Journal of Scientific Computing (2025) 103:43
https://doi.org/10.1007/s10915-025-02855-8

Curvature-Aware Derivative-Free Optimization

Bumsu Kim[1] · Daniel McKenzie[2] · HanQin Cai[3] · Wotao Yin[4]

Received: 30 January 2024 / Revised: 30 December 2024 / Accepted: 22 February 2025

Abstract
The paper discusses derivative-free optimization (DFO), which involves minimizing a function without access to gradients or directional derivatives, only function evaluations. Classical DFO methods such as Nelder-Mead and direct search have limited scalability for high-dimensional problems. Zeroth-order methods, which mimic gradient-based methods, have been gaining popularity due to the demands of large-scale machine learning applications. This paper focuses on the selection of the step size α_k in such methods. The proposed approach, called Curvature-Aware Random Search (CARS), uses first- and second-order finite difference approximations to compute a candidate α_+. A safeguarding step then evaluates α_+ and chooses an alternate step size in case α_+ does not decrease the objective function. We prove that for strongly convex objective functions, CARS converges linearly provided that the search direction is drawn from a distribution satisfying very mild conditions. We also present a **C**ubic **R**egularized variant of CARS, named CARS-CR, which provably converges at a rate of $\mathcal{O}(1/k)$ without the assumption of strong convexity. Numerical experiments show that CARS and CARS-CR match or exceed the state-of-the-art on benchmark problem sets.

Keywords Derivative-free optimization · Zeroth-order optimization · Curvature-aware method · Hessian-aware method · Newton-type method

✉ Daniel McKenzie
dmckenzie@mines.edu

Bumsu Kim
bumsu@ucla.edu

HanQin Cai
hqcai@ucf.edu

Wotao Yin
wotao.yin@alibaba-inc.com

[1] Department of Mathematics, University of California, Los Angeles, Los Angeles, CA, USA

[2] Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, USA

[3] Department of Statistics and Data Science and Department of Computer Science, University of Central Florida, Orlando, FL, USA

[4] Decision Intelligence Lab, DAMO Academy, Alibaba US, Bellevue, WA, USA

Published online: 24 March 2025

1 Introduction

We consider minimizing a function $f : \mathbb{R}^d \to \mathbb{R}$, with only access to function evaluations $f(x)$, and no access to gradients or directional derivatives. This setting is commonly referred to as derivative-free optimization (DFO). DFO has a rich history and has recently gained popularity in various areas such as reinforcement learning [18, 49, 56], hyperparameter tuning [6], adversarial attacks on neural network classifiers [11, 16, 64], and prompt-tuning [61] or fine-tuning [48] for large language models. In all of these applications, evaluating $f(x)$ is either expensive, time-consuming, or inconvenient, and therefore, it is desirable for DFO algorithms to minimize the number of function evaluations required.

Classical methods for DFO include the Nelder-Mead simplex method [52], direct search methods [41], and model-based methods [20]. However, these methods tend to scale poorly with the problem dimension d, although recent works [12, 13, 15] have made progress in this direction. Due to the demands of large-scale machine learning applications, *zeroth-order* (ZO) methods for DFO have gained increasing attention [46]. ZO methods mimic first-order methods like gradient descent but approximate all derivative information using function queries. At each iteration, the algorithm selects a direction u_k and takes a step $x_{k+1} = x_k + \alpha_k u_k$. While the selection of u_k has been well studied (see [3] and references therein), this paper focuses on the selection of α_k, allowing for u_k to be either randomly selected or an approximation to the negative gradient (i.e., $u_k \approx -\nabla f(x_k)$).

Intelligently choosing α_k can lead to convergence in fewer iterations, but this gain may be offset by the number of queries it takes to compute α_k. If we compute $u_k \approx -\nabla f(x_k)$, techniques such as backtracking line search from first-order optimization can be employed [4]. However, obtaining a sufficiently accurate approximation to $-\nabla f(x_k)$ requires $\Omega(d)$ queries per iteration [3], which is impractical for large d. On the other hand, when we take u_k as a random vector, with high probability u_k is almost orthogonal to $-\nabla f(x_k)$. Hence, α_k in [5, 25, 55] is very small to guarantee descent at every iteration (possibly in expectation). Our approach differs from these methods.

We propose using finite difference approximations to the first and second derivatives of the univariate function $\alpha \mapsto f(x_k + \alpha u_k)$ to compute a candidate α_+ for α_k. Specifically, we set

$$\alpha_+ = \frac{d_r}{\hat{L} h_r},$$

where

$$d_r := \frac{f(x_k + r u_k) - f(x_k - r u_k)}{2r},$$
$$h_r := \frac{f(x_k + r u_k) - 2 f(x_k) + f(x_k - r u_k)}{r^2},$$

and $\hat{L}$ is a user-specified parameter. Computing α_+ requires only three queries per iteration. This simple modification to the well-known Random Search algorithm [25, 55] (which takes $\alpha_k = d_r/\hat{L}$ or similar) can be viewed as an inexact one-dimensional Newton's method at each iteration. When encountering low curvature directions, h_r is small and α_+ is large, so this α_+ may occasionally fail to guarantee descent. To remedy this, we combine our step-size rule with a simple safeguarding scheme based on the Stochastic Three Point method [5], thus guaranteeing that $f(x_{k+1}) \leq f(x_k)$ at every iterate. Importantly, we show that α_+ is a good candidate—i.e., $f(x_{k+1})$ is significantly smaller than $f(x_k)$—a positive proportion of the time. From this, we can quantify the expected total number of function queries required

to reach a target solution accuracy. Because our method is a natural extension of Random Search that incorporates second derivative information, we dub it Curvature-Aware Random Search, or CARS.

In addition to CARS, we propose an extension called CARS-CR (CARS with Cubic Regularization) that modifies the stochastic subspace cubic Newton method [34] into a zeroth-order method. CARS-CR is essentially CARS with an adaptive parameter $\hat{L}$ and achieves $\mathcal{O}(k^{-1})$ convergence for convex functions.

Our numerical experiments show that both CARS and CARS-CR outperform state-of-the-art algorithms on benchmarks across various problem dimensions, demonstrating efficiency and robustness. Intriguingly, our results show that CARS (and CARS-CR) are consistently able to achieve a higher accuracy given a fixed query budget than competing algorithms. Furthermore, our results on adversarial attacks show that CARS can be adapted to different sample distributions of u_k. We demonstrate that CARS performs well with a tailored distribution for a particular problem, an adversarial attack on a pre-trained neural network.

Organization. This paper is laid out as follows. In the rest of this section, we fix the notation and discuss prior art. In Sect. 2, we introduce the main algorithm, namely Curvature-Aware Random Search (CARS), along with its convergence analysis. Section 3 extends CARS with Cubic Regularization (CARS-CR) for general convex functions. In Sect. 4, we provide mathematical proofs to support our technical claims. Section 5 contains extensive numerical experiments that empirically verify our technical claims. Section 6 concludes the paper.

