroth-order oracle is a bit more complex: specifically, it is assumed to be implementable over $S_0 = \{(A_0, p_0) : A_0 \geq \epsilon_f \text{ and } p_0 \leq 1 - \exp(a(\epsilon_f - A_0))\}^2$, for some $a > 0$ and $\epsilon_f \geq 0$. This means that the errors in the function value estimates may not be bounded by $\epsilon_f$, but the distribution of the estimates is such that the probability of this error being larger than $\epsilon_f$, while positive, decays exponentially. A more extensive discussion and comparison of different oracles is presented in Section 2.

While most of the prior papers provide the analysis of expected iteration complexity of the corresponding algorithms, in [14], as well as [13], high probability (with exponentially decaying tail) bounds on the iteration complexity are derived. In this paper we draw inspiration from [14] and propose a TR method with a relaxed step acceptance criterion, and provide convergence guarantees under similar oracle conditions. In summary, our improvement upon the latest literature is as follows:

- In comparison to [6], we allow irreducible noise in the zeroth-, first-, and second-order oracles, so our analysis applies to problems where evaluation noise cannot be reduced below certain level. Our zeroth-order oracle is both more relaxed because it allows irreducible noise and yet somewhat stronger because it assumes a light-tailed distribution of the noise. This allows us to derive high probability complexity bounds for both first- and second-order versions of the algorithm as compared to bounds in expectation.

- In comparison to [13], we use more relaxed first- and second-order oracles, by allowing irreducible noise, and also a significantly more relaxed zeroth-order oracle, as it is assumed to be exact in [13].

- In comparison to [14], which analyzed a first-order step search method, we analyze first- and second-order TR methods. The difference between the complexity analyses for the two types of algorithms is significant, particularly in the presence of irreducible noise. New analytical techniques are employed to derive our results, e.g., the step size threshold $\bar{a}$ in [14] is a constant whereas the TR radius threshold $\Delta$ in this paper is a random variable. Second order analysis presented here is the first such analysis for irreducible noise.

In terms of iteration complexity, we obtain similar bounds to those in [6, 13, 14]. It is important to note that in this paper, as in many others that we discuss above, we analyze iteration complexity under specific assumptions on the oracles, but without directly accounting for the oracle costs. All the algorithms mentioned above are designed to update oracle accuracy (and thus their costs) adaptively, which makes the algorithms more practical, but harder to analyze in terms of the total oracle cost (or work).

**Organization** The paper is organized as follows. In Section 2, we introduce the assumptions and oracles, as well as motivate the oracles with examples from the literature. In Section 3, we introduce our modified TR algorithms and present some preliminary technical results and describe the stochastic process used to

---

1These methods change the random search direction even when reducing step size, thus, should not be strictly called “line search” methods. In [14], the term “step search” was introduced to distinguish the two classes of methods.

2This is a simplified version of the actual zeroth-order oracle in [14].
analyze the algorithms. The high probability tail bounds on the iteration complexity of the first- and second-order algorithms are presented in Sections 4 and 5, respectively. In Section 6, we present synthetic numerical experiments that simulate the worst-case behavior allowed under our oracle assumptions to support our theoretical findings. Finally, we test a practical TR algorithm with our proposed modification numerically in Section 7. We summarize the article in Section 8.

2 Assumptions and oracles

We consider the unconstrained optimization problem (1.1) with the following assumptions on \( \phi \). Let \( \langle \cdot, \cdot \rangle \) denote the sum of entry-wise products and \( \| \cdot \| \) the 2-norm.

**Assumption 2.1 (Lipschitz-smoothness).** The function \( \phi \) is continuously differentiable, and the gradient of \( \phi \) is \( L_1 \)-Lipschitz continuous on \( \mathbb{R}^n \), i.e., \( \| \nabla \phi(y) - \nabla \phi(x) \| \leq L_1 \| y - x \| \) for all \( (y, x) \in \mathbb{R}^n \times \mathbb{R}^n \).

**Assumption 2.2 (Lipschitz continuous Hessian).** The function \( \phi \) is twice continuously differentiable, and the Hessian of \( \phi \) is \( L_2 \)-Lipschitz continuous on \( \mathbb{R}^n \), i.e., \( \| \nabla^2 \phi(y) - \nabla^2 \phi(x) \| \leq L_2 \| y - x \| \) for all \( (y, x) \in \mathbb{R}^n \times \mathbb{R}^n \).

**Assumption 2.3 (Lower bound on \( \phi \)).** The function \( \phi \) is bounded below by a scalar \( \bar{\phi} \) on \( \mathbb{R}^n \).

Our algorithms utilize approximations of \( \phi, \nabla \phi \), and \( \nabla^2 \phi \) obtained via stochastic oracles. We assume our zeroth-, first- and second-oracles have the following properties in terms of accuracy and reliability.

**Oracle 0 (Stochastic zeroth-order oracle).** Given a point \( x \in \mathbb{R}^n \), the oracle computes \( f(x, \xi^{(0)}) \), a (random) estimate of the function value \( \phi(x) \), where \( \xi^{(0)} \) is a random variable whose distribution may depend on \( x \). Let \( e(x, \xi^{(0)}) = f(x, \xi^{(0)}) - \phi(x) \). For any \( x \in \mathbb{R}^n \), \( e(x, \xi^{(0)}) \) satisfies at least one of the two conditions:

1. **(Deterministically bounded noise: zeroth-order oracle implementable over \( S_0 = [\epsilon_f, \infty) \times [0, 1] \))** There is a constant \( \epsilon_f \geq 0 \) such that \( |e(x, \xi^{(0)})| \leq \epsilon_f \) for all realizations of \( \xi^{(0)} \).

2. **(Independent subexponential noise: zeroth-order stochastic oracle implementable over \( S_0 = \{ (A_0, p_0) : A_0 \geq \epsilon_f \text{ and } p_0 \leq 1 - \exp(a(\epsilon_f - A_0)) \} \))** There are constants \( \epsilon_f \geq 0 \) and \( a > 0 \) such that \( \mathbb{P}(e(x, \xi^{(0)}) > t) \leq \exp(a(\epsilon_f - t)) \) for all \( t \geq 0 \).

**Remark 2.4.** To be clear, we consider two oracle cases: one case is that of bounded noise, when Oracle 0.1 is implemented for all \( x \in \mathbb{R}^n \). This case includes deterministic or even adversarial noise, as long as it is bounded by \( \epsilon_f \). Otherwise, Oracle 0.2 is assumed to hold for all \( x \in \mathbb{R}^n \), which essentially states that the cumulative distribution of the noise has a subexponential tail whose rate of decay is governed by \( a \), but does not impose any restrictions on the noise distribution on the interval \([-\epsilon_f, \epsilon_f]\) as the right-hand side \( \exp(a(\epsilon_f - t)) \geq 1 \) when \( t \leq \epsilon_f \). The constants \( \epsilon_f \) and \( a \) are considered to be intrinsic to the oracle. The input to the oracle is the point \( x \).

**Oracle 1 (Stochastic first-order oracle implementable over \( S_1 = (\epsilon_g, \infty) \times [0, p_1] \)).** Given \( \delta^{(1)} > 0 \), a probability \( p_1 \in [0.5, 1] \), and a point \( x \in \mathbb{R}^n \), the oracle computes \( g(x, \xi^{(1)}) \), a (random) estimate of the gradient \( \nabla \phi(x) \) that satisfies

\[
\mathbb{P}(\xi^{(1)}) \left\{ \| g(x, \xi^{(1)}) - \nabla \phi(x) \| \leq \epsilon_g + \kappa_{eg}^{\delta^{(1)}} \right\} \geq p_1, \tag{2.1}
\]

where \( \xi^{(1)} \) is a random variable (whose distribution may depend on the input \( x \) and \( \delta^{(1)} \)).
Oracle 2 (Stochastic second-order oracle implementable over $S_2 = [\epsilon_H, \infty) \times [0, p_2]$). Given $\delta^{(2)} > 0$, a probability $p_2 \in [0.5, 1]$ and a point $x \in \mathbb{R}^n$, the oracle computes $H(x, \xi^{(2)})$, a (random) estimate of the Hessian $\nabla^2 \phi(x)$, such that

$$
\mathbb{P}_{\xi^{(2)}} \left\{ \|H(x, \xi^{(2)}) - \nabla^2 \phi(x)\| \leq \epsilon_H + \kappa_{\text{eh}} \delta^{(2)} \right\} \geq p_2,
$$

where $\xi^{(2)}$ is a random variable (whose distribution may depend on the input $x$ and $\delta^{(2)}$).

**Remark 2.5.** The inputs to the first- and second-order oracles are the triplets $(\delta^{(j)}, p_j, x)$, $j = 1, 2$. The constants $\epsilon_g$, $\epsilon_H$, $\kappa_{\text{eg}}$ and $\kappa_{\text{eh}}$ are nonnegative and are intrinsic to the oracles$^3$. Note that $\epsilon_g$ and $\epsilon_H$ limit the achievable accuracy, thus allowing the oracle to have error up to $\epsilon_g$ and $\epsilon_H$, respectively, with probability up to 1. The positive values $\delta^{(1)}$ and $\delta^{(2)}$ will be chosen dynamically by the algorithm, according to the TR radius. The probabilities $p_1$ and $p_2$ will be chosen to be constant and will need to satisfy certain bounds which will be derived in the theoretical analyses of our algorithms.

These oracle definitions are special cases of the oracle defined in the introduction (Definition 1.1). Henceforth, we will use $f(x, \xi^{(0)})$, $g(x, \xi^{(1)})$ and $H(x, \xi^{(2)})$ to denote the outputs of the oracles (instead of $\varphi_j(x, \xi^{(j)})$, $j = 0, 1, 2$, in Definition 1.1). The expressions $g(x, \xi^{(1)})$ and $H(x, \xi^{(2)})$ will be further abbreviated to $g(x)$ and $H(x)$ in the rest of Section 2, and their realizations will be denoted by $g_k$ and $H_k$ once they are put in the context of algorithms (Section 3 and beyond), where $k$ is the iteration index. Similarly, $f(x, \xi^{(0)})$ will be abbreviated as $f(x)$. The realizations of $f(x_k, \xi^{(0)})$ and $e(x_k, \xi^{(0)})$ will be denoted by $f_k$ and $e_k$, respectively.

### 2.1 Stochastic oracles used in prior literature

We now discuss different stochastic oracles used in prior literature for unconstrained optimization and show how our stochastic oracle definition (Definition 1.1) relates to them by comparing their respective sets $S_j$, $j = 0, 1, 2$ to those used by Oracles 0, 1 and 2. A visual comparison of various such sets considered in the literature is provided in Figure 1. Each subfigure of Figure 1 shows a different set $S_j$ on the $A_j$-$p_j$ plane.

There is an important difference between the sets in Figure 1b and Figure 1c. When a stochastic oracle is implementable over the first set (Figure 1b), it means that it is possible to evaluate the function or its derivative exactly, with probability at least $\tilde{p}_j$. In contrast, when a stochastic oracle is implementable over the second set (Figure 1c), it is assumed that the upper bound on the error (that holds with probability $\tilde{p}_j$) can be made arbitrarily small but never zero. An oracle of this second type appears naturally when, for example, $\phi$ is an expectation over a distribution from which one can obtain arbitrarily large number of samples. Figures 1e and 1f depict the two conditions assumed for Oracle 0, the zeroth-order oracle used in this paper. The conditions assumed for Oracle 1 and 2 are both depicted in Figure 1d, with $\epsilon_1 = \epsilon_g + \kappa_{\text{eg}} \delta^{(1)}$ and $\epsilon_2 = \epsilon_H + \kappa_{\text{eh}} \delta^{(2)}$, respectively.

**Probabilistic Taylor-like Conditions.** Corresponding to a norm $\| \cdot \|$, let $B(x, \delta)$ denote a ball of radius $\delta$ centered at $x \in \mathbb{R}^n$. If the gradient and Hessian estimates, $g(x)$ and $H(x)$, respectively, satisfy

$$
\|g(x) - \nabla \phi(x)\| \leq \kappa_{\text{eg}} \delta \quad \text{and} \quad \|H(x)\| \leq \kappa_{\text{hbm}}
$$

for some nonnegative scalars $\kappa_{\text{eg}}$ and $\kappa_{\text{hbm}}$\(^4\), then the model $m : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$
m(y) = \phi(x) + \langle g(x), y - x \rangle + \frac{1}{2} \langle H(x)(y - x), y - x \rangle
$$

$^3$“eg” and “eh” stand for “error in the gradient” and “error in the Hessian”, respectively.

$^4$“hbm” stands for “bound on the Hessian of the model”.


gives an approximation of \( f(y) \) within \( B(x, \delta) \) that is comparable to that given by an accurate first-order Taylor series approximation (with error dependent on \( \delta \)). Such models, introduced in [12], are known as \textit{fully linear} models on \( B(x, \delta) \). Similarly, if

\[
\|g(x) - \nabla \phi(x)\| \leq \kappa_{eg} \delta^2 \quad \text{and} \quad \|H(x) - \nabla^2 \phi(x)\| \leq \kappa_{eh} \delta,
\]

for some nonnegative scalars \((\kappa_{eg}, \kappa_{eh})\), then \( m(y) \) gives an approximation of \( f(y) \) that is comparable to that given by an accurate second-order Taylor series approximation. Such models are known as \textit{fully quadratic} models on \( B(x, \delta) \).

The concept of \textit{\( p \)-probabilistically fully-linear} and \textit{fully-quadratic models} was introduced in [2] and requires conditions (2.3) or (2.4) to hold with probability \( p \) at each iteration of a trust region algorithm, conditioned on the past. Thus, a \( p \)-probabilistically fully-linear model can be constructed given access to Oracles 1 with \( \epsilon_g = 0 \) and input \( \delta_1 = \delta \) and \( p_1 = p \), and a \( p \)-probabilistically fully-quadratic models can be built given access to Oracle 1 with \( \epsilon_0 = 0 \) and input \( \delta^{(1)} = \delta^2 \) and Oracle 2 with \( \epsilon_H = 0 \) and input \( \delta^{(2)} = \delta \) and \( p_1 p_2 \geq p \).

In [2, 13], the convergence and complexity of a trust region method were analyzed under the assumption that Oracles 1 and 2 are implementable over \( \mathcal{S}_j = (0, \infty) \times [0, \bar{p}_j] \), where \( \bar{p}_j \) is sufficiently large, \( j = 1, 2 \) (Figure 1c). On the other hand, the function oracle in [2, 13] was assumed to be exact and not stochastic. In other words, Oracle 0.1 with \( \epsilon_f = 0 \) (Figure 1a).

In [6, 10], the conditions on the zeroth-order oracle were significantly relaxed by assuming (in the first-order analysis) that

\[
P \{ |f(x) - \phi(x)| \leq \kappa_{ef} \delta^2 \} \geq \bar{p}_0
\]

for a sufficiently large \( \bar{p}_0 \), where \( \kappa_{ef} \) is some positive constant. That is, \( \mathcal{S}_0 = (0, \infty) \times [0, \bar{p}_0] \), for some sufficiently large \( \bar{p}_0 \) (see Figure 1c). For the second-order analysis, the assumptions are stronger, requiring

\[
P \{ |f(x) - \phi(x)| \leq \kappa_{ef} \delta^3 \} \leq \bar{p}_0
\]

to hold for some sufficiently large \( \bar{p}_0 \) and

\[
E[|f(x) - \phi(x)|] \leq \kappa_F \delta^3
\]

for some \( \kappa_F \). If \( \delta \) is not bounded below by a positive number, the second condition implies a zeroth-order oracle implementable over \( (0, +\infty) \times [0, 1] \). However, if \( \delta \) is bounded below by a positive number, then
these two conditions imply weaker conditions on \( S_0 \) than our assumptions on Oracle 0, allowing heavy tailed distributions of the error \(|f(x) - \phi(x)|\). Establishing algorithm complexity under the oracles in [6, 10] requires a different type of analysis than the one presented in this paper, which so far has not been extended to deriving a high probability complexity bound and has resulted in worse bounds on \( \bar{p}_1 \) and \( \bar{p}_2 \) as well as worse constants in the complexity bound.

**Gradient Norm Condition.** The fully-linear model condition is strongly tied to the TR algorithm by the use of the TR radius on the right-hand side of (2.3) and (2.4). If, instead of the TR method, a line search method based on inexact gradient estimates \( \{g(x_k)\}_{k=0,1,...} \) is used for obtaining a solution \( x_* \) with \( \|\nabla \phi(x_*)\| \leq \epsilon \) for some \( \epsilon > 0 \), then to establish the computational complexity, it is sufficient that the gradient estimate \( g(x) \) satisfies the gradient norm condition

\[
\|g(x) - \nabla \phi(x)\| \leq \theta \|\nabla \phi(x)\|,
\]

at all the iterates \( x \in \{x_k\}_{k=0,1,...} \), for some \( \theta \in [0, 1) \) with sufficiently high probability. To satisfy this condition, the first order oracle needs to be implementable over \([\epsilon, +\infty) \times [0, \bar{p}_1]\) for a sufficiently large \( \bar{p}_1 \).

The norm condition (2.5) was first introduced in [8] in the context of TR methods. This condition, with sufficiently small \( \theta \) ensures the convergence of popular methods such as Armijo backtracking line-search. Verifying the gradient norm condition (2.5) requires knowledge of \( \|\nabla \phi(x_k)\| \), thus it cannot be enforced, even with some fixed probability, in the case of expected risk minimization [7]. However, in some settings, where \( g(x) \) is a randomized finite difference approximation of the gradient of a (possibly noisy) function [17], it is possible to ensure (2.5) holds with sufficiently high probability [4].

**Stochastic Gradient Norm Conditions.** When the norm condition cannot be ensured, one again can resort to some Taylor-like conditions. In the case of line search, however, those have to hold in a region, whose size is a product of the step size parameter \( \alpha \) and the norm of the search direction \( \|g(x)\| \). Thus, the following conditions inspired by the concept of fully-linear models have been used in the literature to ensure convergence of line search methods [9, 18],

\[
|f(x) - \phi(x)| \leq \kappa_{ef} \alpha^2 \|g(x)\|^2 \quad \text{and} \quad \|g(x) - \nabla \phi(x)\| \leq \kappa_{eg} \alpha \|g(x)\|,
\]

for some nonnegative constants \( \kappa_{ef}, \kappa_{eg} \), where \( \alpha \) is a step size parameter. When (2.6) holds, the linear model \( \phi(x) + \langle g(x), y-x \rangle \) gives an approximation of \( \phi \) around \( y = x - \alpha g(x) \) that is comparable to that given by the first-order Taylor expansion of \( \phi \) in \( B(x, \alpha ||g(x)||) \). In [9, 18], complexity analyses of randomized and stochastic line search algorithms were derived under the condition that (2.6) holds with sufficiently high probability. A possible procedure of ensuring this conditions was outlined in [9] and in essence requires access to a stochastic first-order oracle implementable over \( S_1 = (0, \infty) \times [0, \bar{p}_1] \), where \( \bar{p}_1 \) is sufficiently large. The zeroth-order oracle in [9], as in [2, 13], is exact. In [18], as in [6], the zeroth-order oracle is assumed to be implementable over \((0, \infty) \times [0, \bar{p}_0] \subset S_0 \) (Figure 1c).

Finally, an almost identical Oracle 0 as the one in this paper is proposed in [14], while the first-order oracle is similar to our Oracle 1 but with \( \|g(x)\| \) appearing the bound on the accuracy, as in other papers on line search methods. There is no second-order analysis in [14], thus no second-order oracle is defined.

In summary, our first- and second-order oracles are more general than those used in prior literature. Our assumptions on the zeroth-order oracle, while not general enough to include those in [6, 10], are relevant to practice and yet allow for strong theoretical results. In the next two settings, we discuss two common settings for the stochastic zeroth-, first-, and second-order oracles.
2.2 Expected risk minimization

In this setting, \( \phi(x) = \mathbb{E}_{d \sim D}[l(x, d)] \), where \( x \) are the model parameters, \( d \) is a data sample following distribution \( D \), and \( l(x, d) : \mathbb{R}^n \to \mathbb{R} \) is the loss function of the \( d \)-th data point parametrized by \( x \). The zeroth- and first-order oracles estimates are obtained by sample averages of the loss function and its gradient, respectively, over \( B \) (a mini-batch sampled from \( D \)), i.e.,

\[
    f_B(x) = \frac{1}{|B|} \sum_{d \in B} l(x, d), \quad g_B(x) = \frac{1}{|B|} \sum_{d \in B} \nabla_x l(x, d).
\]  

(2.7)

In [14] it is shown that the conditions of Oracle 0.2 are satisfied for any \( x \) for which \( l(x, d) \) has a subexponential distribution (e.g., when the support of \( D \) is bounded and \( l \) is Lipschitz) by selecting an appropriate sample size \( |B| \).

Let us show how Oracle 1 is easily implemented in this setting and explain the roles of \( \epsilon_g \) and \( \kappa_{eg} \). Assume that the variance of stochastic gradient is bounded, \( \mathbb{E}_{d \sim D} \| \nabla l(x, d) - \nabla \phi(x) \| ^2 \leq \sigma^2 \). Given input \( \delta(1) \) and \( p_1 \), choose random i.i.d. mini-batch \( B \) whose size is at least \( \min \{N, (1 - p_1)\delta(1) \} \), where \( N \) is the maximum possible mini-batch size that the oracle can generate. Then, we have

\[
    \mathbb{E}_B \| g_B(x) - \nabla \phi(x) \| \leq \sqrt{\mathbb{E}_B \| g_B(x) - \nabla \phi(x) \|^2} \leq \max \left\{ \frac{\sigma}{\sqrt{N}}, \sigma(1 - p_1)\delta(1) \right\},
\]

which by Markov inequality implies

\[
    \mathbb{P} \left\{ \| g_B(x) - \nabla \phi(x) \| \leq \frac{\sigma}{(1 - p_1)\sqrt{N}} + \sigma\delta(1) \right\} \geq \mathbb{P} \left\{ \| g_B(x) - \nabla \phi(x) \| \leq \max \left\{ \frac{\sigma}{(1 - p_1)\sqrt{N}}, \sigma\delta(1) \right\} \right\} \geq p_1.
\]

(2.8)

Thus, we have Oracle 1 with input \( \delta(1) \) and \( p_1 \), with \( \epsilon_g = \frac{\sigma}{(1 - p_1)\sqrt{N}} \) and \( \kappa_{eg} = \sigma \). We note that \( \epsilon_g \) and \( \kappa_{eg} \) need not be known for the execution of Algorithm 1, but Algorithm 2 requires an estimate of \( \epsilon_g \).

2.3 Gradient and Hessian approximation via zeroth-order oracle

Let us consider the setting in which only the zeroth-order oracle is available for the objective function. This zeroth-order oracle can come from a variety of settings, such as simulation-based optimization, machine learning, or solutions for complex systems [12]. Both cases, Oracle 0.1 and Oracle 0.2, have numerous applications. Let us now discuss how Oracle 1 and Oracle 2 can be implemented using only function estimates via finite differences.

Consider the following first-order oracle: given \( x \in \mathbb{R}^n \), choose \( \sigma > 0 \) and compute \( f(y) \) for all \( y \) in the set \( \mathcal{Y} = \{x\} \cup \{x + \sigma u_i\}_{i=1}^n \) using Oracle 0, where \( u_i, i = 1, \ldots, n \), denotes the unit vector along the \( i \)-th coordinate. Compute \( g(x) \) as follows

\[
    g(x) = \sum_{i=1}^n \frac{f(x + \sigma u_i) - f(x)}{\sigma} u_i.
\]

(2.9)

The following proposition holds.
Proposition 2.6 ([4] Theorem 2.1). Assume that \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\). Then, under Assumption 2.1
\[
\|g(x) - \nabla \phi(x)\| \leq \frac{\sqrt{nL_1\sigma}}{2} + \frac{\sqrt{n\hat{\epsilon}_f}}{\sigma}.
\] (2.10)

Now consider the following second-order oracle: given \(x \in \mathbb{R}^n\), choose \(\sigma > 0\) and compute \(f(y)\) for all \(y\) in the set \(\mathcal{Y} = \{x\} \cup \{x + \sigma u_1\}_{i=1}^n \cup \{x + \sigma u_i + \sigma u_j\}_{i,j=1}^n\) using Oracle 0. Compute \(H(x)\) as follows
\[
H(x) = \sum_{i=1}^n \sum_{j=1}^n \frac{f(x + \sigma u_i + \sigma u_j) - f(x + \sigma u_i) - f(x + \sigma u_j) + f(x)}{\sigma^2} u_i u_j^\top,
\] (2.11)
The following proposition holds.

Proposition 2.7. Assume that \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\). Then, under Assumption 2.2
\[
\|H(x) - \nabla^2 \phi(x)\| \leq \frac{(\sqrt{2} + 1)nL_2\sigma}{3} + \frac{4n\hat{\epsilon}_f}{\sigma^2}.
\] (2.12)

First let us assume that Oracle 0.1 is used. Then, Propositions 2.6 and 2.7 apply with \(\hat{\epsilon}_f = \epsilon_f\) with probability 1. Hence, by selecting \(\sigma\) that minimizes the right-hand sides of (2.10), the finite difference formula (2.9) gives us Oracle 1 with \(\epsilon_g = \frac{nL_1\epsilon_f}{2}, \kappa_{eg} = 0\) for any \(\delta^{(1)} \geq 0\) and for \(p_1 = 1\). Similarly, by choosing \(\sigma\) to minimize the right-hand side of (2.12), we obtain from formula (2.11) an Oracle 2 with \(\epsilon_H = \frac{\sqrt{3}(\sqrt{2} - 1)^2\epsilon_f L_2^2}{2}\), \(\kappa_{eh} = 0\) for any \(\delta^{(2)} \geq 0\) and for \(p_2 = 1\).

Next, let us consider the case of Oracle 0.2. It is not guaranteed that \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\). However, for any \(\epsilon_f > \hat{\epsilon}_f\), we have that for any \(y\), \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) with probability at least \(1 - e^{\alpha(\epsilon_f - \hat{\epsilon}_f)}\). Thus, with probability at least \((1 - e^{\alpha(\epsilon_f - \hat{\epsilon}_f)})^{n+1}\) it holds that \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\) defined in the first-order oracle. Proposition 2.6 implies that the first-order oracle defined above delivers Oracle 1 with \(\epsilon_g = \frac{nL_1\epsilon_f}{2}, \kappa_{eg} = 0\) for \(\delta^{(1)}\) and for \(p_1 = (1 - e^{\alpha(\epsilon_f - \hat{\epsilon}_f)})^{n+1}\). Similarly, \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\), defined by the second-order oracle, with probability at least \((1 - e^{\alpha(\epsilon_f - \hat{\epsilon}_f)})^{(n+1)(n+2)/2}\), and thus we have Oracle 2 with \(\epsilon_H = \frac{\sqrt{3}(\sqrt{2} - 1)^2\hat{\epsilon}_f L_2^2}{2}, \kappa_{eh} = 0\) for any \(\delta^{(2)}\) and for \(p_2 = (1 - e^{\alpha(\epsilon_f - \hat{\epsilon}_f)})^{(n+1)(n+2)/2}\).

Now, let us consider a first-order oracle based on polynomial interpolation. Specifically, for a given \(x\) choose \(\sigma > 0\) and a linearly independent set of vectors \(\{u_i\}_{i=1}^n\), such that \(|u_i| \leq 1\). Compute \(f(y)\) for all \(y\) in the set \(\mathcal{Y} = \{x\} \cup \{x + \sigma u_i\}_{i=1}^n\) using Oracle 0. Let \(U \in \mathbb{R}^{n \times n}\) denote matrix whose rows are \(\{u_i\}_{i=1}^n\) and let \(\tilde{u}_i\) denote the columns of \(U^{-1}\). Compute
\[
g(x) = \sum_{i=1}^n \frac{f(x + \sigma u_i) - f(x)}{\sigma} \tilde{u}_i.
\]

It is shown in [4] that if \(|f(y) - \phi(y)| \leq \hat{\epsilon}_f\) for all \(y \in \mathcal{Y}\) and under Assumption 2.1 the following bound holds.
\[
\|g(x) - \nabla \phi(x)\| \leq \|U^{-1}\| \left[ \sqrt{nL_1\sigma} \frac{2}{2} + \frac{2\sqrt{n\epsilon_f}}{\sigma} \right],
\]
Using arguments similar to those used above for finite differences, one can easily show that the interpolation oracle provides Oracle 1 with appropriately chosen \(\epsilon_g\) and \(\kappa_{eg} = 0\) and for any \(\delta^{(1)} \geq 0\). The additional nuance of this case is the choice of \(U\), so that \(\|U^{-1}\|\) is bounded from above either deterministically or probabilistically, depending on an algorithm employed, thus \(\hat{\epsilon}^{(1)}\) of this Oracle 1 is defined according to the choice of \(U\), the instance of Oracle 0 employed.
Similarly, Oracle 2 can be implemented using techniques such as quadratic interpolation by choosing an appropriate sample set. How to choose such a sample set involves the convoluted concept of poisedness which is out of the scope of this paper. We refer interested readers to [12, 19, 20]. In [2] a stochastic second-order oracle is generated by using quadratic interpolation a randomly sampled set, which allows for a more efficient second-order oracle, when $\nabla \phi(x)$ is approximately sparse. A further discussion of stochastic oracles in [2] and other settings that fit our generic oracle definition is a worthwhile topic, but is beyond the scope of this paper.

3 Trust-region algorithms for noisy optimization

In this section, we propose first- and second-order modified TR algorithms which utilize the stochastic oracles discussed in Section 2 to produce models of the objective function. We also define the requirements on these models and derive some key properties of both algorithms under these requirements. We finish the section by describing the algorithms as stochastic processes which we then analyze in subsequent sections.

3.1 Algorithms

In every iteration $k \in \{0, 1, \ldots \}$ of our modified first- and second-order TR algorithms, a quadratic model

$$m_k(x_k + s) = \phi(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle H_k s, s \rangle$$

(3.1)

is constructed to approximate the objective function near the iterate $x_k$. The constant term $\phi(x_k)$ appears in (3.1) for clarity, but is not needed in the algorithm, since only changes in the model value $m_k(x_k) - m_k(x_k + s)$ are computed. The model gradient $g_k = \nabla m_k(x_k)$ is computed as a (random) approximation of $\nabla \phi(x_k)$ by the first-order oracle (Oracle 1) with a specified accuracy and reliability for the iterate $x_k$. The model Hessian $H_k = \nabla^2 m_k(x_k)$ can be a quasi-Newton matrix or other (not necessarily random or accurate) approximation of $\nabla^2 \phi(x_k)$, that satisfies the following standard assumption [11].

Assumption 3.1. For all $k = 0, 1, 2, \ldots$, $\|H_k\| \leq \kappa_{bhm}$, where $\kappa_{bhm}$ is a positive constant.

Our modified first-order TR method is stated in Algorithm 1. In the execution of Algorithm 1 (Line 2) it is required that the TR subproblem, defined as

$$\min_{s \in B(x_k, \delta_k)} m_k(x_k + s),$$

(3.2)

where $B(x_k, \delta_k)$ is a Euclidean ball with center $x_k$ and radius $\delta_k > 0$, is consistently solved accurately enough so that the step $s_k$ satisfies

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{fed}}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\},$$

(3.3)

for some (chosen) constant $\kappa_{fed} \in (0, 1]$.\footnote{“fed” stands for “fraction of Cauchy decrease”.} Condition (3.3) is commonly used in the literature and is satisfied by the Cauchy point with $\kappa_{fed} = 1$. See [11, Section 6.3.2] for more details.
Algorithm 1: Modified First-Order Trust-Region Algorithm

**Inputs:** starting point \(x_0\); initial TR radius \(\delta_0 > 0\); hyperparameters for controlling the TR radius \(\eta_1 > 0, \eta_2 > 0, \gamma \in (0, 1)\); tolerance parameter \(r > 0\); and probability parameter \(p_1\).

for \(k = 0, 1, 2, \ldots \) do

1. Compute vector \(g_k\) using stochastic Oracle 1 with input \((\delta_k, p_1, x_k)\) and matrix \(H_k\) that satisfies Assumption 3.1.
2. Compute \(s_k\) by approximately minimizing \(m_k\) in \(B(x_k, \delta_k)\) so that it satisfies (3.3).
3. Compute \(f_k\) using Oracle 0, with input \(x_k\), and \(f_k^+\) using Oracle 0, with input \(x_k + s_k\), and then compute
   \[
   \rho_k = \frac{f_k - f_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)}.
   \]
4. if \(\rho_k \geq \eta_1\) then
   \[
   \delta_{k+1} = \begin{cases} 
   \gamma^{-1} \delta_k & \text{if } \|g_k\| \geq \eta_2 \delta_k \\
   \gamma \delta_k & \text{if } \|g_k\| < \eta_2 \delta_k.
   \end{cases}
   \]
5. else
   \[
   \text{Set } x_{k+1} = x_k \text{ and } \delta_{k+1} = \gamma \delta_k.
   \]

Algorithm 2 is our modified second-order TR algorithm. Similar to Algorithm 1, in the execution of Algorithm 2 (Line 2) the TR subproblem (3.2) needs to be solved sufficiently accurately, and the step \(s_k\) computed needs to satisfy the following stronger condition

\[
\frac{m_k(x_k) - m_k(x_k + s_k)}{\|s_k\|^2} \geq \frac{\kappa_{\text{fod}}}{2} \max \left\{ \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|^2}, \delta_k \right\}, -\lambda_{\min}(H_k) \delta_k^2 \right\}, \quad (3.4)
\]

for some (chosen) constant \(\kappa_{\text{fod}} \in (0, 1]^6\). Contrary to Algorithm 1, the Hessian approximations \(H_k\) in Algorithm 2 are required to be sufficiently accurate and not just bounded in norm. Furthermore, instead of comparing \(\|g_k\|\) to \(\eta_2 \delta_k\) to determine the adjustment of the TR radius, the following value is used:

\[
\beta_k^{\alpha} \overset{\text{def}}{=} \max \{\|g_k\|, -\lambda_{\min}(H_k)\}. \quad (3.5)
\]

We also note that if by any chance \(m_k(x_k) = m_k(x_k + s_k)\) in any iteration of either of these two algorithms, the step \(s_k\) is automatically rejected, i.e., \(x_{k+1} = x_k\) and \(\delta_{k+1} = \gamma \delta_k\).

**Remark 3.2.** Algorithms 1 and 2 are very similar to classical TR algorithms [11]. The major difference pertains to the fact that the step acceptance criterion is relaxed. The relaxation is similar to that in [3, 5, 14] for line/step search methods. A user-defined tolerance parameter is added to the numerator in order to account for the noise in the zeroth-order oracle. The value of \(r\) in Algorithm 1 is set to \(2 \epsilon_f\) if \(\epsilon_f\) is known (for example when the zeroth-order oracle satisfies Oracle 0.1 with a known noise bound); otherwise, we simply let \(r\) be any value large enough to be no less than \(2 \epsilon_f\). Similarly, the value of \(r\) in Algorithm 2 is set to \(2 \epsilon_f + \epsilon_g^{3/2}\) if both \(\epsilon_f\) and \(\epsilon_g\) are known, and set to any large enough value otherwise. The effect of choosing particular values of \(r\) will be discussed later in the paper.

### 3.2 Approximation model accuracy

Let us introduce a definition of a sufficiently accurate model.

\[\text{“fod” stands for “fraction of decrease”}\]

\[\text{11}\]
Algorithm 2: Modified Second-Order Trust-Region Algorithm

**Inputs:** starting point $x_0$; initial TR radius $\delta_0 > 0$; hyperparameters for controlling the TR radius $\eta_1 > 0, \eta_2 > 0, \gamma \in (0, 1)$; tolerance parameter $r > 0$; and probability parameters $p_1$ and $p_2$.

for $k = 0, 1, 2, \ldots$ do

1. Compute vector $g_k$ using stochastic Oracle 1 with input $(\delta_k^2, p_1, x_k)$ and matrix $H_k$ using stochastic Oracle 2 with input $(\delta_k, p_2, x_k)$.

2. Compute $s_k$ by approximately minimizing $m_k$ in $B(x_k, \delta_k)$ so that it satisfies (3.4).

3. Compute $f_k$ using Oracle 0, with input $x_k$, and $f_k^+$ using Oracle 0, with input $x_k + s_k$, and then compute
   \[ \rho_k = \frac{f_k - f_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)} . \]

4. if $\rho_k \geq \eta_1$ then
   Set $x_{k+1} = x_k + s_k$ and using $\beta^{m_k}$ defined in (3.5)
   \[ \delta_{k+1} = \begin{cases} \gamma^{-1} \delta_k & \text{if } \beta^{m_k} \geq \eta_2 \delta_k \\ \gamma \delta_k & \text{if } \beta^{m_k} < \eta_2 \delta_k. \end{cases} \]

5. else
   Set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma \delta_k$.