1.1 Assumptions and Notation

In developing and analyzing CARS, we assume that f is a convex and twice continuously differentiable function. We use $g(x) = \nabla f(x)$ and $H(x) = \nabla^2 f(x)$ briefly in the theoretical analysis of Sect. 2.1. For a fixed initial point x_0, we define the level-set $\mathcal{Q} = \{x \in \mathbb{R}^d : f(x) \leq f(x_0)\}$, $\|\cdot\|$ as the Euclidean norm, and $f_\star := \min_{x\in\mathbb{R}^d} f(x)$. We say x_k is an ε-optimal solution if $f(x_k) - f_\star \leq \varepsilon$. We use $\mathcal{D}$ to denote a probability distribution on $\mathbb{R}^d$. For any measurable set $S \subseteq \mathbb{R}^d$ with finite measure, $\mathrm{Unif}(S)$ denotes the uniform distribution over S. The unit sphere is written as $\mathbb{S}^{d-1} := \{u : \|u\| = 1\} \subseteq \mathbb{R}^d$, and $e_1, \ldots, e_d$ represent the canonical basis vectors in $\mathbb{R}^d$. For two matrices A and B, we write $A \preceq B$ if $B - A$ is positive semi-definite.

Definition 1 We say f is L-smooth, $L > 0$, if $H(x) \preceq L I_d$ for all $x \in \mathcal{Q}$.

Definition 2 We say f is μ-strongly convex, $\mu > 0$, if $\mu I_d \preceq H(x)$ for all $x \in \mathcal{Q}$.

Under strong convexity, $H(z)$ is positive definite for all $z \in \mathcal{Q}$; hence the following inner product and induced norm are well-defined for all $z \in \mathcal{Q}$:

$$\langle x, y\rangle_{H(z)} := \langle H(z)x, y\rangle \qquad \text{and} \qquad \|x\|^2_{H(z)} := \langle x, x\rangle_{H(z)}.$$

Strong convexity also implies the following [30, Proposition 2].

Lemma 1 ($\hat{L}$-Relative Smoothness and $\hat{\mu}$-Relative Convexity) *If f is μ-strongly convex, then f is $\hat{\mu}$-relatively convex and $\hat{L}$-relatively smooth for some $\hat{L} \geq \hat{\mu} > 0$, i.e. for all $x, y \in \mathcal{Q}$*

$$\frac{\hat{\mu}}{2}\|x - y\|^2_{H(y)} \leq f(x) - f(y) - \langle g(y), x - y\rangle \leq \frac{\hat{L}}{2}\|x - y\|^2_{H(y)}.$$

Remark 1 The relationship between L and $\hat{L}$ (resp. μ and $\hat{\mu}$) is not straightforward. For example, if $f(x) = x^\top Ax$ with positive definite A, then $\hat{\mu} = \hat{L} = 1$ as

$$\begin{aligned} f(x) - f(y) - \langle g(y), x - y\rangle &= x^\top Ax - y^\top Ay - 2y^\top A(x-y) \\ &= x^\top Ax + y^\top Ay - 2y^\top Ax \\ &= (x-y)^\top A(x-y) = \frac{1}{2}\|x-y\|_{H(y)}. \end{aligned}$$

This holds even if A is poorly conditioned, i.e. $L/\mu \gg 1$. Thus, the *relative condition number*, defined as $\hat{L}/\hat{\mu}$, should be thought of as measuring how far f is from being quadratic. This is discussed further in [30, Section 4]. Combining [30, Lemma 8] and [38, Theorem 1] gives that if $\nabla^2 f(x)$ is M-Lipschitz, then

$$\frac{\hat{L}}{\hat{\mu}} \leq \left(1 + \frac{M\mathcal{R}}{\mu}\right)^2,$$

where $\mathcal{R}$ bounds the diameter of the level set $\mathcal{Q}$, and is defined rigorously in Definition 3.

We also make the following regularity assumption on H.

Assumption 1 H is a-Hölder continuous for some $a > 0$, i.e.

$$|u^\top (H(x) - H(y))\, u| \leq L_a \|x - y\|^a \tag{1}$$

for any unit vector $u \in \mathbb{S}^{d-1}$ and $x, y \in \mathcal{Q}$.

Hölder continuity reduces to Lipschitz continuity when $a = 1$. Assumption 1 can be used to refine the relative smoothness and relative convexity constants in a smaller region.

1.2 Prior Art

For a comprehensive introduction to DFO we refer the reader to [20] or the more recent survey article [44]. As mentioned above, our interest is in ZO approaches to DFO [46], as these have low per-iteration query complexity (with respect to the dimension of the problem) and have been successfully used in modern machine learning applications, such as adversarial attacks on neural networks [9, 11, 16, 17, 47] and reinforcement learning [19, 24, 56]. Of particular relevance to this work is ZO algorithms based on *line search*:

$$\text{Sample } u_k \text{ from } \mathcal{D},$$

$$\alpha_k \approx \alpha_\star = \arg\min_{\alpha \in \mathbb{R}} f(x_k + \alpha u_k), \tag{2a}$$

$$x_{k+1} = x_k + \alpha_k u_k, \tag{2b}$$

which may be thought of as zeroth-order analogues of coordinate descent [53]. All of the complexity results discussed below assume *noise-free* access to $f(x)$. The noisy case is more complicated, see [37]. The first papers to use this scheme were [39, 40], where convergence is discussed under the assumptions that u_k is a descent direction[1] for all k and (2a) is solved sufficiently accurately. Assuming (2a) is solved *exactly*, [51] proves this scheme finds an ε-optimal solution in $\mathcal{O}(\log(1/\varepsilon))$ iterations when f is a quadratic function (see also [57] for a discussion of these results in English). In [43], $\mathcal{O}(\log(1/\varepsilon))$ iteration complexity was proved

[1] $u_k^\top \nabla f(x_k) < 0$.

Table 1 Comparison of various line-search based ZO algorithms, all of which use random search directions

Algorithm	Strg. Convex	Convex	Queries/Iter	Line Search Oracle
[40]	$\mathcal{O}(\log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon)$	–	Yes
[43]	$\mathcal{O}(\log(1/\varepsilon))$	–	–	Yes
NDFLS [32]	–	–	$< \infty$	No
Random Pursuit [60]	$\mathcal{O}(\log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon)$	4–7 (empirical)	Yes
ZOO-Newton [16]	–	–	3	No
Stochastic 3 Points [5]	$\mathcal{O}(\log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon)$	2	No
CARS (proposed)	$\mathcal{O}(\log(1/\varepsilon))$	$\mathcal{O}(1/\varepsilon)^{\dagger}$	3 or $4^{\dagger}$	No

We refer to algorithms without an agreed-upon name by the paper in which it first appeared. If a quantity (e.g. queries per iteration, convergence rate) is not explicitly computed we denote this with "–"
†: refers to CARS-CR variant

assuming access to an approximate line search oracle that solves (2a) sufficiently accurately, for any strongly convex f, as long as u_k are cyclically sampled coordinate vectors. Similar ideas can be found in [31–33]. More recently, [60] studied (2) under the name *Random Pursuit* which assumes access to an approximate line search oracle satisfying either additive ($\alpha_\star - \delta \leq \tilde{\alpha} \leq \alpha_\star + \delta$) or multiplicative ($(1-\delta)\alpha_\star \leq \tilde{\alpha} \leq \alpha_\star$ and $\text{sign}(\tilde{\alpha}) = \text{sign}(\alpha_\star)$) error bounds. They show Random Pursuit finds an ε-optimal solution in $\mathcal{O}(\log(1/\varepsilon))$ (resp. $\mathcal{O}(1/\varepsilon)$) iterations if f is strongly convex (resp. convex). See Table 1 for a summary of these complexity results. The use of $\mathcal{O}(\cdot)$ above suppresses the dependence of the query complexity on the dimension d. In all results stated, the query complexity scales at least linearly with d. This is unavoidable in DFO for generic f; see [2, 9–11, 14, 63] for recent progress in overcoming this, as well as [22] for work on improving query efficiency using prior information such as surrogate models.