**Definition 3.3.** An approximation model of the form (3.1) is said to be **first-order sufficiently accurate** if there are nonnegative constants $\kappa_{eg}$ and $\epsilon_g$ such that
   \[ \|\nabla \phi(x_k) - g_k\| \leq \kappa_{eg} \delta_k + \epsilon_g, \] (3.6)
   and, is said to be **second-order sufficiently accurate** if there are nonnegative constants $\kappa_{eg}, \kappa_{eh}, \epsilon_g$ and $\epsilon_H$ such that
   \[ \|\nabla^2 \phi(x_k) - H_k\| \leq \kappa_{eh} \delta_k + \epsilon_H \] (3.7a)
   \[ \|\nabla \phi(x_k) - g_k\| \leq \kappa_{eg} \delta_k^2 + \epsilon_g. \] (3.7b)

Note that (3.6) is satisfied with probability $p_1$ by Oracle 1 with input $(\delta_k, p_1, x_k)$, (3.7b) is satisfied with probability $p_2$ by Oracle 1 with input $(\delta_k^2, p_1, x_k)$ and (3.7a) is satisfied with probability $p_2$ by Oracle 2 with input $(\delta_k, p_2, x_k)$.

Under (3.6) (resp., (3.7)), error bounds on the model accuracy can be derived. The first lemma below provides a bound on the approximation error of the model in $B(x_k, \delta_k)$ under (3.6).

**Lemma 3.4.** Under Assumptions 2.1 and 3.1, if (3.6) holds, it follows
   \[ |\phi(x_k + s) - m_k(x_k + s)| \leq (L_1 + \kappa_{lhm} + 2\kappa_{eg}) \delta_k^2 / 2 + \epsilon_g \delta_k \] (3.8)
   for all $x_k + s \in B(x_k, \delta_k)$.
Proof. By the triangle inequality, Assumptions 2.1 and 3.1 and (3.6), it follows that,

\[
|\phi(x_k + s) - m_k(x_k + s)| \\
= |\phi(x_k + s) - \phi(x_k) - (g_k, s) - \langle H_k s, s \rangle / 2| \\
\leq |\phi(x_k + s) - \phi(x_k) - (\nabla \phi(x_k), s)| + |(\nabla \phi(x_k), s) - (g_k, s)| + |(H_k s, s) / 2| \\
\leq L_1 \|s\|^2 / 2 + \|\nabla \phi(x_k) - g_k\| \|s\| + \kappa_{bhm} \|s\|^2 / 2 \\
\leq L_1 \delta_k^2 / 2 + (\kappa_{eg} \delta_k + \epsilon_g) \delta_k + \kappa_{bhm} \delta_k^2 / 2 \\
= (L_1 + \kappa_{bhm} + 2 \kappa_{eg}) \delta_k^2 / 2 + \epsilon_g \delta_k.
\]

\[\square\]

The next result provides a bound on the approximation error of the model in \(B(x_k, \delta_k)\) under (3.7).

**Lemma 3.5.** Under Assumption 2.2, if (3.7) holds, it follows

\[
|\phi(x_k + s) - m_k(x_k + s)| \leq (L_2/6 + \kappa_{eg} + 1 + \kappa_{eb}/2) \delta_k^2 \|s\| + \epsilon_H \delta_k \|s\|^2 / 2 + \epsilon_g^{3/2} \tag{3.9}
\]

for all \(x_k + s \in B(x_k, \delta_k)\).

**Proof.** First let us show that if (3.7b) holds, then

\[
\|\nabla \phi(x_k) - g_k\| \leq (\kappa_{eg} + 1) \delta_k^2 + \min \left\{ \epsilon_g^{3/2} / \delta_k, \epsilon_g \right\}. \tag{3.10}
\]

Clearly if \(\epsilon_g^{3/2} / \delta_k \geq \epsilon_g\), then (3.7b) trivially implies (3.10). On the other hand, if \(\epsilon_g^{3/2} / \delta_k \leq \epsilon_g\), then \(\delta_k \geq \epsilon_g^{1/2}\) and thus \(\kappa_{eg} \delta_k^2 + \epsilon_g \leq (\kappa_{eg} + 1) \delta_k^2\), hence again (3.7b) implies (3.10).

Thus, by the triangle inequality, Assumption 2.2 and (3.7), it follows that,

\[
|\phi(x_k + s) - m_k(x_k + s)| \\
= |\phi(x_k + s) - \phi(x_k) - (g_k, s) - 0.5 (H_k s, s)| \\
\leq |\phi(x_k + s) - \phi(x_k) - (\nabla \phi(x_k), s)| + |(\nabla \phi(x_k), s) - (g_k, s)| + |(H_k s, s) / 2| \\
\leq L_2 \|s\|^2 / 6 + \|\nabla \phi(x_k) - g_k\| \|s\| + \|\nabla^2 \phi(x_k) - H_k \| \|s\|^2 / 2 \\
\leq L_2 \|s\|^3 / 6 + ((\kappa_{eg} + 1) \delta_k^2 + \epsilon_g^{3/2} / \delta_k) \|s\| + (\kappa_{eb} \delta_k + \epsilon_H) \|s\|^2 / 2 \\
\leq (L_2/6 + \kappa_{eg} + 1 + \kappa_{eb}/2) \delta_k^2 \|s\| + \epsilon_H \delta_k \|s\|^2 / 2 + \epsilon_g^{3/2}.
\]

\[\square\]

For the second order analysis we will find the following simple and technical assumption useful. It plays a similar role as [6, Assumption 6b]. Such an assumption can be avoided, but at the expense of a significant increase in the complexity of the analysis (cf. the first version of this paper).

**Assumption 3.6 (Upper bound on convergence accuracy).** The optimization problem is properly scaled so that the desired accuracy \(\epsilon \leq 1\) and the irreducible noises \(\epsilon_g, \epsilon_H \ll 1\).
3.3 Algorithms viewed as stochastic processes

For the purposes of analyzing the convergence of Algorithms 1 and 2, we view the algorithms as stochastic processes. Here we introduce and explain some useful notation.

Let \( \{X_k\}, \{X_k^+\} \) denote the sequences of random vectors in \( \mathbb{R}^n \) whose realizations are \( \{x_k\} \) and \( \{x_k + s_k\} \), respectively. Let \( \{\Delta_k\} \) denote the sequence of random positive numbers whose realizations are \( \{\delta_k\} \), and \( \{M_k\} \) denote the sequence of random models whose realizations are \( \{m_k\} \). For the error in the zeroth-order oracle, we denote by \( \{\mathcal{E}_k\} \) and \( \{\mathcal{E}^+_k\} \) the absolute errors whose realizations are \( \{|f_k - \phi(x_k)|\} \) and \( \{|f_k^+ - \phi(x_k + s_k)|\} \), respectively. Additionally, with a slight abuse of notation, let \( \{\rho_k\} \) be the random variables that share the same symbols as their realizations.

With these random variables, Algorithms 1 or 2 first generate in each iteration (random) gradient and Hessian approximations, \( g(X_k, \xi^{(1)}_k) \) and \( H(X_k, \xi^{(2)}_k) \), using Oracles 1 and 2, respectively. Subsequently, a random model \( M_k \) is constructed deterministically around the current iterate \( X_k \) using these approximations and then minimized within the trust-region to generate \( X_k^+ \) (deterministically, given \( X_k \) and \( M_k \)). Then, random function value estimates \( f(X_k, \xi^{(0)}_k) \) and \( f(X_k^+, \xi^{(0+)}_k) \) are generated using Oracle 0, which define the random errors \( \mathcal{E}_k = |e(X_k, \xi^{(0)}_k)| \) and \( \mathcal{E}^+_k = |e(X_k^+, \xi^{(0+)}_k)| \). And finally, \( \rho_k, \Delta_k \) and \( X_k, \xi^{(1)}_k \) are computed in a deterministic manner (given \( X_k, M_k \) and the function values). Thus, given \( X_k \) and \( \Delta_k \), the randomness at the \( k \)-th iteration is generated by the variables \( \xi_k, \xi_k^+, \xi_k^{(1)} \) and \( \xi_k^+ \).

Let \( \mathcal{F}_{k-1} \) denote the \( \sigma \)-algebra

\[
\mathcal{F}_{k-1} = \sigma\left(\left(\xi^{(0)}_0, \xi^{(0+)}_0, \xi^{(1)}_0, \xi^{(2)}_0\right), \ldots, \left(\xi^{(0)}_{k-1}, \xi^{(0+)}_{k-1}, \xi^{(1)}_{k-1}, \xi^{(2)}_{k-1}\right)\right).
\]  

Similarly, let

\[
\mathcal{F}'_{k-1} = \sigma\left(\left(\xi^{(0)}_0, \xi^{(0+)}_0, \xi^{(1)}_0, \xi^{(2)}_0\right), \ldots, \left(\xi^{(0)}_{k-1}, \xi^{(0+)}_{k-1}, \xi^{(1)}_{k-1}, \xi^{(2)}_{k-1}\right), \left(\xi^{(1)}_k, \xi^{(2)}_k\right)\right)
\]  

We note that the random variables \( \{X_k\} \) and \( \Delta_k \) are defined by \( \mathcal{F}_{k-1} \), the random variables \( \{X_k^+\} \) and \( M_k \) are defined by \( \mathcal{F}'_{k-1} \), and the random variables \( \{\mathcal{E}_k\}, \{\mathcal{E}^+_k\} \) and \( \rho_k \) are defined by \( \mathcal{F}_k \).

4 First-order stochastic convergence analysis

In this section, we analyze the first-order convergence of Algorithm 1. The goal is to derive a probabilistic result of the form

\[
P\left\{\min_{0 \leq k \leq T-1} \|\nabla \phi(X_k)\| < \epsilon\right\} \geq 1 - \delta
\]

for any sufficiently large \( \epsilon \). This result cannot hold for arbitrarily small values of \( \epsilon > 0 \) unless \( \epsilon_f = \epsilon_g = 0 \). The specific lower bounds on \( \epsilon \) in terms of \( \epsilon_f \) and \( \epsilon_g \) will be presented in Theorems 4.11 and 4.18.

We begin by stating and proving three key lemmas about the behavior of Algorithm 1 when (3.6) and Assumptions 2.1 and 3.1 hold. Let \( \epsilon_k = f_k - \phi(x_k) \) and \( \epsilon_k^+ = f_k^+ - \phi(x_k + s_k) \). The first lemma provides a sufficient condition for accepting a step \( (x_{k+1} = x_k + s_k) \).
Lemma 4.1 (Sufficient condition for accepting step). Under Assumptions 2.1 and 3.1, if (3.6) holds, \( r \geq \epsilon_k^+ - \epsilon_k \), and
\[
\delta_k \leq \frac{(1 - \eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} \|g_k\| - \frac{2}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} \epsilon_g,
\] (4.1)
then, \( \rho_k \geq \eta_1 \) in Algorithm 1.

Proof. Since \( \delta_k \leq (1 - \eta_1)\kappa_{\text{fcd}} \|g_k\|/(L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}) \leq \|g_k\|/\kappa_{\text{bhm}} \leq \|g_k\|/\|H_k\| \), (3.3) suggests \( m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{fcd}} \|g_k\| \delta_k/2 \). Combining this inequality with \( \epsilon_k - \epsilon_k^+ + r \geq 0 \) and Lemma 3.4, we have
\[
\rho_k = \frac{\phi(x_k) + \epsilon_k - \phi(x_k + s_k) - \epsilon_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)} \geq \frac{\phi(x_k) - \phi(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq 1 - \frac{(L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}})\delta_k^2/2 - \epsilon_g \delta_k}{\kappa_{\text{fcd}} \|g_k\| \delta_k}
\] (3.4)
\[
\geq 1 - (1 - \eta_1) = \eta_1.
\]

The next lemma provides a sufficient condition for a successful step \( (x_{k+1} = x_k + s_k \) and \( \delta_{k+1} = \gamma^{-1} \delta_k \).

Lemma 4.2 (Sufficient condition for successful step). Under Assumptions 2.1 and 3.1, if (3.6) holds, \( r \geq \epsilon_k^+ - \epsilon_k \), and
\[
\delta_k \leq C_1 \|\nabla \phi(x_k)\| - C_2 \epsilon_g,
\] (4.2)
where
\[
C_1 \overset{\text{def}}{=} \min \left\{ \frac{(1 - \eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}} + \eta_2} \right\}
\] and
\[
C_2 \overset{\text{def}}{=} \max \left\{ \frac{(1 - \eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}}, \frac{1}{\kappa_{\text{eg}} + \eta_2} \right\}
\] (4.3)
then, \( \rho_k \geq \eta_1 \) and \( \|g_k\| \geq \eta_2 \delta_k \) in Algorithm 1.

Proof. By (3.6) we have
\[
\|g_k\| \geq \|\nabla \phi(x_k)\| - \|\nabla \phi(x_k) - g_k\| \geq \|\nabla \phi(x_k)\| - \kappa_{\text{eg}} \delta_k - \epsilon_g.
\]
Then
\[
\frac{(1 - \eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} \|g_k\| - \frac{2}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} \epsilon_g \geq \frac{(1 - \eta_1)\kappa_{\text{fcd}}}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} (\|\nabla \phi(x_k)\| - \kappa_{\text{eg}} \delta_k - \epsilon_g) - \frac{2}{L_1 + \kappa_{\text{bhm}} + 2\kappa_{\text{eg}}} \epsilon_g \overset{(4.2)}{\geq} \delta_k.
\]
The last inequality holds due to (4.2) with \( C_1 \) and \( C_2 \) set to the first term in their corresponding min/maximization operation. Then by Lemma 4.1, we have \( \rho_k \geq \eta_1 \). We also have
\[
\|g_k\| \geq \|\nabla \phi(x_k)\| - \kappa_{\text{eg}} \delta_k - \epsilon_g \overset{(4.2)}{\geq} \eta_2 \delta_k.
\]
The last lemma provides a lower bound on the progress made in each iteration.

**Lemma 4.3 (Progress made in each iteration).** In Algorithm 1, if $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2 \delta_k$, then

$$\phi(x_k) - \phi(x_{k+1}) \geq h(\delta_k) - e_k + \epsilon_k^+ - r,$$

where

$$h(\delta_k) = C_3 \delta_k^2 \quad \text{and} \quad C_3 \overset{\text{def}}{=} \frac{1}{2} \eta_1 \eta_2 \kappa_{\text{fed}} \min \left\{ \frac{\eta_2}{\kappa_{\text{bhm}}}, 1 \right\}.$$  \hfill (4.4)

If $\rho_k \geq \eta_1$ but $\|g_k\| < \eta_2 \delta_k$, then

$$\phi(x_k) - \phi(x_{k+1}) \geq -e_k + \epsilon_k^+ - r.$$  \hfill (4.5)

If $\rho_k < \eta_1$, then $\phi(x_{k+1}) = \phi(x_k)$.

**Proof.** Let $\rho_k \geq \eta_1$. We have

$$\eta_1 \leq \rho_k = \frac{\phi(x_k) + e_k - \phi(x_k + s_k) - \epsilon_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)},$$

which can be rearranged to

$$\eta_1 [m_k(x_k) - m_k(x_k + s_k)] \overset{(3.3)}{=} \frac{\eta_1 \kappa_{\text{fed}}}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\} \geq \frac{\eta_1 \kappa_{\text{fed}}}{2} \eta_2 \delta_k \min \left\{ \frac{\eta_2 \delta_k}{\kappa_{\text{bhm}}}, \delta_k \right\} = h(\delta_k);$$

otherwise

$$\eta_1 [m_k(x_k) - m_k(x_k + s_k)] \overset{(3.3)}{=} \frac{\eta_1 \kappa_{\text{fed}}}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\} \geq 0.$$  \hfill \Box

If $\rho_k < \eta_1$, we have $x_{k+1} = x_k$, so $\phi(x_{k+1}) = \phi(x_k)$.

Next, our analysis relies on categorizing iterations $k = 0, 1, \ldots, T - 1$ into different types, where $T$ is any positive integer. These types are defined using the following random indicator variables:

$I_k = 1\{ \|\nabla \phi(X_k) - \nabla M_k(X_k)\| \leq \kappa_{eg} \Delta_k + \epsilon_g \}$ (whether the model is first-order sufficiently accurate)

$J_k = 1\{ r \geq \epsilon_k^+ + \hat{\epsilon}_k \}$ (whether function evaluation errors are compensated by $r$)

$\Theta_k = 1\{ \rho_k \geq \eta_1 \}$ and $\|\nabla M_k(X_k)\| \geq \eta_2 \Delta_k$ (whether the step is successful)

$\Theta'_k = 1\{ \rho_k \geq \eta_1 \}$ (whether the step is accepted)

$\Lambda_k = 1\{ \Delta_k > \bar{\Delta} \}$, $\Lambda'_k = 1\{ \Delta_k \geq \bar{\Delta}' \}$,

where $\bar{\Delta}$ and $\bar{\Delta}'$ are defined as

$$\bar{\Delta} = C_1 \min_{0 \leq k \leq T - 1} \|\nabla \phi(X_k)\| - C_2 \epsilon_g,$$

$$\bar{\Delta}' = \min_l \{ \gamma^l \delta_0 : \gamma^l \delta_0 > \gamma \Delta \text{ and } l \in \mathbb{Z} \};$$

and the positive constants $C_1$ and $C_2$ are defined in (4.3).

Notice that under Oracle 0.1, the condition $r \geq 2 \epsilon_f \geq \epsilon_k^+ + \hat{\epsilon}_k$ always holds, thus $J_k = 1$. The random variable $\Delta$ is crucial to our analysis not only because it involves the value $\min_{0 \leq k \leq T - 1} \|\nabla \phi(X_k)\|$, but also because of the following corollary to Lemma 4.2, which shows that if the errors from Oracles 0 and 1 are small, and the TR radius is also small, then the iteration is successful.

**Corollary 4.4.** If $I_k J_k = 1$ and $\Lambda_k = 0$, then $\Theta_k = 1$.  

16
We also have the following lemma as a direct consequence of the definitions of Oracle 1 and I_k.

**Lemma 4.5.** The random sequence \{M_k\} satisfies the submartingale condition
\[
P\{I_k = 1 \mid F_{k-1}\} \geq p_1 \text{ for all } k \in \{0, 1, \ldots\},
\]
where \(F_{k-1}\) is defined in (3.11).