We highlight a shortcoming of the aforementioned works: Although they provide essentially optimal bounds on the *iteration* complexity, they do not bound the *query* complexity. Indeed, the true query complexity will depend on the inner workings of the solver employed to solve (2a). For example, [60] reports each call to the line search oracle requires an average of 4 function queries when $d \leq 128$ which increases to 7 when $d = 1024$. In contrast, CARS requires *only three queries* per iteration, independent of d. The recently introduced Stochastic Three Point (STP) method [5, 8] uses only two queries per iteration. However, in practice we find the performance of STP compares poorly against CARS (see Sect. 5).

We are partially motivated by ZOO-Newton [16], which is essentially CARS with $\mathcal{D} = \text{Unif}(\{e_1, \ldots, e_d\})$. In [16], it is demonstrated empirically that ZOO-Newton performs well but no theoretical guarantees are provided. Our convergence guarantees for CARS imply convergence of ZOO-Newton as a special case. Many other works consider adapting Newton's method to the derivative-free setting. However, obtaining an estimate of the $d \times d$ Hessian $\nabla^2 f(x_k)$ for general (i.e. unstructured) $f(x)$ is difficult. Thus, one needs to either use $\Omega(d^2)$ queries [23] in order to obtain an accurate estimate of $\nabla^2 f(x_k)$—far too many for most applications—or use a high-variance approximation to $\nabla^2 f(x_k)$ [26, 58, 65–67]. CARS sidesteps this dichotomy, as it applies Newton's method to a one-dimensional function. Thus the "Hessian" to be estimated is 1×1. Finally, we note that CARS may be seen as a zeroth-order analogue of the Randomized Subspace Newton (RSN) [30]. Similarly, CARS-CR may be seen a zeroth-order analogue of the Stochastic Subspace Cubic Newton method [34]. Although the analysis of CARS (respectively CARS-CR) is inspired by that of RSN

(respectively SSCN), the use of finite difference approximations instead of exact derivatives provides non-trivial obstacles. We elaborate on this further in the text.

1.3 Main Contributions

We propose a simple and lightweight zeroth-order algorithm: CARS. To derive convergence rates for CARS we use a novel convergence analysis that hinges on the insight that CARS need only significantly decrease the objective function on a positive proportion of the iterates. Our results allow for a Hölder continuous Hessian—a weaker assumption than the Lipschitz continuity typically considered in such settings. We also propose a cubic-regularized variant, CARS-CR. The analysis of CARS-CR extends that of a special case of the Stochastic Subspace Newton method [34]—specifically: the case where the subspace in question is one-dimensional—to the zeroth-order setting. The key ingredient is a careful handling of the errors introduced by replacing directional derivatives with their finite difference counterparts. Our theoretical results are corroborated by rigorous benchmarking on two datasets: Moré–Garbow–Hillstrom [50] and CUTEst [29], which reveal that CARS outperforms existing line-search based ZOO algorithms. Our paper is accompanied by an open-source implementation of CARS (and CARS-CR), available online at https://github.com/bumsu-kim/CARS.

2 Curvature-Aware Random Search

Given u_k sampled from $\mathcal{D}$, consider the one-dimensional Taylor expansion:

$$T_2(\alpha; x_k, u_k) := f(x_k) + \alpha u_k^\top g_k + \frac{\alpha^2}{2} u_k^\top H_k u_k \approx f(x_k + \alpha u_k). \tag{3}$$

CARS selects $\alpha_k \approx \arg\min_\alpha T_2(\alpha; x_k, u_k)$. The exact minimizer $u_k^\top g_k / u_k^\top H_k u_k$ depends on unavailable quantities, so CARS uses $\alpha_k = d_{r_k}/h_{r_k}$, where d_r and h_r are finite difference approximations:

$$d_{r_k}(x_k; u_k) := \frac{f(x_k + r_k u_k) - f(x_k - r_k u_k)}{2r_k} = u_k^\top g_k + \mathcal{O}(r_k^{1+a}\|u_k\|^{2+a}), \tag{4}$$

$$h_{r_k}(x_k; u_k) := \frac{f(x_k + r_k u_k) - 2f(x_k) + f(x_k - r_k u_k)}{r^2} = u_k^\top H_k u_k + \mathcal{O}(r_k^{a}\|u_k\|^{2+a}). \tag{5}$$

(We write d_{r_k} and h_{r_k}, in place of $d_{r_k}(x_k; u_k)$ and $h_{r_k}(x_k; u_k)$ when x_k and u_k are clear from context.) Thus each iteration of CARS is a zeroth-order analogue of a single iteration of Newton's method applied to f restricted to the line spanned by u_k. As is well-known [54], pure Newton's method may not converge. So, following [30] we add a fixed step-size[2] $1/\hat{L}$ and define:

$$x_{\mathrm{CARS},k} = x_k - \frac{d_{r_k}}{\hat{L} h_{r_k}} u_k. \tag{6}$$

We allow the distribution $\mathcal{D}$ to be iteration dependent, i.e. u_k can be sampled from $\mathcal{D}_k$. In computing $d_{r_k}(x_k)$ and $h_{r_k}(x_k)$, CARS queries f at the symmetric points $x_k + r_k u_k$ and $x_k - r_k u_k$. Thus, at no extra cost we may incorporate STP [5] as a *safeguarding mechanism*

[2] Recall that $\hat{L} > 0$ by Lemma 1

[35] for CARS and choose the next iterate as

$$x_{k+1} = \arg\min\{f(x_{\mathrm{CARS},k}),\, f(x_k),\, f(x_k - r_k u_k),\, f(x_k + r_k u_k)\},$$

which ensures monotonicity: $f(x_0) \geq f(x_1) \geq f(x_2) \geq \cdots$. CARS requires two input parameters, $\hat{L}$ and C. Ideally, $\hat{L}$ should be the relative smoothness parameter (see Lemma 1), although CARS-CR (see Sect. 3) introduces a mechanism for selecting $\hat{L}$ adaptively. The selection of C is the subject of the next section.

Algorithm 1 **C**urvature-**A**ware **R**andom **S**earch (CARS)

1: **Input:** x_0: initial point; $\hat{L}$: relative smoothness parameter, C: scale-free sampling radius limit.
2: Get the oracle $f(x_0)$.
3: **for** $k = 0$ to K **do**
4: Sample u_k from $\mathcal{D}_k$.
5: Set $r_k \leq C/\|u_k\|$.
6: Evaluate and store $f(x_k \pm r_k u_k)$.
7: Compute d_{r_k} and h_{r_k} using (4) and (5).
8: Compute $x_{\mathrm{CARS},k} = x_k - \frac{d_{r_k}}{\hat{L} h_{r_k}} u_k$.
9: $x_{k+1} = \arg\min\{f(x_{\mathrm{CARS},k}),\, f(x_k),\, f(x_k - r_k u_k),\, f(x_k + r_k u_k)\}$.
10: **end for**
11: **Output:** x_K: estimated optimum point.