**Remark 4.6.** To be specific, the random process \(\{\sum_{k=0}^{t-1} I_k - p_1 t\}_{t=0,1,\ldots}\) is a submartingale.

Before we state and prove the main results of this section (Theorem 4.11 for Oracle 0.1 and Theorem 4.18 for Oracle 0.2), we state and prove three technical lemmas that are used in the analysis of Algorithm 1 under both Oracle 0.1 and Oracle 0.2. The first lemma provides an upper bound on the number of successful iterations with large (defined by \(\Lambda_k\)) TR radius.

**Lemma 4.7.** For any positive integer \(T\), we have
\[
h(\gamma \Delta) \sum_{k=0}^{T-1} \Theta_k \Lambda_k' < \phi(x_0) - \hat{\phi} + \sum_{k=0}^{T-1} \Theta_k' (\mathcal{E}_k + \mathcal{E}_k^+ + r).
\]

**Proof.** Notice \(h(\cdot)\) (defined in (4.4)) is a monotonically non-decreasing function, so \(h(\Delta_k) \geq h(\Delta')\) if \(\Lambda_k' = 1\). By Lemma 4.3,
\[
\phi(X_k) - \phi(X_{k+1}) \geq \begin{cases} h(\Delta') - \mathcal{E}_k - \mathcal{E}_k^+ + r & \text{if } \Theta_k \Lambda_k' = 1 \\ -\mathcal{E}_k - \mathcal{E}_k^+ + r & \text{if } \Theta_k' = 1 \\ 0 & \text{otherwise.} \end{cases}
\]
Thus,
\[
\phi(x_0) - \hat{\phi} \geq \phi(x_0) - \phi(X_T) = \sum_{k=0}^{T-1} \phi(X_k) - \phi(X_{k+1}) \geq \sum_{k=0}^{T-1} \Theta_k \Lambda_k' h(\Delta') - \sum_{k=0}^{T-1} \Theta_k' (\mathcal{E}_k + \mathcal{E}_k^+ + r).
\]
Since using \(\Delta'\) over-complicates later analysis, we derive a slightly weaker inequality in (4.8) by using the fact that \(h(\Delta') > h(\gamma \Delta)\).

The next lemma bounds the difference between the number of iterations where the TR radius moves towards \((\Theta_k(1 - \Lambda_k')\) and \((1 - \Theta_k)\Lambda_k)\) and away from \(((1 - \Theta_k)(1 - \Lambda_k)\) and \(\Theta_k \Lambda_k')\) the interval \([\Delta', \Delta]\).

**Lemma 4.8.** For any positive integer \(T\), we have
\[
\sum_{k=0}^{T-1} \Theta_k (1 - \Lambda_k') - (1 - \Theta_k)(1 - \Lambda_k) + (1 - \Theta_k)\Lambda_k - \Theta_k \Lambda_k' \leq \left| \log_{\gamma} \frac{\Delta'}{\delta_0} \right| < \left| \log_{\gamma} \frac{\Delta}{\delta_0} \right| + 1.
\]

**Proof.** Consider the sequence
\[
\zeta_k = \max \left\{ \log(\Delta_k/\Delta'), 0 \right\}.
\]
This non-negative value starts at \(\zeta_0 = \max \left\{ \log(\delta_0/\Delta'), 0 \right\}\) and increases by \(-\log \gamma\) in iteration \(k\) if \(\Theta_k \Lambda_k' = 1\) or decreases by \(-\log \gamma\) if \((1 - \Theta_k)\Lambda_k = 1\). Other types of iterations do not affect this value. Thus,
\[
\zeta_T = \zeta_0 + \sum_{k=0}^{T-1} -\Theta_k \Lambda_k' \log \gamma + (1 - \Theta_k)\Lambda_k \log \gamma \geq 0,
\]
17
from which it follows that,
\[ T - 1 \sum_{k=0}^{T-1} -\Theta_k'\Lambda_k' + (1 - \Theta_k)\Lambda_k \leq -\frac{\zeta_0}{\log \gamma} = \max \left\{ \log, \frac{\bar{\Delta}'\delta_0}{\bar{\Delta}'} \right\}. \]

Similarly, consider the sequence
\[ \zeta_k' = \max \left\{ \log(\frac{\bar{\Delta}/\Delta_k}{\bar{\Delta}'})/\Delta_k, 0 \right\}. \]
It decreases by \(-\log \gamma\) if \(\Theta_k(1 - \Lambda_k) = 1\) and increases by \(-\log \gamma\) if \((1 - \Theta_k)(1 - \Lambda_k) = 1\). Thus,
\[ \zeta_T' = \zeta_0' + T - 1 \sum_{k=0}^{T-1} -(1 - \Theta_k)(1 - \Lambda_k) \log \gamma + \Theta_k(1 - \Lambda_k') \log \gamma \geq 0 \]
from which it follows that,
\[ T - 1 \sum_{k=0}^{T-1} -(1 - \Theta_k)(1 - \Lambda_k) + \Theta_k(1 - \Lambda_k') \leq -\frac{\zeta_0'}{\log \gamma} = \max \left\{ \log, \frac{\delta_0}{\bar{\Delta}'}, 0 \right\}. \]

The first inequality in (4.9) follows by combining the above two results. The second inequality is trivially true. As in the previous lemma, we relax the right-hand side to a function of \(\bar{\Delta}\) instead of \(\bar{\Delta}'\) in order to simplify the subsequent analysis.

The last lemma uses the Azuma-Hoeffding inequality to establish a probabilistic lower bound on the number of iterations with sufficiently accurate models.

**Lemma 4.9.** For any positive integer \(T\) and any \(\hat{p}_1 \in [0, p_1]\), we have
\[ \mathbb{P} \left\{ \sum_{k=0}^{T-1} I_k > \hat{p}_1 T \right\} \geq 1 - \exp \left( -\frac{(1 - \hat{p}_1/p_1)^2}{2} T \right). \] \hspace{1cm} (4.10)

**Proof.** Consider the submartingale \(\left\{ \sum_{k=0}^{t-1} I_k - p_1 t \right\}_{t=0,1,...} \). Since
\[ \left| \left( \sum_{k=0}^{(t+1)-1} I_k - p_1(t+1) \right) - \left( \sum_{k=0}^{t-1} I_k - p_1 t \right) \right| = |I_t - p_1| \leq \max\{|0 - p_1|, |1 - p_1|\} = p_1 \]
for any \(t \in \mathbb{N}\), by the Azuma-Hoeffding inequality, we have for any positive integer \(T\) and any positive real \(\epsilon\)
\[ \mathbb{P} \left\{ \sum_{k=0}^{T-1} I_k - Tp_1 \leq -\epsilon \right\} \leq \exp \left( -\frac{\epsilon^2}{2Tp_1^2} \right). \]

Setting \(\epsilon = (p_1 - \hat{p}_1)T\) and subtracting 1 from both sides yields the result.

**4.1 Convergence Analysis: The Bounded Noise Case**

We present in this subsection our result on Algorithm 1 with Oracle 0.1. The following lemma combines the inequalities from Lemmas 4.7-4.9.
Lemma 4.10. For any positive integer $T$ and any $\hat{p}_1 \in [0, p_1]$, we have
\[
\mathbb{P} \left\{ \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma \Delta)} \right) T < \frac{\phi(x_0) - \hat{\phi} + \frac{1}{2} \left| \log_{\gamma} \Delta \right| + \frac{1}{2}}{h(\gamma \Delta)} \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{p}_1 / p_1)^2}{2} T \right). \quad (4.11)
\]

Proof. Multiply
\[
\frac{2\epsilon_f + r}{h(\gamma \Delta)} \left[ \sum_{k=0}^{T-1} \Theta_k (1 - \Lambda_k) + (1 - \Theta_k)(1 - \Lambda_k) + (1 - \Theta_k)\Lambda_k + \Theta_k \Lambda'_k \right] = \sum_{k=0}^{T-1} 1 = T
\]
by $(2\epsilon_f + r)/h(\gamma \Delta) - 0.5$ and (4.9) by 0.5, then add the results together to obtain
\[
\frac{2\epsilon_f + r}{h(\gamma \Delta)} \left[ \sum_{k=0}^{T-1} (1 - \Theta_k)(1 - \Lambda_k) + \Theta_k \Lambda'_k \right] < \left( \frac{2\epsilon_f + r}{h(\gamma \Delta)} - \frac{1}{2} \right) T + \frac{1}{2} \left| \log_{\gamma} \Delta \right| + \frac{1}{2} \quad (4.12)
\]
The probabilistic bound (4.11) can be derived as
\[
\mathbb{P} \left\{ \sum_{k=0}^{T-1} \Theta_k \Lambda'_k + \frac{2\epsilon_f + r}{h(\gamma \Delta)} \Theta_k' < \left( \frac{2\epsilon_f + r}{h(\gamma \Delta)} - \frac{1}{2} \right) T + \frac{1}{2} \left| \log_{\gamma} \Delta \right| + \frac{1}{2} \right\} \\
\geq \mathbb{P} \left\{ \left[ \sum_{k=0}^{T-1} (1 - \Theta_k)(1 - \Lambda_k) \right] + \frac{2\epsilon_f + r}{h(\gamma \Delta)} \left[ \sum_{k=0}^{T-1} 1 - \Theta_k' \right] < (1 - \hat{p}_1)T \text{ and } (4.12) \text{ holds.} \right\} \\
= \mathbb{P} \left\{ \left[ \sum_{k=0}^{T-1} (1 - \Theta_k)(1 - \Lambda_k) \right] + \frac{2\epsilon_f + r}{h(\gamma \Delta)} \left[ \sum_{k=0}^{T-1} 1 - \Theta_k' \right] < (1 - \hat{p}_1)T \right\} \\
\geq \mathbb{P} \left\{ \sum_{k=0}^{T-1} (1 - \Theta_k)(1 - \Lambda_k) + (1 - I_k)\Theta_k + (1 - I_k)(1 - \Theta_k)\Lambda_k < (1 - \hat{p}_1)T \right\} \\
= \mathbb{P} \left\{ \sum_{k=0}^{T-1} I_k(1 - \Theta_k)(1 - \Lambda_k) + (1 - I_k) < (1 - \hat{p}_1)T \right\} \\
= \mathbb{P} \left\{ \sum_{k=0}^{T-1} (1 - I_k) < (1 - \hat{p}_1)T \right\} = \mathbb{P} \left\{ \sum_{k=0}^{T-1} I_k > \hat{p}_1 T \right\} \\
\geq 1 - \exp \left( - \frac{(1 - \hat{p}_1 / p_1)^2}{2} T \right). \quad (4.10)
\]
The first inequality holds because the event in the first line is an inequality obtained by the sum of (4.12) with the first inequality in the second line. The second inequality holds because the left-hand side of the inequality in the fourth line is greater than or equal to the left-hand side of the inequality in the third line. The second last equality holds since $J_k = 1$ for all $k$ and then $\sum_{k=0}^{T-1} I_k(1 - \Theta_k)(1 - \Lambda_k) = 0$ due to Corollary 4.4.

Meanwhile, by Lemma 4.7 and Oracle 0.1, we have
\[
h(\gamma \Delta) \sum_{k=0}^{T-1} \Theta_k \Lambda'_k < \phi(x_0) - \hat{\phi} + (2\epsilon_f + r) \sum_{k=0}^{T-1} \Theta_k' \]
or equivalently
\[
- \frac{\phi(x_0) - \hat{\phi}}{h(\gamma \Delta)} \leq - \sum_{k=0}^{T-1} \Theta_k \Lambda'_k + \frac{2\epsilon_f + r}{h(\gamma \Delta)} \Theta_k'.
\]
Combining the above two results, it follows that

\[ P\{ A \} = P \left\{ \frac{\phi(x_0) - \hat{\phi}}{h(\gamma \Delta)} < \left( \frac{2\epsilon_f + r}{h(\gamma \Delta)} + \frac{1}{2} - \hat{p}_1 \right) T + \frac{1}{2} \log_\gamma \frac{\Delta}{\delta_0} + \frac{1}{2} \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{p}_1/p_1)^2}{2} T \right). \]

\[ \] \[ \]

Our main theorem follows the above lemma.

**Theorem 4.11.** Let Assumptions 2.1 and 2.3 hold for the objective function \( \phi \). Let Assumptions 3.1 and \( r \geq 2\epsilon_f \) hold for Algorithm 1. Give any \( \epsilon > 2 \epsilon_f + r \), where \( C_1, C_2 \) and \( C_3 \) are defined in (4.3) and (4.4), the sequence of iterates generated by Algorithm 1 with Oracle 0.1 satisfies

\[ P \left\{ \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \leq \epsilon \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{p}_1/p_1)^2}{2} T \right) \] \[ (4.13) \]

for any \( \hat{p}_1 \in \left( \frac{1}{2} + \frac{2\epsilon_f + r}{C_1 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2}, p_1 \right) \) and any

\[ T \geq \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_1 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi}}{C_3 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} + \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{C_1 \epsilon - C_2 \epsilon_g}{\delta_0}, \frac{\delta_0}{C_1 \| \nabla \phi(x_0) \| - C_2 \epsilon_g} \right\} \right) + \frac{1}{2} \right]. \] \[ (4.14) \]

**Proof.** Recall \( \Delta \) is defined as \( C_1 \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| - C_2 \epsilon_g \). If \( - \Delta \geq 0 \), then \( \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \leq C_2 \epsilon_g/C_1 \), and the result is trivially true. Now assume \( \Delta > 0 \). Consider the univariate function

\[ Q(y) = \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi}}{C_3 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} + \frac{1}{2} \log_\gamma \left( \frac{C_1 \epsilon - C_2 \epsilon_g}{\delta_0}, \frac{\delta_0}{C_1 \| \nabla \phi(x_0) \| - C_2 \epsilon_g} \right) + \frac{1}{2} \right]. \]

The probabilistic bound \( (4.11) \) we proved in Lemma 4.10 is

\[ P \left\{ T < Q \left( \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \right) \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{p}_1/p_1)^2}{2} T \right), \]

and \( (4.14) \) implies \( T \geq Q(\epsilon) \). To prove \( (4.13) \), we only need to show that \( T < Q(\min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \|) \) implies \( \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \leq \epsilon \). Suppose, for the sake of contradiction, that \( T < Q(\min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \|) \) but \( \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| > \epsilon \).

If \( \Delta \leq \delta_0 \), we have \( C_1 \epsilon - C_2 \epsilon_g \leq C_1 \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| - C_2 \epsilon_g = \Delta \leq \delta_0 \) and the function \( Q \) decreases monotonically between \( \epsilon \) and \( \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \) (recall \( 0 < \gamma < 1 \)). It follows that \( Q(\epsilon) > Q(\min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \|) \), which contradicts \( Q(\epsilon) \leq T < Q(\min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \|) \).

Alternatively, if \( \Delta > \delta_0 \), the condition \( T < Q(\min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \|) \) implies

\[ T \leq \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma \Delta)} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi}}{h(\gamma \Delta)} + \frac{1}{2} \log_\gamma \frac{\delta_0}{\Delta} + \frac{1}{2} \right]. \] \[ \]

\[ \leq \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma \Delta)} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi}}{h(\gamma \Delta)} + \frac{1}{2} \log_\gamma \frac{\delta_0}{C_1 \| \nabla \phi(x_0) \| - C_2 \epsilon_g} + \frac{1}{2} \right]. \]

\[ \leq \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi}}{C_3 \gamma (C_1 \epsilon - C_2 \epsilon_g)^2} + \frac{1}{2} \log_\gamma \frac{\delta_0}{C_1 \| \nabla \phi(x_0) \| - C_2 \epsilon_g} + \frac{1}{2} \right]. \]
where the last inequality holds because of the assumption $\min_{0 \leq k \leq T-1} \|\nabla \phi(X_k)\| > \epsilon$ and the fact that the function

$$Q'(y) = \left( \hat{p}_1 - \frac{1}{2} - \frac{2\epsilon_f + r}{C_3\gamma^2(C_1y - C_2\epsilon_g)^2} \right)^{-1} \left[ \phi(x_0) - \hat{\phi} \right] C_3\gamma^2(C_1y - C_2\epsilon_g)^2 + \frac{1}{2} \log \gamma \frac{\delta_0}{C_1\|\nabla \phi(x_0)\| - C_2\epsilon_g} + \frac{1}{2} \right]$$

decreases monotonically on $[\epsilon, +\infty)$. However, this contradicts (4.14). Thus, $T < Q(\min_{0 \leq k \leq T-1} \|\nabla \phi(X_k)\|)$ implies $\min_{0 \leq k \leq T-1} \|\nabla \phi(X_k)\| \leq \epsilon$.

**Remark 4.12.** The bound (4.14) is rather complicated. Notice it can also be written as

$$T \geq \frac{\phi(x_0) - \hat{\phi}}{(\hat{p}_1 - 0.5)C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2 - 2\epsilon_f - r} + \frac{1}{2} \log \gamma \left( \min \left\{ \frac{C_1\epsilon - C_2\epsilon_g}{\delta_0}, \frac{\delta_0}{C_1\|\nabla \phi(x_0)\| - C_2\epsilon_g} \right\} \right) + \frac{1}{2}.$$

The first term on the right-hand side is term that results in $O(\epsilon^{-2})$ complexity, which is typical (and optimal) for first-order optimization algorithms on smooth nonconvex functions. The second term represents the number of iterations the algorithm takes to adjust the TR radius to a desired level. If the initial radius $\delta_0$ is too big, then $\delta_0/(C_1\epsilon - C_2\epsilon_g)$ is larger and the second term represents the number of iterations needed for the TR to shrink to a level that can achieve a solution with $\epsilon$-accuracy. Alternatively, if the initial radius is too small, then $(C_1\|\nabla \phi(x_0)\| - C_2\epsilon_g)/\delta_0$ is larger and the second term represents the number of iterations needed for the TR to expand to a level so that meaningful steps can be taken. Since $\epsilon$ is always assumed to be small, the second term is typically much smaller than the first term.

The above theorem has a “moving component” in $\hat{p}_1$. Maximizing the right-hand side of (4.13) over $\hat{p}_1$ subject to the constraint (4.14) gives us the optimal value for $\hat{p}_1$, which is

$$\frac{1}{2} + \frac{2\epsilon_f + r}{C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2} + \frac{1}{T} \left[ \phi(x_0) - \hat{\phi} \right] C_3\gamma^2(C_1\epsilon - C_2\epsilon_g)^2 + \frac{1}{2} \log \gamma \left( \min \left\{ \frac{C_1\epsilon - C_2\epsilon_g}{\delta_0}, \frac{\delta_0}{C_1\|\nabla \phi(x_0)\| - C_2\epsilon_g} \right\} \right) + \frac{1}{2}.$$
Lemma 4.14. Let
\[ p_0 = 1 - 2 \exp \left( a(\epsilon_f - r/2) \right), \]
where \( a \) is the positive constant from Oracle 0.2. Then, we have the submartingale condition
\[ \mathbb{P} \{ J_k = 1 | \mathcal{F}_{k-1}^r \} \geq p_0, \quad \text{for all } k \in \{0, 1, \ldots \}, \]
where \( \mathcal{F}_{k-1}^r \) is the \( \sigma \)-algebra defined in (3.12). Furthermore, if \( r > 2 \epsilon_f + \frac{2}{a} \log 4 \) (equivalent to \( p_0 > 1/2 \)), then for any positive integer \( T \) and any \( \hat{p}_0 \in [0, p_0] \), it follows that
\[ \mathbb{P} \left\{ \sum_{k=0}^{T-1} J_k > \hat{p}_0 T \right\} \geq 1 - \exp \left( \frac{-(1-\hat{p}_0/p_0)^2}{2} T \right). \]  

Proof. By the definition of Oracle 0.2, we have
\[ \mathbb{P} \{ \xi_k^{(0)} \left| \{ \epsilon(X_k, \xi_k(0)) > t \} \mathcal{F}_{k-1}^r \} \leq \exp (a(\epsilon_f - t)) \]
and
\[ \mathbb{P} \{ \xi_k^{(0+)} \left| \{ \epsilon(X_k^+, \xi_k(0+)) > t \} \mathcal{F}_{k-1}^r \} \leq \exp (a(\epsilon_f - t)) \]
for all \( k \in \{0, 1, \ldots \} \). Consequently,
\[ \mathbb{P} \{ J_k = 0 | \mathcal{F}_{k-1}^r \} = \mathbb{P} \left\{ r < \mathcal{E}_k + \mathcal{E}_k^+ | \mathcal{F}_{k-1}^r \right\} \]
\[ \leq \mathbb{P} \left\{ \mathcal{E}_k > r/2 \text{ or } \mathcal{E}_k^+ > r/2 | \mathcal{F}_{k-1}^r \right\} \]
\[ \leq \mathbb{P} \{ \mathcal{E}_k > r/2 | \mathcal{F}_{k-1}^r \} + \mathbb{P} \{ \mathcal{E}_k^+ > r/2 | \mathcal{F}_{k-1}^r \} \]
\[ \leq \exp (a(\epsilon_f - r/2)) + \exp (a(\epsilon_f - r/2)) = 1 - p_0. \]

Thus, (4.17) holds, and the random process \( \left\{ \sum_{k=0}^{t-1} J_k - p_0 t \right\} \) is a submartingale. Since
\[ \left| \left( \sum_{k=0}^{(t+1)-1} J_k - p_0(t+1) \right) - \left( \sum_{k=0}^{t-1} J_k - p_0 t \right) \right| = |J_t - p_0| \leq \max \{ |0 - p_0|, |1 - p_0| \} = p_0 \]
for any \( t \in \mathbb{N} \), by the Azuma-Hoeffding inequality, we have for any positive integer \( T \) and any positive real \( \epsilon \)
\[ \mathbb{P} \left\{ \sum_{k=0}^{T-1} J_k - T p_0 \leq -\epsilon \right\} \leq \exp \left( \frac{-\epsilon^2}{2T p_0^2} \right). \]

Setting \( \epsilon = (p_0 - \hat{p}_0)T \) and subtracting 1 from both sides yields (4.18). \( \square \)

We largely rely on [22, Chapter 2] when it comes to the analysis of subexponential random variables, but the results in that book are not sufficient for our convergence analysis as the subexponential distribution defined there is controlled by one parameter, whereas the one in Oracle 0.2 is controlled by two parameters, \( a \) and \( \epsilon_f \). A somewhat stronger result is stated in the following proposition and its proof offered in Appendix A.

Proposition 4.15. Let \( X \) be a random variable such that for some \( a > 0 \) and \( b \geq 0 \),
\[ \mathbb{P} \{ |X| \geq t \} \leq \exp (a(b - t)), \quad \text{for all } t > 0. \]

Then, it follows that
\[ \mathbb{E} \exp (\lambda |X|) \leq \frac{1}{1 - \lambda/a} \exp (\lambda b), \quad \text{for all } \lambda \in [0, a). \]  

22
With Proposition 4.15, we establish in Lemma 4.16 a probabilistic upper bound on the total increase in the objective function value caused by the noise in the function evaluations.

**Lemma 4.16.** With Oracle 0.2, for any \( t \geq 0 \),

\[
P \left\{ \sum_{k=0}^{T-1} \Theta_k^\prime \left( \mathcal{E}_k + \mathcal{E}_k^+ \right) \geq T(4/a + 2\epsilon_f) + t \right\} \leq \exp \left( -\frac{a}{4} t \right). \tag{4.21}
\]

**Proof.** With Oracle 0.2, \( P \{ \mathcal{E}_k \geq t | \mathcal{F}_{k-1}^\prime \} \leq \exp(a(\epsilon_f - t)) \), and by Proposition 4.15 it follows that

\[
E \{ \exp(2\lambda \mathcal{E}_k) | \mathcal{F}_{k-1}^\prime \} \leq \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f) \quad \text{for all } \lambda \in \left[ 0, \frac{a}{2} \right).
\]

A similar result applies to \( \mathcal{E}_k^+ \). Then, by the Cauchy-Schwarz inequality, it follows that for all \( \lambda \in \left[ 0, \frac{a}{2} \right) \)

\[
E \{ \exp(\lambda\mathcal{E}_k + \lambda\mathcal{E}_k^+) | \mathcal{F}_{k-1}^\prime \} \leq \sqrt{E \{ \exp(2\lambda \mathcal{E}_k) | \mathcal{F}_{k-1}^\prime \} \cdot E \{ \exp(2\lambda \mathcal{E}_k^+) | \mathcal{F}_{k-1}^\prime \}}
\leq \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f). \tag{4.22}
\]

By Markov’s inequality, for any \( \lambda \in \left[ 0, \frac{a}{2} \right), t \geq 0 \), and positive integer \( T \),

\[
P \left\{ \sum_{k=0}^{T-1} \Theta_k^\prime \left( \mathcal{E}_k + \mathcal{E}_k^+ \right) \geq t \right\} \leq P \left\{ \sum_{k=0}^{T-1} (\mathcal{E}_k + \mathcal{E}_k^+) \geq t \right\} = P \left\{ \exp \left( \lambda \sum_{k=0}^{T-1} (\mathcal{E}_k + \mathcal{E}_k^+) \right) \geq \exp(\lambda t) \right\}
\leq e^{-\lambda t} \left\{ \exp \left( \lambda \sum_{k=0}^{T-1} (\mathcal{E}_k + \mathcal{E}_k^+) \right) \right\} \leq e^{-\lambda t} \left( \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f) \right)^T,
\]

where the last inequality can be proved by induction as follows. Firstly, this inequality holds for \( T = 1 \) due to (4.22). Now if it holds for any positive integer \( T \), then for \( T + 1 \)

\[
e^{-\lambda t} \left( \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f) \right)^T \leq e^{-\lambda t} \left( \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f) \right)^{T+1} = e^{-\lambda t} \left( \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f) \right)^T \cdot \frac{1}{1 - 2\lambda/a} \exp(2\lambda \epsilon_f),
\]

which shows this inequality holds for \( T + 1 \) and thus for any positive integer \( T \). For ease of exposition, we use the fact that \( 1/(1 - x) \leq \exp(2x) \) for all \( x \in [0, 1/2] \) to simplify the above result

\[
P \left\{ \sum_{k=0}^{T-1} \Theta_k^\prime \left( \mathcal{E}_k + \mathcal{E}_k^+ \right) \geq t \right\} \leq e^{-\lambda t} \left[ \exp(4\lambda/a) \exp(2\lambda \epsilon_f) \right]^T = \exp \left( \lambda \left[ T(4/a + 2\epsilon_f) - t \right] \right) \quad \text{for all } \lambda \in \left[ 0, \frac{a}{4} \right).
\]

Clearly the right-hand side is only less than or equal to 1 when \( t \geq T(4/a + 2\epsilon_f) \), which makes the right-hand side a monotonically non-increasing function of \( \lambda \). We choose \( \lambda = a/4 \) and apply a change of variable to obtain the final result. \( \square \)
Similar to Lemma 4.10, we combine the inequalities from the established lemmas.

**Lemma 4.17.** Under Oracle 0.2, it holds for Algorithm 1 that

\[
\Pr \left\{ \left( \hat{\rho}_0 + \hat{\rho}_1 - \frac{3}{2} - \frac{2\epsilon f + 4/a + r}{h(\gamma \Delta)} \right) T < \frac{\phi(x_0) - \hat{\phi} + t}{h(\gamma \Delta)} + \frac{1}{2} \log \frac{\Delta}{\delta_0} + \frac{1}{2} \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{\rho}_1/p_1)^2}{2} T \right) - \exp \left( - \frac{(1 - \hat{\rho}_0/p_0)^2}{2} T \right) - \exp \left( - \frac{a}{4} t \right)
\]

(4.23)

for any positive integer \( T \), any \( \hat{\rho}_1 \in [0, p_1] \), and any \( t \geq 0 \).

**Proof.** Multiply

\[
\sum_{k=0}^{T-1} \Theta_k(1 - \Lambda'_k) + (1 - \Theta_k)(1 - \Lambda_k) + (1 - \Theta_k)\Lambda_k + \Theta_k \Lambda'_k = \sum_{k=0}^{T-1} 1 = T
\]

with \( (2\epsilon f + 4/a + r)/h(\gamma \Delta) + 0.5 \) and (4.9) with 0.5 and add the two expressions to obtain

\[
\frac{2\epsilon f + 4/a + r}{h(\gamma \Delta)} T + \sum_{k=0}^{T-1} \Theta_k(1 - \Lambda'_k) + (1 - \Theta_k)\Lambda_k < \left( \frac{2\epsilon f + 4/a + r}{h(\gamma \Delta)} + \frac{1}{2} \right) T + \frac{1}{2} \log \frac{\Delta}{\delta_0} + \frac{1}{2}.
\]

(4.24)

Then,

\[
\Pr \left\{ \frac{2\epsilon f + 4/a + r}{h(\gamma \Delta)} T - \sum_{k=0}^{T-1} \Theta_k \Lambda'_k < \left( \frac{2\epsilon f + 4/a + r}{h(\gamma \Delta)} + \frac{3}{2} - \hat{\rho}_0 - \hat{\rho}_1 \right) T + \frac{1}{2} \log \frac{\Delta}{\delta_0} + \frac{1}{2} \right\} \geq \Pr \left\{ \sum_{k=0}^{T-1} -\Theta_k - (1 - \Theta_k)\Lambda_k < (1 - \hat{\rho}_0 - \hat{\rho}_1) T \right\}
\]

\[
= \Pr \left\{ \sum_{k=0}^{T-1} -\Theta_k - (1 - \Theta_k)\Lambda_k < (1 - \hat{\rho}_0 - \hat{\rho}_1) T \right\}
\]

\[
\geq \Pr \left\{ \sum_{k=0}^{T-1} [-\Theta_k - (1 - \Theta_k)\Lambda_k] I_k J_k + (1 - I_k)(1 - J_k) < (1 - \hat{\rho}_0 - \hat{\rho}_1) T \right\}
\]

\[
= \Pr \left\{ \sum_{k=0}^{T-1} -I_k J_k + (1 - I_k)(1 - J_k) < (1 - \hat{\rho}_0 - \hat{\rho}_1) T \right\}
\]

\[
\geq \Pr \left\{ \sum_{k=0}^{T-1} I_k > \hat{\rho}_1 T \text{ and } \sum_{k=0}^{T-1} J_k > \hat{\rho}_0 T \right\}
\]

\[
= \Pr \left\{ \sum_{k=0}^{T-1} I_k > \hat{\rho}_1 T \right\} + \Pr \left\{ \sum_{k=0}^{T-1} J_k > \hat{\rho}_0 T \right\} - \Pr \left\{ \sum_{k=0}^{T-1} I_k > \hat{\rho}_1 T \text{ or } \sum_{k=0}^{T-1} J_k > \hat{\rho}_0 T \right\}
\]

\[
\geq 1 - \exp \left( - \frac{(1 - \hat{\rho}_1/p_1)^2}{2} T \right) + 1 - \exp \left( - \frac{(1 - \hat{\rho}_0/p_0)^2}{2} T \right) - 1
\]

\[
= 1 - \exp \left( - \frac{(1 - \hat{\rho}_1/p_1)^2}{2} T \right) - \exp \left( - \frac{(1 - \hat{\rho}_0/p_0)^2}{2} T \right),
\]

24
where the second equality holds due to Corollary 4.4, and the last inequality holds due to Lemmas 4.9 and 4.14. Unlike the bounded noise case, Lemmas 4.7 and 4.16 imply
\[
\begin{align*}
    &h(\gamma \Delta) \sum_{k=0}^{T-1} \Theta_k \Lambda_k \leq \phi(x_0) - \hat{\phi} + \sum_{k=0}^{T-1} \Theta_k (\mathcal{E}_k + \mathcal{E}_k^+ + r) \\
    &\leq \phi(x_0) - \hat{\phi} + (2\epsilon_f + 4/a + r)T + t
\end{align*}
\]
with probability at least \(1 - \exp(-at/4)\) for any \(t \geq 0\). Thus, we can combine the two previous results to obtain (4.23).

We present our high probability complexity bound for Algorithm 1 with Oracle 0.2 in the following theorem. The proof is analogous to that of Theorem 4.11 and hence omitted.

**Theorem 4.18.** Let Assumptions 2.1 and 2.3 hold for the objective function \(\phi\). Let Assumption 3.1 hold for Algorithm 1. Let \(p_0\) be defined as in (4.16). Given any \(\epsilon > \sqrt{4\epsilon_f + 8/a + 2r} + \epsilon_g\), where \(C_1, C_2\) and \(C_3\) are defined in (4.3) and (4.4), the sequence of iterates generated by Algorithm 1 with Oracle 0.2 satisfy
\[
\mathbb{P} \left\{ \min_{0 \leq k \leq T-1} \| \nabla \phi(X_k) \| \leq \epsilon \right\} \geq 1 - \exp\left( -\frac{(1 - \hat{p}_1/p_1)^2}{2} T \right) - \exp\left( -\frac{(1 - \hat{p}_0/p_0)^2}{2} T \right) - \exp\left( -\frac{a}{4} t \right)
\]
for any \(\hat{p}_0\) and \(\hat{p}_1\) such that \(\hat{p}_0 + \hat{p}_1 \in \left[ \frac{a}{2} + \frac{2\epsilon_f + 4/a + r}{C_3 \gamma^2(C_1 \epsilon - C_2 \epsilon_g)^2}, p_0 + p_1 \right] \), any \(t \geq 0\), and any
\[
T \geq \left( \hat{p}_0 + \hat{p}_1 - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{C_3 \gamma^2(C_1 \epsilon - C_2 \epsilon_g)^2} \right)^{-1} \left[ \frac{\phi(x_0) - \hat{\phi} + t}{C_3 \gamma^2(C_1 \epsilon - C_2 \epsilon_g)^2} + \frac{1}{2} \log \gamma \left( \min \left\{ \frac{C_1 \epsilon - C_2 \epsilon_g}{\delta_0}, \frac{\delta_0}{C_1 \| \nabla \phi(x_0) \| - C_2 \epsilon_g} \right\} \right) + \frac{1}{2} \right].
\]
(4.26)

Similar to Theorem 4.11, one can optimize the bounds in Theorem 4.18 by maximizing the right-hand side of (4.25) while satisfying (4.26). However, since there is no closed form solution, we omit this corollary.

## 5 Second-order stochastic convergence analysis

The convergence analysis of Algorithm 2 is analogous to that of Algorithm 1. The main goal is to derive a probabilistic result of the form
\[
\mathbb{P} \left\{ \min_{0 \leq k \leq T-1} \beta(X_k) < \epsilon \right\} \geq \text{a function of } T \text{ that converges to 1 as } T \text{ increases}
\]
for any sufficiently large \(\epsilon\), where
\[
\beta(x) \triangleq \max \left\{ C_4 \| \nabla \phi(x) \| - C_5 \epsilon_g, -C_6 \lambda_{\min}(\nabla^2 \phi(x)) - C_7 \epsilon_H \right\},
\]
and \(C_4, C_5, C_6,\) and \(C_7\) are positive constants defined in (5.3). The specific lower bounds on \(\epsilon\) will be presented in the main theorems.
Analogous to Section 4, we begin by stating and proving three key lemmas about the behavior of Algorithm 2 under Assumptions 2.2 and 3.1 and assuming (3.7) holds. The first lemma provides a sufficient condition for accepting a step.

**Lemma 5.1 (Sufficient condition for accepting step).** Under Assumptions 2.2 and 3.1, if (3.7) holds, \( r \geq e^+_k - e_k + e^3/2 \), \( \delta_k \leq 1 \) from Assumption 5.6, and

\[
\delta_k \leq \max \left\{ \min \left\{ \frac{1}{\kappa_{bh}} - \frac{1}{L_2 + 2\kappa_{eg} + 2 + \kappa_{eh}} \right\} \right\} \cdot (g_k), \quad (5.1)
\]

then \( \rho_k \geq \eta_1 \) in Algorithm 2.