2.1 Convergence Guarantees

Before proceeding we list two necessary assumptions on $\mathcal{D}_k$. To describe the assumptions, introduce:

$$\eta(g, H; \mathcal{D}) = \mathbb{E}_{u\sim\mathcal{D}}\left[\frac{(u^\top g)^2}{(u^\top H u)(g^\top H^{-1} g)}\right]. \tag{7}$$

By Cauchy-Schwarz $\eta(g, H; \mathcal{D}) \leq 1$ for all g, $\mathcal{D}$, and positive definite H. This quantity is similar to, but distinct from, $\mu_{\mathcal{D}}$ introduced in [5]. We use η to measure the quality of the sampling distribution $\mathcal{D}$ with respect to the Newton vector $H^{-1}g$, and it is exactly 1 when all $u \sim \mathcal{D}$ are parallel to $H^{-1}g$. Our analysis assumes $\eta(g, H; \mathcal{D})$ is bounded away from zero, and this property holds for common choices of $\mathcal{D}$ as shown in Lemma 2. Since replacing $(r_k, \mathcal{D}_k)$ by $(\beta^{-1} r_k, \beta\mathcal{D}_k)$, for any $\beta > 0$, will not affect CARS, we use the *scale-free sampling radius*, $r_k\|u_k\|$, and define the following constants depending on the Hölder continuity of H:

$$C_{1,a} = \left(\frac{(a+1)(a+2)}{2^{1/2+a} L_a}\right)^{1/(1+a)} \quad \text{and} \quad C_{2,a} = \left(\frac{(a+1)(a+2)}{4(\sqrt{2}+1) L_a}\right)^{1/a}.$$

Our analysis requires us to define the following sampling radius limit, C, which also depends on the target accuracy ε and a free parameter $\gamma \in (0, 1]$:

$$C := \min\left\{C_{1,a}(\gamma\sqrt{2\mu\varepsilon})^{1/(1+a)}, C_{2,a}\mu^{1/a}\right\}. \tag{8}$$

CARS uses C to choose the sampling radius r_k *after* sampling u_k (see Line 6 of Algorithm 1). For instance, when H is Lipschitz continuous, this rule gives $r_k\|u_k\| = \mathcal{O}(\varepsilon^{1/4})$. Note that C is scale-invariant, i.e. replacing (f, ε) by $(\lambda f, \lambda\varepsilon)$ for any $\lambda > 0$ does not change C.

Theorem 1 (Expected descent of CARS) *Suppose f is μ-strongly convex and its Hessian H is a-Hölder continuous. Let ε be the target accuracy. Suppose further that $\eta(g_k, H_k; \mathcal{D}_k) \geq \eta_0 > 0$. For any $\gamma \in (0, 1]$ take C as in (8) and a sampling radius satisfying $0 < r_k \leq C/\|u_k\|$, and let $x_{\text{CARS},k}$ be as in (6). Let $\mathcal{A}_k$ denote the event:*

$$\gamma \|u_k\| \sqrt{2\mu\varepsilon} \leq |u_k^\top g_k|. \tag{9}$$

Then,

$$\mathbb{E}\left[f(x_{\text{CARS},k}) - f_\star \mid \mathcal{A}_k\right] \leq \left(1 - \eta_0 \frac{\hat{\mu}}{2\hat{L}}\right)(f(x_k) - f_\star). \tag{10}$$

In words, by choosing the sampling radius appropriately given $\|u_k\|$, and conditioning on u_k being "good enough" (i.e. $\mathcal{A}_k$ occurs), we obtain linear descent in expectation. The appropriate choice of γ is distribution dependent, and is discussed further in Corollary 1. The proof of Theorem 1 can be found in Sect. 4. Although $\mathcal{A}_k$ does not occur with probability 1, we show $\mathcal{A}_k$ occurs for a positive fraction of CARS iterations. When $\mathcal{A}_k$ does not occur, the safeguarding mechanism (Line 10 of Algorithm 1) still ensures monotonicity: $f(x_{k+1}) \leq f(x_k)$. This reveals the key idea behind CARS: *it exploits good search directions u_k when they arise yet is robust against poor search directions.* Carefully quantifying this intuition, we have:

Corollary 1 (Convergence of CARS) *Take the assumptions of Theorem 1. Suppose further that there exists $\gamma \in (0, 1]$ such that*

$$p_\gamma := \inf_{k \geq 0} \mathbb{P}_{u_k \sim \mathcal{D}_k}\left[|u_k^\top g_k| \geq \gamma \|u_k\| \|g_k\|\right] > 0 \tag{11}$$

for all $k \geq 0$, and use γ to define C in (8). Then, Algorithm 1 converges linearly. More specifically, for any

$$K \geq \frac{2\hat{L}}{\eta_0 p_\gamma \hat{\mu}} \log\left(\frac{f(x_0) - f_\star}{\varepsilon}\right),$$

we have $\mathbb{E}[f(x_K)] - f_\star \leq \varepsilon$.

The additional assumption on $\mathcal{D}_k$, i.e. the existence of γ, is very mild, and is discussed in Sect. 2.2.

2.2 Further Results on the Sampling Distribution

The speed of convergence of CARS depends crucially on the lower bounds η_0 and p_γ (see (7) and (11)). The following Lemma computes η_0 for several commonly used distributions. These can be compared with [5, Lemma 3.4] where similar bounds are computed for a quantity $\mu_{\mathcal{D}}$ defined such that

$$\frac{\mathbb{E}_{u \sim \mathcal{D}}\left[|u^\top g|\right]}{\|g\|} \geq \mu_{\mathcal{D}} \text{ for all } g \in \mathbb{R}^d.$$

Lemma 2 *1. (Isotropic distributions) When*

$$\mathcal{D} = \text{Unif}(\mathbb{S}^{d-1}), \quad \text{Unif}(\{e_1, \cdots, e_d\}), \quad \mathcal{N}(0, I_d), \quad \text{or} \quad \text{Unif}(\{\pm 1\}^d),$$

we have $\eta(g, H; \mathcal{D}) \geq \mu/(dL)$. The distributions in the above equation are uniform on sphere, coordinate directions, Gaussian, and Rademacher, respectively.

2. *(Approximate gradient direction) If $\mathcal{D}$ satisfies*

$$\mathbb{E}_{u\sim\mathcal{D}}\left[\left(\frac{|u^\top g|}{\|u\|\|g\|}\right)^2\right] \geq \beta > 0 \tag{12}$$

for some $\beta > 0$, then $\eta(g, H; \mathcal{D}) \geq \beta\mu/L$.
3. *(Newton direction) When u is parallel to $H^{-1}g$ with probability 1, we have $\eta(g, H; \mathcal{D}) = 1$.*

Proof Since $u^\top Hu \leq L\|u\|^2$ and $g^\top H^{-1}g \leq \mu^{-1}\|g\|^2$,

$$\eta(g, H; \mathcal{D}) \geq \frac{\mu}{L}\,\mathbb{E}_{u\sim\mathcal{D}}\left[\left(\frac{|u^\top g|}{\|u\|\|g\|}\right)^2\right]. \tag{13}$$