**Proof.** First, let

\[
\rho_k = \frac{\phi(x_k) - \phi(x_k + s_k) - e_k^+ + r}{m_k(x_k) - m_k(x_k + s_k)}
\]

\[
\geq \frac{\phi(x_k) - \phi(x_k + s_k) + e^3/2}{m_k(x_k) - m_k(x_k + s_k)}
\]

\[
(3.9)
\]

\[
\geq \frac{\phi(x_k) - m_k(x_k + s_k) - (L_2/6 + \kappa_{eg} + 1 + \kappa_{eh}/2)\delta_k^3 - \epsilon_H\delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)}
\]

\[
= 1 - \frac{(L_2/6 + \kappa_{eg} + 1 + \kappa_{eh}/2)\delta_k^3 + \epsilon_H\delta_k^2/2}{m_k(x_k) - m_k(x_k + s_k)}
\]

\[
(3.4)
\]

\[
\geq 1 - \frac{\kappa_{fo} \max \{g_k \min \{g_k, H_k, \delta_k\}, -\lambda_{min}(H_k)\delta_k\}}{\kappa_{fo} g_k \delta_k}
\]

We consider two cases. If the first term in the maximization operation in (5.1) is larger, it follows that

\[
\rho_k \geq 1 - \frac{(L_2/3 + 2\kappa_{eg} + 2 + \kappa_{eh})\delta_k^3 + \epsilon_H\delta_k^2}{\kappa_{fo} g_k \delta_k} \geq 1 - \frac{(L_2/3 + 2\kappa_{eg} + 2 + \kappa_{eh})\delta_k^3 + \epsilon_H\delta_k^2}{\kappa_{fo} g_k \delta_k} \geq 1 - (1 - \eta_1) = \eta_1.
\]

If the second term in the maximization operation in (5.1) is larger, then

\[
\rho_k \geq 1 - \frac{(L_2/3 + 2\kappa_{eg} + 2 + \kappa_{eh})\delta_k^3 + \epsilon_H\delta_k^2}{\kappa_{fo} \lambda_{min}(H_k)\delta_k^2} \geq 1 - (1 - \eta_1) = \eta_1.
\]

\[
\square
\]

The next result provides a sufficient condition for a successful step.

**Lemma 5.2 (Sufficient condition for successful step).** Under Assumptions 2.2 and 3.1, if (3.7) holds, \( r \geq e^+_k - e_k + e^3/2 \), \( \delta_k \leq 1 \), and

\[
\delta_k \leq \beta(x_k) \overset{\text{def}}{=} \max \left\{ C_4 \| \nabla \phi(x_k) \| - C_5 \epsilon, -C_6 \lambda_{min}(\nabla^2 \phi(x_k)) - C_7 \epsilon_H \right\},
\]

where

\[
C_4 \overset{\text{def}}{=} \min \left\{ \frac{1}{\kappa_{bh} + \kappa_{eg}}, \frac{(1 - \eta_1)\kappa_{fo}}{L_2 + 2\kappa_{eg} + 3 + \kappa_{eh} + (1 - \eta_1)\kappa_{fo} \kappa_{eg}}, \frac{1}{\kappa_{eg} + \eta_2} \right\}
\]

\[
C_5 \overset{\text{def}}{=} \max \left\{ \frac{1}{\kappa_{bh} + \kappa_{eg}}, \frac{(1 - \eta_1)\kappa_{fo}}{L_2 + 2\kappa_{eg} + 3 + \kappa_{eh} + (1 - \eta_1)\kappa_{fo} \kappa_{eg}}, \frac{1}{\kappa_{eg} + \eta_2} \right\}
\]

\[
C_6 \overset{\text{def}}{=} \min \left\{ \frac{1}{(1 - \eta_1)\kappa_{fo} \kappa_{eg} + \kappa_{eh} + \eta_2}, \frac{1}{\kappa_{fo} \kappa_{eg} + \kappa_{eh} + \eta_2} \right\}
\]

\[
C_7 \overset{\text{def}}{=} \max \left\{ \frac{1}{(1 - \eta_1)\kappa_{fo} + 1}, \frac{1}{L_2 + 2\kappa_{eg} + 2 + \kappa_{eh} + (1 - \eta_1)\kappa_{fo} \kappa_{eg} + \kappa_{eh} + \eta_2} \right\}
\]
then, \( \rho_k \geq \eta_1 \) and \( \beta_k^m \geq \eta_2 \delta_k \) in Algorithm 2.

**Proof.** By (3.7) we have

\[
\|g_k\| \geq \|\nabla \phi(x_k)\| - \|g_k - \nabla \phi(x_k)\| \quad \text{and} \quad -\lambda_{\min}(H_k) \geq -\lambda_{\min}(\nabla^2 \phi(x_k)) - H_k - \nabla^2 \phi(x_k) \geq -\lambda_{\min}(\nabla^2 \phi(x_k)) - \kappa_{ch} \delta_k - \epsilon_H.
\]

We first show \( \rho_k \geq \eta_1 \). We consider two cases. If the first term in the maximization operation in (5.2) is larger, then

\[
\|g_k\| \geq \|\nabla \phi(x_k)\| - \kappa_{eg} \delta_k^2 - \epsilon_g, \quad \text{and} \quad \|\nabla \phi(x_k)\| - \kappa_{eg} \delta_k - \epsilon_g \geq \delta_k.
\]

and

\[
\frac{(1 - \eta_1)\kappa_{fod}}{L_2/3 + 2\kappa_{eg} + 2 + \kappa_{ch} + \epsilon_H} \|g_k\| \geq \frac{(1 - \eta_1)\kappa_{fod}}{L_2/3 + 2\kappa_{eg} + 2 + \kappa_{ch} + 1}(\|\nabla \phi(x_k)\| - \kappa_{eg} \delta_k - \epsilon_g) \geq \delta_k.
\]

Alternatively, if the second term in the maximization operation in (5.2) is larger, then

\[
\frac{-\lambda_{\min}(H_k) - \epsilon_H}{L_2/3 + 2\kappa_{eg} + 2 + \kappa_{ch}} \geq \frac{-(\lambda_{\min}(\nabla^2 \phi(x_k)) - \kappa_{ch} \delta_k - \epsilon_H)}{L_2/3 + 2\kappa_{eg} + 2 + \kappa_{ch}} \geq \delta_k.
\]

Thus, \( \rho_k \geq \eta_1 \) according to Lemma 5.1.

For the condition \( \beta_k^m \geq \eta_2 \delta_k \), first consider the case where the first term in the maximization operation in (5.2) is larger, then

\[
\|g_k\| \geq \|\nabla \phi(x_k)\| - \kappa_{eg} \delta_k^2 - \epsilon_g \geq \|\nabla \phi(x_k)\| - \kappa_{eg} \delta_k - \epsilon_g \geq \eta_2 \delta_k.
\]

If the second term in the maximization operation in (5.2) is larger, then

\[
-\lambda_{\min}(H_k) \geq -\lambda_{\min}(\nabla^2 \phi(x_k)) - \kappa_{ch} \delta_k - \epsilon_H \geq \eta_2 \delta_k.
\]

The last result provides a bound on the progress made at each iteration. The proof uses similar arguments as in Lemma 4.3 and [13, Lemma 3.7].

**Lemma 5.3 (Progress made in each iteration).** Let Assumption 3.6 hold. In Algorithm 2, if \( \rho_k \geq \eta_1 \) and \( \beta_k^m \geq \eta_2 \delta_k \), then

\[
\phi(x_k) - \phi(x_{k+1}) \geq h(\delta_k) - \epsilon_k + e_k^+ - r,
\]

where

\[
h(\delta) = C_8 \min \{1, \delta^3\}, \text{ where } C_8 \overset{\text{def}}{=} \frac{\eta_1 \eta_2 \kappa_{fod}}{2 \kappa_{fod} \min \left\{ \frac{\eta_2}{\kappa_{fod}}, 1 \right\} \min \left\{ \frac{\eta_2}{\kappa_{fod}}, 1 \right\} (5.4)
\]

If \( \rho_k \geq \eta_1 \) but \( \beta_k^m < \eta_2 \delta_k \), then

\[
\phi(x_k) - \phi(x_{k+1}) \geq -\epsilon_k + e_k^+ - r. \quad (5.5)
\]

If \( \rho_k < \eta_1 \), then \( \phi(x_{k+1}) = \phi(x_k) \).

27
Proof. Similar to Lemma 4.3, let \( \rho_k \geq \eta_1 \) and we have \( \phi(x_k) - \phi(x_{k+1}) \geq \eta_1 [m_k(x_k) - m_k(x_k + s_k)] - \epsilon_k + \epsilon_k^+ - r \). If \( \beta_k^m = -\lambda_{\min}(H_k) \geq \eta_2 \delta_k \), the first term of this expression satisfies
\[
\eta_1 [m_k(x_k) - m_k(x_k + s_k)] \quad (3.4) \quad \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \left( -\lambda_{\min}(H_k) \delta_k^2 \right) \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \eta_2 \delta_k^3;
\]
If \( \beta_k^m = \|g_k\| \geq \eta_2 \delta_k \), it satisfies
\[
\eta_1 [m_k(x_k) - m_k(x_k + s_k)] \quad (3.4) \quad \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\} \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \eta_2 \delta_k \min \left\{ \frac{\eta_2 \delta_k}{\kappa_{bhm}}, \delta_k \right\}
\]
\[
\geq \left\{ \begin{array}{ll}
\frac{\eta_1 \kappa_{\text{fod}}}{2} \min \left\{ \frac{\eta_2}{\kappa_{bhm}}, 1 \right\} \delta_k^2 & \text{if } \delta_k \leq 1 \\
\frac{\eta_1 \kappa_{\text{fod}}}{2} \min \left\{ \frac{\eta_2}{\kappa_{bhm}}, 1 \right\} \delta_k^2 & \text{if } \delta_k > 1
\end{array} \right.
\]
If \( \beta_k^m < \delta_k \), then
\[
\eta_1 [m_k(x_k) - m_k(x_k + s_k)] \quad (3.4) \quad \geq \frac{\eta_1 \kappa_{\text{fod}}}{2} \max \left\{ \|g_k\| \min \left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\}, -\lambda_{\min}(H_k) \delta_k^2 \right\} \geq 0.
\]
If \( \rho_k < \eta_1 \), we have \( x_{k+1} = x_k \), so \( \phi(x_{k+1}) = \phi(x_k) \). \( \Box \)

Next, to categorize the iterations \( k = 0, 1, \ldots, T - 1 \) into different types, we redefine the random indicator variables as follows:
\[
I_k = \mathbb{I} \left\{ \|\nabla^2 M_k(X_k) - \nabla^2 \phi(X_k)\| \leq \kappa_{\text{eh}} \Delta_k + \epsilon_H \right\},
J_k = \mathbb{I} \left\{ r \geq E_k^+ + \epsilon_k + \epsilon_k^{3/2} \right\},
\Theta_k = \mathbb{I} \left\{ \rho_k \geq \eta_1 \text{ and } \beta_k^m \geq \eta_2 \Delta_k \right\},
\]
where \( \beta_k^m \) is defined in (3.5) and \( \Delta \) is now defined as
\[
\Delta = \min_{0 \leq k \leq T-1} \beta(X_k).
\]
The interpretations of the indicators variables \( I_k \), \( J_k \) and \( \Theta_k \) remain the same as in Section 4, as does the definition of \( \Theta'_k \). The definitions of \( \Lambda_k, \Lambda'_k \), and \( \Delta' \) also remain the same but with the newly defined \( \Delta \).

Similar to the first-order case, with the redefined random indicator variables, it follows from Lemma 5.2 that if \( I_k J_k = 1 \) and \( \Lambda_k = 0 \), then \( \Theta_k = 1 \) (and the step is successful). We formalize this in the corollary below.

**Corollary 5.4 (Corollary to Lemma 5.2).** If \( I_k J_k = 1 \) and \( \Lambda_k = 0 \), then \( \Theta_k = 1 \).

Corollary 5.4 shows that if the gradient and Hessian approximations are accurate and the relaxation parameter is sufficiently large (relative to the noise in the function evaluations), and the TR radius is larger than a threshold, then the iteration is successful.

The following lemma is a direct consequence of the definitions of Oracles 1 and 2 and the new definition of \( I_k \) in (5.6).

**Lemma 5.5.** The random sequence \( \{M_k\} \) satisfies the submartingale condition
\[
P \{ I_k = 1 | \mathcal{F}_{k-1} \} \geq p_1 p_2 \overset{\text{def}}{=} p_{12},
\]
where \( \mathcal{F}_{k-1} \) is defined in (3.11).
With the redefined random variables, Lemmas 4.8 and 4.9 hold for Algorithm 2. Lemma 4.7 also holds for Algorithm 2 with the function \( h \) defined in (5.4), and Lemma 4.14 holds with \[
po = 1 - 2 \exp \left( \frac{a}{2} (2\epsilon_f + \epsilon_g^{3/2} - r) \right). \tag{5.9}
\]
Furthermore, Lemma 4.16 holds without any changes. Therefore, with the newly redefined random variables, \( h \) and \( p_0 \), the two main lemmas that combine inequalities, Lemmas 4.10 and 4.17, still hold. We can arrive at the following two theorems for both bounded and subexponential noise cases. Their proofs are analogous to those of Theorems 4.11 (with Lemma 4.10) and 4.18 (with Lemma 4.17), respectively, and, thus, for brevity we provide a sketch of the proof for Theorem 5.6 and omit the others.

**Theorem 5.6.** Let Assumptions 2.2 and 2.3 hold for the objective function \( \phi \). Let Assumption 3.1 and \( r \geq 2\epsilon_f + \epsilon_g^{3/2} \) hold for Algorithm 2. Under Assumption 3.6, given any \( \epsilon > \sqrt{\frac{4\epsilon_f + 2r}{C_8\gamma^4(2p_{12} - 1)}} \), where \( C_8 \) is defined in (5.4), the sequence of iterates generated by Algorithm 2 with Oracle 0.1 satisfies

\[
\Pr \left\{ \min_{0 \leq k \leq T-1} \beta(X_k) \leq \epsilon \right\} \geq 1 - \exp \left( -\frac{(1-\hat{p}_{12}/p_{12})^2}{2} T \right) \tag{5.10}
\]

for any \( \hat{p}_{12} \in \left[ \frac{1}{2} + \frac{2\epsilon_f + r}{C_8(\gamma)^3}, p_{12} \right] \) and any \( T \geq \left( \hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma)} \right)^{-1} \left[ \phi(x_0) - \hat{\phi} \right] + \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{\epsilon}{\delta_0}, \frac{\delta_0}{\beta(x_0)} \right\} \right) + \frac{1}{2} \right]. \tag{5.11}

**Proof.** Analogous to Lemma 4.10, one can prove that for any positive integer \( T \) and any \( \hat{p}_1 \in [0, p_1] \)

\[
\Pr \left\{ \left( \hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{h(\gamma)} \right) T < \phi(x_0) - \hat{\phi} \right\} + \frac{1}{2} \left( \frac{1}{\delta_0} + \frac{1}{y} \right) \geq 1 - \exp \left( -\frac{(1-\hat{p}_{12}/p_{12})^2}{2} T \right). \tag{5.12}
\]

Now recall \( h(\gamma) = C_8 \min\{1, \gamma^3 \Delta^3\} = C_8 \min\{1, \gamma^3 \min_{0 \leq k \leq T-1} \beta(X_k)\}^3 \) and \( 0 < \gamma < 1. \) Analogous to Theorem 4.11, when \( \Delta \leq \delta_0, \) since the function \( Q \) defined below is a decreasing function of \( y, \)

\[
Q(y) = \left( \hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma)^3 y^3} \right)^{-1} \left[ \phi(x_0) - \hat{\phi} \right] + \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{\epsilon}{\delta_0}, \frac{\delta_0}{\beta(x_0)} \right\} \right) + \frac{1}{2},
\]

condition (5.11) implies \( T \geq Q(\epsilon), \) and the event in (5.12) is \( T < Q(\min\{\gamma^{-1}, \min_{0 \leq k \leq T-1} \beta(X_k)\}), \) we have \( \min\{\gamma^{-1}, \min_{0 \leq k \leq T-1} \beta(X_k)\} < \epsilon. \) Considering \( \epsilon \leq 1 < \gamma^{-1}, \) this means \( \min_{0 \leq k \leq T} \beta(X_k) < \epsilon. \) If \( \Delta > \delta_0, \) we replace the \( 0.5 \log_\gamma(y/\delta_0) \) in function \( Q \) with the constant \( 0.5 \log_\gamma(\delta_0/\beta(x_0)) \) and apply the same argument with the decreasing function.

Additionally, the optimal value for \( \hat{p}_{12} \) in Theorem 5.6 is

\[
1 + \frac{2\epsilon_f + r}{C_8(\gamma)^3} + \frac{1}{T} \left[ \phi(x_0) - \hat{\phi} \right] = \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{\epsilon}{\delta_0}, \frac{\delta_0}{\beta(x_0)} \right\} \right) + \frac{1}{2}
\]

and by setting \( \hat{p}_{12} \) to this value, we derive the following corollary to Theorem 5.6.

**Corollary 5.7.** Under the setting of Theorem 5.6, given any \( \epsilon > \sqrt{\frac{4\epsilon_f + 2r}{C_8\gamma^4(2p_{12} - 1)}} \), it follows that

\[
\Pr \left\{ \min_{0 \leq k \leq T-1} \beta(X_k) \leq \epsilon \right\} \geq 1 - \exp \left( -\frac{(1-\hat{p}_{12}/p_{12})^2}{2p_{12}T} \left( \hat{p}_{12} - \frac{1}{2} - \frac{2\epsilon_f + r}{C_8(\gamma)^3} \right) T - \left[ \phi(x_0) - \hat{\phi} \right] + \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{\epsilon}{\delta_0}, \frac{\delta_0}{\beta(x_0)} \right\} \right) + \frac{1}{2} \right]^2 \right\}
\]

for any \( T \) satisfying (5.11).
The final theorem is for the setting of Oracle 0.2.