1. When $\mathcal{D} = \mathcal{N}(0, I_d)$ or $\mathrm{Unif}(\mathbb{S}^{d-1})$, we can replace g by the standard basis vector e_1 by symmetry, and it immediately follows that $\eta(g, H; \mathcal{D}) \geq \mu/(dL)$. When $\mathcal{D} = \mathrm{Unif}(\{e_1, \cdots, e_d\})$,

$$\mathbb{E}_{u\sim\mathcal{D}}\left[\left(\frac{|u^\top g|}{\|u\|\|g\|}\right)^2\right] = \frac{1}{d}\sum_{i=1}^{d}|g_i|^2/\|g\|^2 = \frac{1}{d}$$

and when $\mathcal{D} = \mathrm{Unif}(\{\pm 1\}^d)$,

$$\mathbb{E}_{u\sim\mathcal{D}}\left[\left(\frac{|u^\top g|}{\|u\|\|g\|}\right)^2\right] = \frac{1}{2^d}\sum_{u\in\{\pm1\}^d}\frac{\sum_{i=1}^{d}|g_i|^2 + \sum_{i\neq j}u_iu_jg_ig_j}{d\|g\|^2} = \frac{1}{d}.$$

Hence, again from (13), we have the same lower bound $\mu/(dL)$.
2. When (12) holds, (13) provides the lower bound $\eta(g, H; \mathcal{D}) \geq \beta\mu/L$. In particular, when u is parallel to g (i.e. *gradient direction*) with probability p, then $\eta \geq p\mu/L$.
3. When u is the Newton direction, i.e. u is parallel to $H^{-1}g$ with probability 1, $u^\top g = u^\top Hu = g^\top H^{-1}g$, and so $\eta(g, H; \mathcal{D}) = 1$.

This finishes the proof. □

Lemma 2 suggests that assuming $\eta(g_k, H_k; \mathcal{D}_k) \geq \eta_0 > 0$ for all $k \geq 0$ is reasonable in practice. Note Case 3 yields the best possible η, as $\eta \leq 1$ by Cauchy-Schwarz. The next lemma suggests that choosing $\gamma = \mathcal{O}(1/\sqrt{d})$ is appropriate for several common distributions.

Lemma 3 (Lower Bounds of p_γ for Various Distributions)

1. *(Uniform on sphere and Gaussian) If $\mathcal{D} = \mathcal{N}(0, I_d)$ or $\mathrm{Unif}(\mathbb{S}^{d-1})$ and $\gamma = \frac{1}{\sqrt{d}}$ then $p_\gamma \geq 0.315603$.*
2. *(Random coordinate direction) If $\mathcal{D} = \mathrm{Unif}(\{e_1, \cdots, e_d\})$ and $\gamma = 1/\sqrt{d}$ then $p_\gamma = 1/d$.*

Proof 1. First note that the definition of p_γ is homogeneous with respect to u, so we may assume $\|u\| = 1$, and thus we only need to consider the case $\mathcal{D} = \mathrm{Unif}(\mathbb{S}^{d-1})$. In this case, $\mathcal{D}$ is invariant under rotation so we can take $g = e_1$ and

$$\mathbb{P}_{u\sim\mathcal{D}}\left[|u^\top g| \geq \gamma\|u\|\|g\|\right] = \mathbb{P}[|u_1| \geq \gamma] = I_{1-\gamma^2}\left(\frac{d-1}{2}, \frac{1}{2}\right)$$

where I is the regularized incomplete Beta function as in [10, Theorem 2.3]. In particular, when $\gamma = 1/\sqrt{d}$, the function $d \mapsto I_{1-1/d}\left(\frac{d-1}{2}, \frac{1}{2}\right)$ is decreasing for $d \geq 2$ and bounded below by 0. Thus $p_\gamma \geq \lim_{d\to\infty} I_{1-1/d}\left(\frac{d-1}{2}, \frac{1}{2}\right) = 0.315603\cdots$.

2. When $\mathcal{D} = \text{Unif}\{e_1, \ldots, e_d\}$,

$$\mathbb{P}_{u\sim\mathcal{D}}\left[|u^\top g| \geq \gamma \|u\|\|g\|\right] = \frac{1}{d}\sum_{i=1}^{d} 1_{|g_i|\geq\gamma\|g\|}.$$

Recall that $\|g\|^2 = \sum_i |g_i|^2$. Hence, we have $\max_i |g_i| \geq \|g\|/\sqrt{d}$, which implies $\mathbb{P}_{u\sim\mathcal{D}}\left[|u^\top g| \geq \|u\|\|g\|/\sqrt{d}\right] \geq 1/d$. Note that this bound is tight; the equality holds when, for example, $g = (1, 0, \cdots, 0)$.

This finishes the proof. □

Lemma 3 shows that choosing $\mathcal{D} = \text{Unif}(\{e_1, \ldots, e_d\})$ with $\gamma = 1/\sqrt{d}$ will lead to $p_\gamma = 1/d$. As a result, the query complexity will have a quadratic dependence on the dimension d, which is worse compared to other distributions such as Gaussian and uniform on the sphere. This worse dependence on d when $\mathcal{D} = \text{Unif}(\{e_1, \ldots, e_d\})$ also manifests in other DFO studies, for instance, [5, Theorem 6.2].

2.3 Combining CARS with Gradient Estimators

When γ is small enough and $\mathcal{D}_k$ approximates the gradient or Newton direction close enough, both $\eta_{\mathcal{D}_k}$ and p_γ do not depend on d, leading to dimension independent convergence rates. So, CARS can be combined with other derivative-free techniques that estimate the gradient (or Newton direction)—at the cost of three additional function queries per iteration CARS will choose an approximately optimal step-size in this computed direction. Our analysis easily extends to such combined methods, and we sketch how to do so for the widely used [19, 24, 55, 56] variance-reduced Nesterov-Spokoiny gradient estimate:

$$\tilde{g}_k := \frac{1}{m}\sum_{i=1}^{m} d_r(x_k; u_{k,i})u_{k,i} \approx g_k. \tag{14}$$

For simplicity, we assume access to exact directional derivatives (as in [55]).

Corollary 2 *Let f be μ-strongly convex and H be a-Hölder continuous. Suppose, at each step, u_k is generated by first sampling $v_1, \ldots, v_m$ from Gaussian distribution $\mathcal{N}(0, I_d)$ and defining:*

$$u_k = \frac{1}{m}\sum_{j=1}^{m}(g_k^\top v_j)v_j.$$

Then $\eta(g_k, H_k; \mathcal{D}) \geq \frac{m}{m+d+1}\frac{\mu}{L}$.

Remark 2 Compared to a single-query algorithm, which guarantees $\eta(g_k, H_k; \mathcal{D}) \geq \frac{\mu}{dL}$ as in Lemma 2, the upper bound of the number of iterations is decreased by a factor of $\frac{1}{m} + \frac{1}{d} + \frac{1}{md}$. This shows a parallel querying strategy can achieve nearly linear speed up when $d \gg m$.

Proof Without loss of generality, assume $g_k = e_1 = (1, 0, \ldots, 0)$. Denoting the i-th component of v_j by $v_{j,i}$, we have

$$\eta_{\mathcal{D}} \geq \frac{\mu}{L}\,\mathbb{E}\left[\frac{\left(\sum_{j=1}^m v_{j,1}^2\right)^2}{\left(\sum_{j=1}^m v_{j,1}^2\right)^2 + \sum_{i=2}^d\left(\sum_{j=1}^m v_{j,1}v_{j,i}\right)^2}\right].$$

from (13). Let X^2 and Z^2 denote the numerator and the denominator in the expectation above, respectively. By the Cauchy-Schwarz inequality,

$$\mathbb{E}\left[\frac{X^2}{Z^2}\right]\mathbb{E}[Z^2] \geq \mathbb{E}\left[\frac{X}{Z}Z\right]^2 = \mathbb{E}[X]^2$$

and thus

$$\eta_{\mathcal{D}} \geq \frac{\mu}{L}\frac{\mathbb{E}[X]^2}{\mathbb{E}[Z^2]} = \frac{\mu}{L}\frac{m^2}{m^2+2m+(d-1)m} = \frac{\mu m}{L(m+d+1)}.$$

Combining this with Corollary 1 yields the claim. □

3 CARS with Cubic Regularization for General Convex Functions

Here, we adopt cubic regularization [34, 54], a technique to achieve global convergence of a second-order method for convex functions, in CARS and prove convergence. We drop strong convexity and assume only convexity and L-smoothness. We assume Lipschitz continuity of the Hessian (i.e. $a = 1$ in Assumption 1) and let $M = L_1$ be the Lipschitz constant. Instead of (3), we now use

$$P(\alpha; d, h) := d\alpha + \frac{1}{2}h\alpha^2 + \frac{M}{6}|\alpha|^3, \tag{15}$$

with the exact derivatives $P(\cdot; d_0, h_0)$ and the finite difference approximations $P(\cdot; \pm d_{r_k}, h_{r_k})$. The method of Stochastic Subspace Cubic Newton (SSCN) [34], specialized to the case of one-dimensional subspaces, takes exact derivatives and uses the following inequality [34, Lemma 2.3]

$$f(x_k + \alpha u_k) \leq f(x_k) + P(\alpha; d_0(x_k; u_k), h_0(x_k; u_k)) \tag{16}$$

to derive the algorithm $x_{k+1} = x_k + \hat{\alpha}_k u_k$, where $\hat{\alpha}_k = \arg\min_\alpha P(\alpha; d_0, h_0)$. We propose using $\alpha_k^{\pm} = \arg\min_\alpha P(\alpha; \pm d_{r_k}, h_{r_k})$ in place of $\hat{\alpha}_k$. By solving $P'(\alpha; \pm d_{r_k}, h_{r_k}) = 0$ we obtain

$$\alpha_k^{\pm} = -\frac{\pm 2 d_{r_k}}{h_{r_k} + \sqrt{h_{r_k}^2 + 2M|d_{r_k}|}}.$$

This step-size equals $-\frac{\pm d_{r_k}}{h_{r_k}\hat{L}_k}$ with

$$\hat{L}_k = \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{M|d_{r_k}|}{2h_{r_k}^2}}, \tag{17}$$

so it is just CARS with an adaptive relative smoothness constant. We formalize this as Algorithm 2.

To analyze CARS-CR (Algorithm 2), we make a boundedness assumption.

Definition 3 Recall that $\mathcal{Q} = \{x \in \mathbb{R}^d : f(x) \leq f(x_0)\}$. We say f has an $\mathcal{R}$*-bounded level set* if the diameter of $\mathcal{Q}$ is $\mathcal{R} < \infty$.

Without loss of generality, we may assume the distribution is normalized (i.e. $\|u\| = 1$ w.p. 1.) This is because we only need to bound the scale-free sampling radius $r_k\|u_k\|$, as before. To ensure that the finite difference error is insignificant, we need the sampling radius

Algorithm 2 CARS with **C**ubic **R**egularization (CARS-CR)

1: **Input:** ε: target accuracy; x_0: initial point; r_0: initial sampling radius; M: Lipschitz constant of Hessian.
2: Get the oracle $f(x_0)$.
3: **for** $k = 0$ to K **do**
4: Sample u_k from $\mathcal{D}_k$.
5: Set $r_k \leq \rho\sqrt{\varepsilon}/\sqrt{k+2}$ where $\rho = R/\sqrt{2B}$ as defined in Theorem 3.
6: Evaluate and store $f(x_k \pm r_k u_k)$.
7: Compute d_{r_k} and h_{r_k} using (4) and (5).
8: Compute $\hat{L}_k$ using (17).
9: Compute $x_{\mathrm{CR}\pm,k} = x_k \pm \frac{d_{r_k}}{\hat{L}_k h_{r_k}} u_k$.
10: $x_{k+1} = \arg\min\{f(x_{\mathrm{CR}+,k}), f(x_{\mathrm{CR}-,k}), f(x_k), f(x_k - r_k u_k), f(x_k + r_k u_k)\}$.
11: **end for**
12: **Output:** x_K: estimated optimum point.

small enough. However, for a more concise analysis, it is helpful to have an upper bound, which can be chosen arbitrarily. Let $R > 0$ be an upper bound of r_k for all $k \geq 0$. Note that any r_k selected by CARS-CR automatically satisfies $r_k \leq R$ (see line 5 of Algorithm 2). Using this notation, we get:

Lemma 4 (Finite difference error bound for the minimum of P) *Let $P(\cdot) = P(\cdot; d_0, h_0)$. Then for any $0 \leq r_k \leq R$,*

$$\min(|P(\hat{\alpha}_k) - P(\alpha_k^+)|, |P(\hat{\alpha}_k) - P(\alpha_k^-)|) \leq \frac{2B}{R^2} r_k^2, \tag{18}$$

where $B = \max(LR^2, MR^3, f(x_0) - f_\star)$.

If the sampling distribution is isotropic in expectation, i.e. it satisfies $\mathbb{E}\left[u_k u_k^\top\right] = \frac{1}{d} I_d$,, we get the following descent lemma:

Theorem 2 (Expected descent of CARS-CR) *Suppose f is convex, L-smooth, and has M-Lipschitz Hessian. If $\mathcal{D}_k$ is isotropic in expectation, then with Algorithm 2, we have*

$$\mathbb{E}\left[f(x_{k+1}) \mid x_k\right] \leq \left(1 - \frac{1}{d}\right) f(x_k) + \frac{1}{d} f(x_k + z) + \frac{L}{2d}\|z\|^2 + \frac{M}{6d}\|z\|^3 + \frac{2B}{R^2} r_k^2 \tag{19}$$

for any $z \in \mathbb{R}^d$.

A similar expected descent bound for SSCN is given in [34, Lemma 5.7]. However, this bound does not contain the term $\frac{2B}{R^2} r_k^2$. This is to be expected, as this term results from finite differencing, and SSCN uses exact derivative information. By combining Theorem 2 with a decreasing r_k (as in Algorithm 2), we obtain the $\mathcal{O}(k^{-1})$ convergence rate for CARS-CR.

Theorem 3 (Convergence of CARS-CR) *Take the assumptions of Theorem 2, and further assume f has an $\mathcal{R}$-bounded level set. Set $r_k \leq \frac{\rho\sqrt{\varepsilon}}{\sqrt{k+2}}$ where $\rho = \frac{R}{\sqrt{2B}}$. Let x_K be the output of Algorithm 2. Then $\mathbb{E}[f(x_K)] - f_\star \leq \varepsilon$ if*

$$K \geq \max\left\{16L\mathcal{R}^2\frac{d}{\varepsilon}, \sqrt{32M\mathcal{R}^3}\frac{d}{\sqrt{\varepsilon}}, (1024(f(x_0) - f_\star))^{1/4}\frac{d^{5/4}}{\varepsilon^{1/4}}, 50\right\} \tag{20}$$

Clearly, the dominating term is $16L\mathcal{R}^2\frac{d}{\varepsilon}$. This theorem is directly comparable to [5, Theorem 5.5] and [34, Theorem 5.8]. In [34, Theorem 5.8] it is shown that SSCN, specialized to the

case of one-dimensional subspaces, returns an x_K satisfying $\mathbb{E}[f(x_K)] - f_\star \leq \varepsilon$ for $K \geq 4.5L\mathcal{R}^2\frac{d}{\varepsilon}$ plus lower order terms. We note that SSCN requires exact gradient information. In [5, Theorem 5.5] it is shown that, under similar—but not identical—assumptions, STP returns an x_K satisfying $\mathbb{E}[f(x_K)] - f_\star \leq \varepsilon$ for $K \geq 4\,L\mathcal{R}^2\left(\frac{d}{\varepsilon} - \frac{1}{f(x_0)-f_\star}\right)$. Although our rate is worse (by a constant factor of 4) our rule for selecting r_k is more practical than the rule for selecting α_k—the analogous parameter—given in [5, Theorem 5.5] which depends on $\mathbb{E}[f(x_{k-1})] - f_\star$. Moreover, we have not attempted to optimize our parameters. Finally, and most importantly, we note that CARS-CR consistently outperforms STP in practice.