**Theorem 5.8.** Let Assumptions 2.2 and 2.3 hold for the objective function $\phi$. Let Assumptions 3.1 holds for Algorithm 2. Let $p_0$ be defined as (5.9). Given $\epsilon > \sqrt{\frac{4T + 8/\alpha + 2\tau}{c_8^2(12\epsilon + 2(1 - 12\epsilon) - 3)}}$, where $C_8$ is defined in (5.4), the sequence of iterates generated by Algorithm 2 with Oracle 0.2 satisfies

$$
\mathbb{P} \left\{ \min_{0 \leq k \leq T-1} \beta(X_k) \leq \epsilon \right\} \geq 1 - \exp \left( - \frac{(1 - \hat{p}_{12}/p_0)^2}{2} T \right) - \exp \left( - \frac{(1 - \hat{p}_{12}/p_0)^2}{2} T \right) - \exp \left( - \frac{\epsilon}{4} t \right)
$$

(5.13)

for any $p_0$ and $\hat{p}_{12}$ such that $\hat{p}_0 + \hat{p}_{12} \in \left( \frac{a}{2} + \frac{2\epsilon_f + 1/\alpha + \tau}{c_8(\gamma t)^3}, p_0 + p_{12} \right]$, any $t \geq 0$, and any

$$
T \geq \left( \hat{p}_0 + \hat{p}_{12} - \frac{3}{2} - \frac{2\epsilon_f + 4/a + r}{C_8(\gamma t)^3} \right)^{-1} \left[ \phi(x_0) - \hat{\phi} + t + \frac{1}{2} \log_\gamma \left( \min \left\{ \frac{\epsilon}{\delta_0}, \frac{\delta_0}{\beta(x_0)} \right\} \right) + \frac{1}{2} \right].
$$

(5.14)

6 Numerical experiments: Adversarial example

In this section, we explore the tightness of our theoretical results through numerical experiments. The goal is to investigate whether over the course of optimization the minimum gradient norm $\|
\nabla \phi(x_k)\|$ encountered by Algorithm 1 in the presence of noise is consistent with our theoretical lower bounds on $\epsilon$. Since our analysis is for the worst-case scenario, we consider synthetic experiments where we injected noise in an adversarial manner to make the algorithm perform as poorly as possible at each step.

First, we choose a simple function, $\phi(x) = L_1 \|x\|^2/2$, that satisfies Assumptions 2.1 and 2.3. We apply Algorithm 1 using linear models, thus, Assumption 3.1 is satisfied with $\kappa_{\text{lin}} = 0$. We do not consider quadratic models in this experiment because developing the adversarial numerical example becomes significantly more complex. The solution of the trust-region subproblem for linear models can be expressed as

$$
s_k = \begin{cases} 
0 & \text{if } g_k = 0 \\
-\frac{\delta}{\|g_k\|} g_k & \text{otherwise}.
\end{cases}
$$

(6.1)

When $s_k = g_k = 0$, the step is rejected. In most of the discussion below, $g_k$ is assumed to be nonzero unless stated otherwise. Given the trial step $s_k$, the noise in the function value is set as follows to “encourage” the algorithm to reject good steps (those that decrease $\phi$) and to accept bad steps (those that increase $\phi$). Specifically, the noise was set as follows,

$$
(e_k, e_k^+) = \begin{cases} 
(-\epsilon_f, +\epsilon_f) & \text{if } \phi(x_k + s_k) \leq \phi(x_k) \\
(+\epsilon_f, -\epsilon_f) & \text{otherwise}.
\end{cases}
$$

(6.2)

The noise setting (6.2) does not ensure the worst-case behavior over all iterations of the algorithm, since accepting a good step may increase the trust-region radius and result in worse performance later on. However, this setting reflects our theoretical analysis, which considers only the worse outcome of each iteration separately.

To generate gradient approximations in an adversarial manner, we again aim for the algorithm to accept steps that make $\phi$ increase as much as possible and avoid taking steps when increasing $\phi$ is not possible. Since our Oracle 1 only offers sufficiently accurate gradient with probability $p_1$, at the beginning of each iteration, we set a random variable $I_k$ to 1 with probability $p_1$ and 0 with probability $1 - p_1$. During iterations where the gradient is not sufficiently accurate ($I_k = 0$), $g_k$ is chosen so that the increase in function value
\( \phi(x_k + s_k) - \phi(x_k) = -L_1 \delta_k \langle x_k, g_k/\|g_k\| \rangle + L_1 \delta_k^2 / 2 \) is maximized under condition that the step is accepted, i.e., \( \rho_k \geq \eta_1 \). Thus, the following problem is solved.

\[
\begin{align*}
\max_{g_k} \quad & \phi(x_k + s_k) - \phi(x_k) = -L_1 \delta_k \langle x_k, g_k/\|g_k\| \rangle + L_1 \delta_k^2 / 2 \\
\text{s.t.} \quad & \rho_k = \frac{L_1 \|x_k\|^2 / 2 - L_1 \|x_k - \delta_k g_k/\|g_k\|\|^2 / 2 + 2 \epsilon_f + r}{\|g_k\| \delta_k} \geq \eta_1 \quad \text{Step accepted.} \quad (6.3)
\end{align*}
\]

If (6.3) is infeasible or its optimal value is less than zero, then \( g_k \) is set to zero so that the step is rejected, otherwise \( g_k \) is set to its optimal solution. We note that expressions for \( \rho_k \) depends on how we use (6.2). Since the goal here is to accept a step with possible increase in \( \phi \), we then set \( (\epsilon_k, \epsilon_k^+) = (+\epsilon_f, -\epsilon_f) \) and expression for \( \rho_k \) is

\[
\rho_k = \frac{\phi(x_k) - \phi(x_k + s_k) + 2 \epsilon_f + r}{m(x_k) - m(x_k + s_k)} = \frac{L_1 \|x_k\|^2 / 2 - L_1 \|x_k - \delta_k g_k/\|g_k\|\|^2 / 2 + 2 \epsilon_f + r}{\|g_k\| \delta_k}. \quad (6.4)
\]

On the other hand, if \( I_k = 1 \), problem (6.3) needs to be solved with the additional constraint that the gradient is sufficiently accurate, i.e.,

\[
\|g_k - \nabla \phi(x_k)\| = \|g_k - L_1 x_k\| \leq \kappa_{eg} \delta_k + \epsilon_g \quad \text{Gradient sufficiently accurate.} \quad (6.5)
\]

Either with the additional constraint (6.5) or not, problem (6.3) is not solved as is, but with the change of variables \( y_1 = \langle x_k, g_k/\|g_k\| \rangle \) and \( y_2 = \|g_k\| \). The reformulation is

\[
\begin{align*}
\min_{y_1,y_2} \quad & y_1 \\
\text{s.t.} \quad & \eta_1 y_2 - L_1 y_1 \leq (2 \epsilon_f + r) / \delta_k - L_1 \delta_k / 2 \quad \text{Step accepted.} \quad (6.6)
\end{align*}
\]

where \( y_2 = \|g_k\| > 0 \) is replaced by \( y_2 \geq \min\{10^{-6}, 10^{-2} \|L_1 x_k\| \} \) so that the “minimization” is well-defined. Constraint (6.5) is then written as

\[
y_2^2 - 2 L_1 y_1 y_2 + (L_1 \|x_k\|)^2 \leq (\kappa_{eg} \delta_k + \epsilon_g)^2 \quad \text{Gradient sufficiently accurate.} \quad (6.7)
\]

Problem (6.6) (with or without (6.7)) can be solved analytically, and the corresponding optimal \( g_k \) can be recovered from the optimal \( y_1 \) and \( y_2 \). The procedures are detailed in Appendix B.

When \( I_k = 1 \), if the optimal value of (6.6)-(6.7) is less than \( \delta_k / 2 \) (meaning the loss is greater than 0), \( g_k \) is set using the optimal values of \( y_1 \) and \( y_2 \); if this problem is infeasible, then \( g_k \) is simply set to \( \nabla \phi(x_k) = L_1 x_k \), since an acceptable step does not exist anyway; and, if the optimal value of this problem is greater than \( \delta_k / 2 \), the algorithm cannot be tricked into taking a step that increases \( \phi \), so the worst-case scenario is when no step is taken at all. In the third case, we try to find a value for \( g_k \) to prevent any step from being taken by solving the two optimization problems (6.8) and (6.9) described below:

\[
\begin{align*}
\max_{y_1,y_2} \quad & \eta_1 y_2 - L_1 y_1 \\
\text{s.t.} \quad & y_2^2 - 2 L_1 y_1 y_2 + (L_1 \|x_k\|)^2 \leq (\kappa_{eg} \delta_k + \epsilon_g)^2 \\
& y_1 < \delta_k / 2 \\
& |y_1| \leq \|x_k\| \quad \text{Try to get the step rejected.} \quad \text{Gradient sufficiently accurate.} \quad (6.8)
\end{align*}
\]

Step leads to loss of progress.
\[
\begin{align*}
\max_{y_1, y_2} & \quad n_1 y_2 - L_1 y_1 \\
\text{s.t.} & \quad y_2^2 - 2L_1 y_1 y_2 + (L_1 \|x_k\|^2)^2 \leq (\kappa_{eg} \delta_k + \epsilon_g)^2 \\
& \quad y_1 \geq \delta_k / 2 \\
& \quad |y_1| \leq \|x_k\| \text{ and } y_2 \geq \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}.
\end{align*}
\]

There are two problems because of (6.2). In the first problem where the second constraint is \(\phi(x_k + s_k) > \phi(x_k)\sqrt{r}\) \(6.4\) is calculated with a \(+2\epsilon_f\) term in the numerator; and, in the second problem where the second constraint is \(\phi(x_k + s_k) \leq \phi(x_k)\), \(\rho_k\) \(6.4\) is calculated with a \(-2\epsilon_f\) term in the numerator. If either one of the two optimal values is greater than \((\pm2\epsilon_f + r)/\delta_k - L_1 \delta_k/2\), then we set \(g_k\) to the corresponding optimal solution, which will lead to a rejection of the step; otherwise, a step that decreases \(\phi\) would be unavoidable, so we try to inject noise that can minimize the decrease in \(\phi\) by solving the following problem:

\[
\begin{align*}
\min_{y_1, y_2} & \quad y_1 \\
\text{s.t.} & \quad y_2^2 - 2L_1 y_1 y_2 + (L_1 \|x_k\|^2)^2 \leq (\kappa_{eg} \delta_k + \epsilon_g)^2 \\
& \quad |y_1| \leq \|x_k\| \text{ and } y_2 \geq \min\{10^{-6}, 10^{-2}\|L_1 x_k\|\}.
\end{align*}
\]

Similar to (6.6), the optimization problems (6.8), (6.9), (6.10), can be solved analytically. See Appendix B for details.

For our experiments, we set the parameters of the objective function to \(n = 20, L_1 = 1\); the parameters for the approximation models to \(p_1 = 0.8, \gamma_{eg} = 1\); and the parameters for the algorithm to \(n_1 = 0.25, n_2 = 1\), \(r = 2\epsilon_f\). Then, the theoretical lower bound on \(\epsilon\) in Theorem 4.11 is \(5\sqrt{30\epsilon_f} + 7/3\epsilon_f \approx 27.39\sqrt{\epsilon_f} + 2.33\epsilon_f\). A minor detail here in calculating this theoretical value is that \(\kappa_{cd}\) is set to 2 because the model decrease is \(\|g_k\|\delta_k\), even though we assumed \(\kappa_{cd} \in (0, 1)\). This is not an issue because the property \(\kappa_{cd} \leq 1\) was never used in the analysis.

We experiment with four noise settings where \((\epsilon_f, \epsilon_g)\) is set to \((0.2, 4), (0.4, 0.2, 0)\) and \((0, 0)\), respectively. We initialize Algorithm 1 at \(x_0 = 1.4 \cdot 1\) and with \(\delta_0 = 0.5\), and inject adversarial noise as described above at each iteration. Figure 2 shows how \(\|\nabla \phi(x_k)\|\) and \(\delta_k\) change over the first 250 iterations. We note that \(\|\nabla \phi(x_k)\|\) stabilizes around 4.8, 4, 1.2, and 0, respectively, for the four noise settings, and executing the same experiment multiple times yields similar results. In comparison, their theoretical lower bounds on \(\epsilon\) are \(21.58(= 12.25 + 9.33 \approx 5\sqrt{30\epsilon_f} + 7/3\epsilon_f/3), 9.33, 12.25,\) and 0, respectively. This indicates that the lower bound on \(\epsilon\) in Theorem 4.11 the coefficient of \(\epsilon_g\) is at most \(7/3\) times its optimal value, but the coefficient of \(\sqrt{\epsilon_f}\) can be up to 10 times as big. This indicates the theoretical lower bound on \(\epsilon\) is not unreasonably large.

## 7 Numerical experiments: Investigating the effect of \(r\)

In this section, we explore numerically the effect of the choice of the hyperparameter \(r\) on the performance of the algorithm. Our theory requires that \(r \geq 2\epsilon_f\) in Algorithm 1 and \(r \geq 2\epsilon_f + \epsilon_g^{3/2}\) in Algorithm 2 to offset the function evaluation errors at \(x_k\) and \(x_k + s_k\). If \(r\) is set smaller than these values, it is possible, under our particular assumptions on the zeroth-order oracle, that the algorithm will fail to make successful steps due to noise in the function evaluations, even if the gradient \(\|\nabla \phi(x)\|\) is not small. Setting \(r\) to be larger than \(2\epsilon_f\) or \(2\epsilon_f + \epsilon_g^{3/2}\) allows the algorithm to progress until \(\|\nabla \phi(x)\|\) reaches the lower bound \(\epsilon\), whose value is monotonically increasing with \(r\), as can be seen in both Theorem 4.11 and Theorem 5.6. In other words, the larger the value of \(r\) the larger is the best achievable accuracy \(\epsilon\) and the complexity bound on \(T\). Thus, when Oracle 0.1 is used, it is clearly optimal to set \(r = 2\epsilon_f\) in Algorithm 1 and \(r = 2\epsilon_f + \epsilon_g^{3/2}\) in Algorithm 2. However, as \(\epsilon_f\) (and \(\epsilon_g\)) may not be known in practice, here we explore the effect of setting \(r\) to a variety of different values with respect to \(\epsilon_f\).
Figure 2: Performance of Algorithm 1 with linear approximation models on $\phi(x) = \frac{L_1 \|x\|^2}{2}$ under adversarial noise when $r = 2\epsilon_f$. 
In the first set of experiments, we used the same setting as described in Section 6, with $\epsilon_f = 0.2$ and $\epsilon_g = 4$ but with $r$ set to different values. Figure 3, along with the first line of Figure 2, shows the change of $\|\nabla \phi(x_k)\|$ and $\delta_k$ over the iterations when $r = 0, \epsilon_f, 2\epsilon_f, 4\epsilon_f, \text{ and } 8\epsilon_f$. For experiments with $r \geq 2\epsilon_f$, the level at which $\|\nabla \phi(x_k)\|$ stabilizes get larger with larger values of $r$. This phenomenon is already suggested by the theory and is expected. However, while the theory suggests that the algorithm may get “stuck” if $r < 2\epsilon_f$, this did not occur in our experiments. Instead, we observe the following behavior when $r = 0$: in the initial stages of the optimization the algorithm makes consistent progress because both $\|\nabla \phi(x_k)\|$ and $\delta_k$ are large enough to overcome the noise. Moreover, setting $r = 0$ prevents increases in $\phi$. However, as $\|\nabla \phi(x_k)\|$ decreases, the gradient estimate and the decrease in function value $f(x_k) - f(x_k + s_k)$ become more dominated by noise. Without any relaxation in the step acceptance criterion, the noise may cause many successive rejected steps, which would also shrink the trust-region. As a result, as $\delta_k$ decreases, function evaluation noise becomes more dominated in $f(x_k) - f(x_k + s_k)$, while the predicted decrease $\|g_k\|\delta_k$ becomes smaller. The resulting effect is that it becomes easier for the adversarial noise to be set so that steps for which $\phi$ actually increases get accepted, which explains the monotonic increase of $\phi$ (and $\|\nabla \phi\|$) in the later stage of the experiment. As $r$ increases, this effect becomes less prominent and did not appear in the case where $r = \epsilon_f$.

In the second set of experiments, we examine how the relaxation $r$ affects a practical derivative-free algorithm known as DFO-TR [1], which does not always abide by the theory in this paper. As a practical algorithm, DFO-TR is different from Algorithms 1 and 2 and contains many small practical enhancements to improve the numerical performance, but in its essence, employs quadratic interpolation approximation and trust-region method. Most importantly, it calculates $\rho_k$ just like Algorithms 1 and 2 except without the relaxation. We add $r$ to the numerator of $\rho_k$ and see how DFO-TR performs with different values for $r$.

The experiment is conducted on the Moré & Wild benchmarking problem set [15]. We first give DFO-TR infinite budget to solve all the problems and record the best solution for each problem as $\phi$. Then the output of the 53 problems are scaled linearly so that $\phi(x_0) = 100$ and $\phi = 0$ for every problem. We replicate each problem 4 times (leading to a total of $53 \times 5 = 265$ problems) and then artificially inject noise uniformly distributed on the interval $[-\epsilon_f, +\epsilon_f]$ with $\epsilon_f = 0.2$ to function evaluations. With the function outputs scaled and the noise added, the problems are solved by DFO-TR subsequently with $r = 0, \epsilon_f, 2\epsilon_f, 4\epsilon_f, \text{ and } 8\epsilon_f$. Each variant of the algorithm is given a 2000 function evaluation budget for each problem. The hyperparameters were tuned to make sure the best of the solutions found by all variants of DFO-TR has a scaled function value close enough to 0. The results are compared and presented in performance and data profiles with the (relative) accuracy level $\tau$ set to $10^{-3}$ and $10^{-5}$ (see [15] for the detail on how these profiles are created). As Figure 4 shows, DFO-TR encounters difficulty if $r < 2\epsilon_f$, especially when trying to solve the problems to higher accuracy. Having an $r$ too large also affects the performance adversely. The best performance comes from the variant with $r = 4\epsilon_f$. Considering that DFO-TR uses Hessian approximation and thus resembles Algorithm 2, whose theory suggests that the optimal $r$ needs to be larger than $2\epsilon_f$, this result is consistent with our theory.

Finally, we repeated the experiment with the uniformly distributed noise replaced by unbounded subexponential noise (Oracle 0.2). Specifically, each time an objective function is evaluated, two random variables are generated - one uniformly distributed on $[0, \epsilon_f]$ and the other exponentially distributed with parameter $a$. The sum of the two random variables is then multiplied by $-1$ with probability 0.5 and then added to the true function value. We chose $\epsilon_f = 0.1$ and $a = 20$ so that the expected magnitude of the noise is 0.1. We test five levels for $r = 0, 0.2, 0.5, 1.25, \text{ and } 3.125$. The results are presented in Figure 5. We see that in this case setting $r$ to 0.5 or 1.25 is advantageous for the algorithm, which is also consistent with our theory.
8 Conclusions

We have proposed first- and second-order modified trust-region algorithms for solving noisy (possibly stochastic) unconstrained nonconvex continuous optimization problems. The algorithms utilize estimates of function and derivative information computed via noisy probabilistic zeroth-, first- and second-order oracles. The noise in these oracles is not assumed to be smaller than constants $\epsilon_f$, $\epsilon_g$ and $\epsilon_H$, respectively. We show that the first-order method (Algorithm 1) can find an $\epsilon$-first-order stationary point with high probability after $O(\epsilon^{-2})$ iterations for any $\epsilon \geq [O(\sqrt{\epsilon_f}) + O(\epsilon_g)]$, and that the second-order method (Algorithm 2) can find an $\epsilon$-second-order critical point for any $\epsilon \geq [O(\sqrt{\epsilon_f}) + O(\sqrt{\epsilon_g}) + O(\epsilon_H)]$ after $O(\epsilon^{-3})$ iterations. Numerical experiments on standard derivative-free optimization problems and problems with adversarial noise illustrate the performance of the modified trust-region algorithms.