4 Proofs

Here we collect the proofs of the results of Sects. 2.1 and 3, and state and prove some auxiliary lemmas needed in the proofs of the main results. We begin with a lemma quantifying the expected descent given access to exact derivatives.

4.1 Proofs for Results in Sect. 2.1

Lemma 5 (Expected descent of CARS with exact derivatives) *Let $u_k \sim \mathcal{D}_k$ and $x_{\mathrm{ED},k}$ be the CARS step with exact derivatives*

$$x_{\mathrm{ED},k} = x_k - \frac{u_k^\top g_k}{\hat{L} u_k^\top H_k u_k} u_k. \tag{21}$$

Then letting $\eta_k = \eta(g_k, H_k; \mathcal{D}_k)$,

$$\mathbb{E}\left[f(x_{\mathrm{ED},k}) \mid x_k\right] - f_\star \leq \left(1 - \eta_k \frac{\hat{\mu}}{\hat{L}}\right)(f(x_k) - f_\star). \tag{22}$$

Remark 3 Lemma 5 is similar to [30, Corollary 1] and [42, Corollary 1 part (ii)]. However, Lemma 5 allows for more general sampling distributions $\mathcal{D}$.

Proof From $\hat{\mu}$-relative strong convexity we have

$$f_\star - f(x_k) \geq \langle g_k, x_\star - x_k\rangle + \frac{\hat{\mu}}{2}\|x_\star - x_k\|_{H_k}^2 \geq -\frac{1}{2\hat{\mu}}\|g_k\|_{H_k^{-1}}^2, \tag{23}$$

where the second inequality follows by taking $x = x_\star - x_k$ and $c = \hat{\mu}$ in the following general inequality [30, Lemma 9]:

$$\arg\min_{x\in\mathbb{R}^d} \langle g, x\rangle + \frac{c}{2}\|x\|_H^2 = -\frac{1}{c}H^{-1}g \quad \text{if } H \succ 0 \text{ and } c > 0.$$

Rearranging (23) yields $-\|g_k\|_{H_k^{-1}}^2 \leq 2\hat{\mu}(f_\star - f(x_k))$. Let $M_k := \frac{u_k u_k^\top}{u_k^\top H_k u_k}$. Then, from $\hat{L}$-relative smoothness and [30, Lemma 5],

$$f(x_{\mathrm{ED},k}) \leq f(x_k) - \frac{1}{2\hat{L}}\|g_k\|_{M_k}^2 = f(x_k) - \frac{1}{2\hat{L}}\frac{\langle u_k u_k^\top g_k, g_k\rangle}{u_k^\top H_k u_k} = f(x_k) - \frac{1}{2\hat{L}}\frac{(u_k^\top g_k)^2}{u_k^\top H_k u_k}. \tag{24}$$

Now let $\mathbb{E}_k[\cdot] := \mathbb{E}[\cdot|x_k]$ and take the conditional expectation of both sides of (24):

$$\begin{aligned}\mathbb{E}_k\left[f(x_{\mathrm{ED}})\right] &\le f(x_k) - \frac{1}{2\hat{L}}\mathbb{E}_k\left[\frac{(u_k^\top g_k)^2}{u_k^\top H_k u_k}\right] \\ &= f(x_k) - \frac{\eta(g_k, H_k; \mathcal{D}_k)}{2\hat{L}}\|g_k\|^2_{H_k^{-1}} \\ &\le f(x_k) - \eta_k\frac{\hat{\mu}}{\hat{L}}(f(x_k) - f_\star)\end{aligned}$$

Subtracting $f_\star$ from both sides yields the desired result. □

Proof (of Theorem 1) In this proof, for notational convenience let $d_0 = g_k^\top u_k$ for the first-order directional derivative, and $h_0 = u_k^\top H_k u_k$ for the second-order, and denote r_k by r. From the definition of $\hat{L}$-relative smoothness, how much we progress at each step can easily be described by a quadratic function $q(t)$:

$$f(x_k) - f(x_k + tu_k) \ge q(t) := -d_0 t - \frac{1}{2}\hat{L}h_0 t^2.$$

As in the exact derivatives case, the maximizer of q is $t_\star = -d_0/(\hat{L}h_0)$, with corresponding maximum $q(t_\star) = d_0^2/(2\hat{L}h_0) = \|g_k\|_{M_k}/(2\hat{L})$, where $M_k := \frac{u_k u_k^\top}{u_k^\top H_k u_k}$ as before. Recall that $x_{\mathrm{CARS},k} = x_k - d_r/(\hat{L}h_r)u_k$. Our goal is to show that the finite difference estimate $t_r := -d_r/(\hat{L}h_r)$ approximates $t_\star$ well enough so that $q(t_r) \ge q(t_\star)/2$. Observe that if

$$|t_r/t_\star - 1| \le \sqrt{1-c} \iff |t_r - t_\star|^2 \le (1-c)t_\star^2 \tag{25}$$

holds for some $0 < c < 1$, then by completing the square in $q(t)$:

$$q(t_r) = -\frac{\hat{L}h_0}{2}(t_r - t_\star)^2 + q(t_\star) \ge -(1-c)q(t_\star) + q(t_\star) = cq(t_\star).$$

Because we want to show $q(t_r) \ge q(t_\star)/2$, it suffices to show (25) holds for $c = 1/2$, i.e.,

$$\left|\frac{t_r}{t_\star} - 1\right| = \left|\frac{d_r/d_0}{h_r/h_0} - 1\right| \le \sqrt{1 - \frac{1}{2}} = \frac{1}{\sqrt{2}}. \tag{26}$$

To prove (26), we further bound the left-hand side by the two separate (relative) finite difference errors. Let e_d and e_h be the absolute errors in estimating d_0 and h_0, respectively, i.e. $e_d = |d_0 - d_r|$ and $e_h = |h_0 - h_r|$. Then, when $e_h < h_0$, which will be shown shortly,

$$\left|\frac{d_r/d_0}{h_r/h_0} - 1\right| = \left|\frac{-\frac{d_0-d_r}{d_0} + \frac{h_0-h_r}{h_0}}{1 - \frac{h_0-h_r}{h_0}}\right| \le \frac{e_d/|d_0| + e_h/h_0}{1 - e_h/h_0},$$

and thus, for (26) we only need to prove

$$\frac{e_d}{|d_0|} + \left(1 + \frac{1}{\sqrt{2}}\right)\frac{e_h}{h_0} \le \frac{1}{\sqrt{2}}. \tag{27}$$