Acknowledgements

The authors are grateful to the Associate Editor and two anonymous referees for their valuable comments and suggestions.

References

[1] Afonso S Bandeira, Katya Scheinberg, and Luís N Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. Mathematical programming, 134(1):223–257, 2012.

[2] Afonso S Bandeira, Katya Scheinberg, and Luís N Vicente. Convergence of trust-region methods based on probabilistic models. SIAM Journal on Optimization, 24(3):1238–1264, 2014.

[3] Albert S Berahas, Richard H Byrd, and Jorge Nocedal. Derivative-free optimization of noisy functions via quasi-newton methods. SIAM Journal on Optimization, 29(2):965–993, 2019.

[4] Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. Foundations of Computational Mathematics, pages 1–54, 2021.

[5] Albert S Berahas, Liyuan Cao, and Katya Scheinberg. Global convergence rate analysis of a generic line search algorithm with noise. SIAM Journal on Optimization, 31(2):1489–1518, 2021.

[6] Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust-region method via supermartingales. INFORMS journal on optimization, 1(2):92–119, 2019.

[7] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. Mathematical programming, 134(1):127–155, 2012.

[8] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. SIAM Journal on Numerical Analysis, 28(1):251–265, 1991.

[9] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Mathematical Programming, 169(2):337–375, 2018.

[10] Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. Mathematical Programming, 169(2):447–487, 2018.
A Proofs

Proof to Proposition 2.7

We prove a longer version of Proposition 2.7 which also includes the gradient estimate.
Proposition A.1. Let \( g(x) \) and \( H(x) \) be computed using second-order finite difference:

\[
g(x) = \sum_{i=1}^{n} \frac{4f(x + \sigma u_i) - f(x + 2\sigma u_i) - 3f(x)}{2\sigma} u_i
\]

\[
H(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{f(x + \sigma u_i + \sigma u_j) - f(x + \sigma u_i) - f(x + \sigma u_j) + f(x)}{\sigma^2} u_i u_j
\]

where \( u_i \) denotes the \( i \)-th column of the identity matrix and \( \sigma > 0 \). Then, if \( |f(y) - \phi(y)| \leq \epsilon_f \) when \( y \) is any interpolation point and Assumption 2.2 holds, then for all \( x \in \mathbb{R}^n \) the approximation errors are bounded as

\[
\|g(x) - \nabla \phi(x)\| \leq \sqrt{n}L_2\sigma^2 + \frac{4\sqrt{n}\epsilon_f}{\sigma}
\]

\[
\|H(x) - \nabla^2 \phi(x)\| \leq \frac{(\sqrt{2} + 1)nL_2\sigma}{3} + \frac{4n\epsilon_f}{\sigma^2}.
\]

Proof. It was shown in [16] that under Assumption 2.2 for any \( (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \)

\[
\left| \phi(y) - \phi(x) - \langle \nabla \phi(x), y - x \rangle - \frac{1}{2} \langle \nabla^2 \phi(x)(y - x), y - x \rangle \right| \leq \frac{L_2}{6}||y - x||^3.
\]

As a result, for any \( i \in \{1, 2, \ldots, n\} \), we have the following three inequalities:

\[4\phi(x + \sigma u_i) \leq 4 \left[ \phi(x) + \langle \nabla \phi(x), hu_i \rangle + \frac{1}{2} \langle \nabla^2 \phi(x)\sigma u_i, \sigma u_i \rangle + \frac{L_2}{6}\sigma^3 \right]\]

\[-\phi(x + 2\sigma u_i) \leq -\phi(x) - \langle \nabla \phi(x), 2\sigma u_i \rangle - \frac{1}{2} \langle \nabla^2 \phi(x)2\sigma u_i, 2\sigma u_i \rangle + \frac{4L_2}{3}\sigma^3\]

\[-3\phi(x) \leq -3\phi(x),\]

which can be added together as

\[4\phi(x + \sigma u_i) - \phi(x + 2\sigma u_i) - 3\phi(x) \leq 2\langle \nabla \phi(x), \sigma u_i \rangle + 2L_2\sigma^3.\]

With bounded noise, we have

\[4f(x + \sigma u_i) - f(x + 2\sigma u_i) - 3f(x) \leq 2\langle \nabla \phi(x), \sigma u_i \rangle + 2L_2\sigma^3 + 8\epsilon_f.\]

Using the same argument as above, we also have

\[-4f(x + \sigma u_i) + f(x + 2\sigma u_i) + 3f(x) \leq -2\langle \nabla \phi(x), \sigma u_i \rangle + 2L_2\sigma^3 + 8\epsilon_f.\]

Combining the last two inequalities, we obtain

\[|\langle g(x), u_i \rangle - \langle \nabla \phi(x), u_i \rangle| \leq L_2\sigma^2 + \frac{4\epsilon_f}{\sigma}.
\]

Thus,

\[\|g(x) - \nabla \phi(x)\| \leq \sqrt{n} \left( L_2\sigma^2 + \frac{4\epsilon_f}{\sigma} \right)^2 = \sqrt{n}L_2\sigma^2 + \frac{4\sqrt{n}\epsilon_f}{\sigma},\]

which is the first result of the proposition.
Next, for any $i, j \in \{1, 2, \ldots, n\}^2$, we have four inequalities

$$
\phi(x + x_i + x_j) \leq \phi(x) + \langle \nabla \phi(x), x_i + x_j \rangle + \frac{1}{2} \langle \nabla^2 \phi(x)(x_i + x_j), x_i + x_j \rangle + \frac{L_2}{6}(\sqrt{2}\sigma)^3
$$

$$
-\phi(x + x_i) \leq -\phi(x) - \langle \nabla \phi(x), x_i \rangle - \frac{1}{2} \langle \nabla^2 \phi(x)x_i, x_i \rangle + \frac{L_2}{6}\sigma^3
$$

$$
-\phi(x + x_j) \leq -\phi(x) - \langle \nabla \phi(x), x_j \rangle - \frac{1}{2} \langle \nabla^2 \phi(x)x_j, x_j \rangle + \frac{L_2}{6}\sigma^3
$$

$$
\phi(x) \leq \phi(x),
$$

which can be added together as

$$
\phi(x + x_i + x_j) - \phi(x + x_i) - \phi(x + x_j) + \phi(x) \leq \langle \nabla^2 \phi(x)x_i, x_i \rangle + \langle \nabla^2 \phi(x)x_j, x_j \rangle + \frac{(\sqrt{2} + 1)L_2}{3}\sigma^3.
$$

With bounded noise, we have

$$
f(x + x_i + x_j) - f(x + x_i) - f(x + x_j) + f(x) \leq \langle \nabla^2 \phi(x)x_i, x_i \rangle + \langle \nabla^2 \phi(x)x_j, x_j \rangle + \frac{(\sqrt{2} + 1)L_2}{3}\sigma^3 + 4\epsilon_f.
$$

Again, using the same argument as above, we also have

$$
-f(x + x_i + x_j) + f(x + x_i) + f(x + x_j) - f(x) \leq -\langle \nabla^2 \phi(x)x_i, x_i \rangle - \langle \nabla^2 \phi(x)x_j, x_j \rangle + \frac{(\sqrt{2} + 1)L_2}{3}\sigma^3 + 4\epsilon_f.
$$

Combining the last two inequalities, we obtain

$$
|\langle H(x)x_i, x_j \rangle - \langle \nabla^2 \phi(x)x_i, x_i \rangle| \leq \frac{(\sqrt{2} + 1)L_2}{3}\sigma + 4\epsilon_f.
$$

Then

$$
\|H(x) - \nabla^2 \phi(x)\| \leq \|H(x) - \nabla^2 \phi(x)\|_F \leq \sqrt{n^2 \left( \frac{(\sqrt{2} + 1)L_2}{3}\sigma + \frac{4\epsilon_f}{\sigma^2} \right)^2} = \frac{(\sqrt{2} + 1)L_2}{3}\sigma + \frac{4n\epsilon_f}{\sigma^2},
$$

which is the second result of the proposition.

\[\square\]

**Proof to Proposition 4.15**

*Proof.* By the Taylor series of the exponential functions, it follows that

$$
E\exp(\lambda|X|) = E\sum_{p=0}^{\infty} \frac{1}{p!}(\lambda|X|)^p = 1 + \sum_{p=1}^{\infty} \frac{1}{p!}E|X|^p
$$

for any real $\lambda$. Applying the integral identity [22, Lemma 1.2.1]),

$$
E|X|^p = \int_0^{\infty} P(|X|^p \geq u)du = \int_0^{\infty} P(|X|^p \geq t^p)dt^p \leq \int_0^{\infty} \min\{1, \exp(a(b-t))\}dt^p,
$$

which is valid for all $p > 0$. Since $1 \leq \exp(a(b-t))$ when $t \leq b$, the above result can be written as

$$
E|X|^p \leq \int_0^b pt^{p-1}dt + \int_b^{\infty} pt^{p-1}\exp(a(b-t))dt.
$$
Thus, for all $\lambda \in [0, a)$ it follows that

$$\mathbb{E}\exp(\lambda |X|) \leq 1 + \sum_{p=1}^{\infty} \frac{1}{p!} \lambda^p \left[ \int_0^b pt^{p-1}dt + \int_b^\infty pt^{p-1}\exp(a(b-t))dt \right]$$

$$= 1 + \lambda \sum_{p=1}^{\infty} \frac{1}{(p-1)!} (\lambda t)^{p-1}dt + \lambda e^{ab} \int_b^\infty \lambda e^{-at}dt$$

$$= 1 + \lambda \int_0^b \exp(\lambda t)dt + \lambda e^{ab} \int_b^\infty \exp((\lambda - a)t)dt$$

$$= 1 + \exp(\lambda b) - 1 + \lambda e^{ab} \frac{1}{a - \lambda} \exp((\lambda - a)b)$$

$$= \frac{a}{a - \lambda} \exp(\lambda b).$$

\[\square\]

**B Details on the adversarial gradient estimate**

When $I_k = 1$, problem (6.6) can be written as

$$\min_{y_1, y_2} \quad y_1 \quad \text{Maximize the loss.}$$

s.t. \[y_1 \geq \frac{\eta_1 y_2}{L_1} + \frac{\delta_k}{2} - \frac{2\epsilon_f + r}{L_1 b} y_1 \geq \frac{y_2}{2L_1} + \frac{(L_1 \|x_k\|)^2 - (\kappa_{eg} \delta_k + \epsilon_g)^2}{2L_1 y_2} \quad \text{Step is accepted.}\]

\[-\|x_k\| \leq y_1 \leq \|x_k\| \quad \text{Gradient is sufficiently accurate.} \quad \text{(B.1)}\]

\[y_2 \geq \min\{10^{-6}, 10^{-2}L_1 \|x_k\|\} \quad \text{Ensure } |\langle x_k, g_k/\|g_k\|\rangle| \leq \|x_k\|.\]

\[y_2 \geq \min\{10^{-6}, 10^{-2}L_1 \|x_k\|\} \quad \text{Ensure } \|g_k\| > 0.\]

If $L_1 \|x_k\| \leq \kappa_{eg} \delta_k + \epsilon_g$, all three lower bounds on $y_1$ are monotonically non-decreasing with respect to $y_2$, so we set $y_2$ to min$\{10^{-6}, 10^{-2}L_1 \|x_k\|\}$ and $y_1$ to the largest of the three lower bounds. Then if $y_1 \leq \|x_k\|$ holds, the problem is solved; otherwise the problem is infeasible. Alternatively, if $L_1 \|x_k\| > \kappa_{eg} \delta_k + \epsilon_g$, the second lower bound of $y_1$ becomes a positive convex function of $y_2$. We set $y_2$ to the minimizer of this convex function $\sqrt{(L_1 \|x_k\|)^2 - (\kappa_{eg} \delta_k + \epsilon_g)}^2$, and $y_1$ to its minimum value $\sqrt{(L_1 \|x_k\|)^2 - (\kappa_{eg} \delta_k + \epsilon_g)}^2 / L_1^\frac{3}{2}$. Now if $y_1 \geq \delta_k / 2$, the algorithm cannot be tricked into taking a bad step and we move on to problems (6.8) and (6.9); otherwise all constraints are satisfied except maybe the first one, so we check it. If it is satisfied, the problem is solved; if not, $y_2$ needs to be reduced until the first and second lower bounds of $y_1$ are equal, which requires solving a quadratic equation. We set $y_2$ to the root between 0 and $\sqrt{(L_1 \|x_k\|)^2 - (\kappa_{eg} \delta_k + \epsilon_g)}^2$ and $y_1$ to the resulting lower bound. If this solution satisfies $y_1 \leq \|x_k\|$, the problem is solved, otherwise it is infeasible.

For problems (6.8) and (6.9), where we try to find a value for $g_k$ that would lead to a step being rejected, if $L_1 \|x_k\| \leq \kappa_{eg} \delta_k + \epsilon_g$, we can simply set $g_k = 0$. Now we assume $L_1 \|x_k\| > \kappa_{eg} \delta_k + \epsilon_g$. With the additional
substitution $y_3 = \eta_1 y_2 - L_1 y_1$, problem (6.8) is formulated as

$$\max_{y_2, y_3} y_3$$

s.t. $$y_3 \leq \frac{2\eta_1 - 1}{2} y_2 - \frac{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}{2y_2}$$

$$y_3 \geq \eta_1 y_2 - L_1 \|x_k\|$$

$$y_2 \geq \min\{10^{-6}, 10^{-2} \|L_1 x_k\|\}$$

$$y_3 > \eta_1 y_2 - L_1 \delta_k / 2$$

Try to get the step rejected.

Ensure $\|x_k, g_k / \|g_k\|| \leq \|x_k\|$.

Gradient is sufficiently accurate.

Step leads to loss of progress.

Gradient is sufficiently accurate.

Ensure $\|g_k\| > 0$.

Step leads to gain of progress.

Note the constraint $y_3 \leq \eta_1 y_2 + L_1 \|x_k\|$, which ensures $y_1 = \langle x_k, g_k / \|g_k\| \rangle \geq -\|x_k\|$ is not present because it is covered by the first constraint in (B.2). If $2\eta_1 < 1$, the upper bound on $y_3$ is a concave function of $y_2$. We set $y_2$ to its maximizer $\sqrt[2]{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}/(1 - 2\eta_1)$ and $y_3$ to the optimal value $-\sqrt[2]{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}/(1 - 2\eta_1)$. If this solution is feasible, the problem is solved; otherwise the upper is lower than at least one of the lower bounds, in which case we reduce $y_2$ until the upper and lower bounds of $y_3$ are equal. If $2\eta_1 \geq 1$, the upper bound on $y_3$ increases as $y_2$ increases. When $y_2$ is sufficiently large, the two lower bounds on $y_3$ increase faster with $y_2$ than the upper bound, so $y_2$ can only be increased until the bounds meet. Thus, in either case, we need to solve the quadratic equation

$$y_2 - \frac{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}{2y_2} = \eta_1 y_2 - L_1 \min\{\|x_k\|, \delta_k / 2 - 10^{-7}\}$$

$$y_2^2 - 2L_1 \min\{\|x_k\|, \delta_k / 2 - 10^{-7}\} y_2 + (L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2 = 0,$$

where the $10^{-7}$ is there to deal with the strict inequality. The optimal value for $y_2$ should be its larger root, and $y_3$ is the corresponding bound. If there is no real root, the problem is infeasible.

With the substitutions, problem (6.9) is formulated as

$$\max_{y_3} y_3$$

s.t. $$y_3 \leq \frac{2\eta_1 - 1}{2} y_2 - \frac{(L_1 \|x_k\|)^2 - (\kappa_{\text{eg}} \delta_k + \epsilon_g)^2}{2y_2}$$

$$y_3 \geq \eta_1 y_2 - L_1 \|x_k\|$$

$$y_2 > 0$$

$$y_3 \leq \eta_1 y_2 - L_1 \delta_k / 2$$

Try to get the step rejected.

The gradient is sufficiently accurate.

To ensure $\langle x_k, g_k / \|g_k\|| \leq \|x_k\|$.

Assume $\|x_k\| \geq \delta_k / 2$ for feasibility. If $2\eta_1 < 1$, we first set $y_2, y_3$ to the maximizer and maximum value of the concave right-hand side of the first constraint. Then if this solution violates the last constraint, we set $y_3, y_3$ to the larger one of the two points where the two upper bounds of $y_3$ meet. If this solution instead violates the second constraint or $2\eta_1 \geq 1$, we set $y_2, y_3$ to the larger one of the two points where the right-hand sides of the first two constraints are equal.

Problem (6.10) can be reformulated as (B.1) but without the acceptance constraint. Since we have explained how to analytically solve (B.1), it should be clear how to solve the simpler (6.10). Same goes for problem (6.6) without the sufficiently accurate gradient constraint.

The optimal $g_k$ needs to be recovered from $y_1, y_2$. We let $g_k = \alpha_1 x_k + \alpha_2 v$, where $\alpha_1, \alpha_2$ are real variables, and $v \in \mathbb{R}^n$ is a unit vector with a random direction. By solving the system of equations

$$\langle x_k, \alpha_1 x_k + \alpha_2 v \rangle = y_1 y_2$$

$$\|\alpha_1 x_k + \alpha_2 v\| = y_2,$$

we obtain the values for $\alpha_1, \alpha_2$, and hence can recover $g_k$. 
Figure 3: Performance of Algorithm 1 with linear approximation models on $\phi(x) = L_1 \|x\|^2/2$ under adversarial noise when $r$ is set to various values and $(\epsilon_f, \epsilon_g) = (0.2, 4)$. 
Figure 4: Performance of DFO-TR with relaxed step acceptance criterion on the Moré & Wild problem set with uniformly distributed function evaluation noise.
Figure 5: Performance of DFO-TR with relaxed step acceptance criterion on the Moré & Wild problem set with subexponential function evaluation noise.