Now we bound e_d and e_h using Taylor's theorem and Assumption 1. Because we have

$$f(x_k \pm ru_k) = f(x_k) \pm rg_k^\top u_k + r^2\int_0^1 (1-t)u_k^\top H(x_k \pm tru_k)u_k\, dt, \tag{28}$$

we get the following representation for the error of the first-order directional derivative:

$$\begin{aligned} d_r - d_0 &= \frac{f(x_k + ru_k) - f(x_k - ru_k)}{2r} - g_k^\top u_k \\ &= \frac{r}{2}\int_0^1 (1-t) u_k^\top \left[H(x_k + tru_k) - H(x_k - tru_k)\right] u_k \, dt. \end{aligned}$$

By Assumption 1, $\left|u_k^\top \left[H(x_k + tru_k) - H(x_k - tru_k)\right] u_k\right| \le L_a (2tr)^a \|u_k\|^{a+2}$ and therefore,

$$e_d = |d_r - d_0| \le 2^{a-1} L_a r^{a+1} \|u_k\|^{a+2} \int_0^1 (1-t) t^a \, dt = \left(\frac{r\|u_k\|}{C_{1,a}}\right)^{1+a} \frac{\|u_k\|}{2\sqrt{2}}. \tag{29}$$

Similarly, for the second-order directional derivative,

$$e_h = |h_r - h_0| \le 2L_a r^a \|u_k\|^{a+2} \int_0^1 (1-t) t^a \, dt = \left(\frac{r\|u_k\|}{C_{2,a}}\right)^{a} \frac{\|u_k\|^2}{2\sqrt{2}+2} \tag{30}$$

We see that $r\|u_k\| \le C = \min\{C_{1,a}(\gamma\sqrt{2\mu\varepsilon})^{1/(1+a)}, C_{2,a}\mu^{1/a}\}$ implies two separate bounds

$$e_d \le \frac{\gamma\sqrt{\mu\varepsilon}\|u_k\|}{2} \overset{(a)}{\le} \frac{|d_0|}{2\sqrt{2}} \quad \text{and} \quad e_h \le \frac{\mu\|u_k\|^2}{2\sqrt{2}+2} \overset{(b)}{\le} \frac{h_0}{2\sqrt{2}+2}, \tag{31}$$

where (a) holds assuming $\mathcal{A}_k$ occurs and (b) follows from strong convexity:

$$h_0 = u_k^\top H_k u_k \ge \mu. \tag{32}$$

As (31) implies (27) we have proved the theorem. □

We now are ready to prove the convergence of CARS (Algorithm 1).

Proof (of Corollary 1) From strong convexity we have

$$f_\star - f(x) \ge \langle g(x), x_\star - x\rangle + \frac{\mu}{2}\|x_\star - x\|^2 \ge -\frac{1}{2\mu}\|g(x)\|^2,$$

for any $x \in \mathbb{R}^d$, where the second inequality comes from

$$\arg\min_{x\in\mathbb{R}^d} \langle g, x\rangle + \frac{c}{2}\|x\|^2 = -\frac{1}{c}g.$$

Thus $\|g(x)\|^2 \ge 2\mu(f(x) - f_\star)$. Taking expectation on both sides $\mathbb{E}[\|g(x_k)\|^2] \ge 2\mu(\mathbb{E}[f(x_k)] - f_\star)$.

If $\|g(x_k)\|^2 \le 2\mu\varepsilon$ at the k-th step with $k \le K$, then $f(x_K) - f_\star \le \varepsilon$ as $f(x_k)$ is monotonically decreasing by definition (See line 9 of Algorithm 1.) Thus we need only consider the case where $\|g(x_k)\|^2 > 2\mu\varepsilon$ for all $k < K$; because if the expectation of $f(x_K)$ conditioned on this event is less than or equal to $f_\star + \varepsilon$, then the total expectation is also bounded by the same value.

The key of the proof is that $\mathcal{A}_k$ occurs with probability at least $p_\gamma > 0$. Indeed, we have $|u_k^\top g_k| \ge \gamma\|u_k\|\|g_k\|$ with probability at least p_γ, and since $\|g_k\| > \sqrt{2\mu\varepsilon}$,

$$\mathbb{P}[\mathcal{A}_k] \ge \mathbb{P}\left[|u_k^\top g_k| \ge \gamma\|u_k\|\|g_k\| \ge \gamma\|u_k\|\sqrt{2\mu\varepsilon}\right] \ge p_\gamma.$$

If $\mathcal{A}_k$ occurs then by Theorem 1, we get

$$\mathbb{E}[f(x_{k+1})|\mathcal{A}_k] - f_\star \le \left(1 - \eta_{\mathcal{D}}\frac{\hat{\mu}}{2\hat{L}}\right)(f(x_k) - f_\star).$$

If $\mathcal{A}_k$ does not occur then, as CARS is non-increasing, $f(x_{k+1}) \leq f(x_k)$. Thus

$$\begin{aligned}
\mathbb{E}\left[f(x_{k+1}) \mid x_k\right] - f_\star &= \mathbb{E}[f(x_{k+1}) - f_\star | \mathcal{A}_k]\mathbb{P}[\mathcal{A}_k] + \mathbb{E}[f(x_{k+1}) - f_\star | \mathcal{A}_k^c]\mathbb{P}[\mathcal{A}_k^c] \\
&\leq \left(1 - \eta_{\mathcal{D}} \frac{\hat{\mu}}{2\hat{L}}\right)(f(x_k) - f_\star)\mathbb{P}[\mathcal{A}_k] + (f(x_k) - f_\star)(1 - \mathbb{P}[\mathcal{A}_k]) \\
&= \left(1 - \eta_{\mathcal{D}} \mathbb{P}[\mathcal{A}_k] \frac{\hat{\mu}}{2\hat{L}}\right)(f(x_k) - f_\star) \\
&\leq \left(1 - \eta_{\mathcal{D}} p_\gamma \frac{\hat{\mu}}{2\hat{L}}\right)(f(x_k) - f_\star) \\
\Rightarrow \mathbb{E}[f(x_{k+1})] - f_\star &\leq \left(1 - \eta_{\mathcal{D}} p_\gamma \frac{\hat{\mu}}{2\hat{L}}\right)^{k+1} (f(x_0) - f_\star),
\end{aligned}$$

whence solving for K in

$$\left(1 - \eta_{\mathcal{D}} p_\gamma \frac{\hat{\mu}}{2\hat{L}}\right)^{K} (f(x_0) - f_\star) \leq \varepsilon \tag{33}$$

completes the proof. □

4.2 Proofs for Results in Sect. 3

Recall that:

$$P(\alpha; d, h) := d\alpha + \frac{1}{2}h\alpha^2 + \frac{M}{6}|\alpha|^3$$

(we write $P(\alpha)$ in place of $P(\alpha; d, h)$ when d and h are clear from context.) Define the map $\phi : \mathbb{R} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$:

$$\phi(d, h) := \arg\min_{\alpha} P(\alpha; d, h).$$

Note that not only $h_0 \geq 0$, but also $h_{r_k} \geq 0$ due to the convexity of f:

$$h_{r_k}(x_k; u_k) = \frac{2}{r_k^2}\left(\frac{f(x_k + r_k u_k) + f(x_k - r_k u_k)}{2} - f(x_k)\right) \geq 0.$$

Then $\hat{\alpha}_k = \phi(d_0, h_0)$ and $\alpha_k^{\pm} = \phi(\pm d_{r_k}, h_{r_k})$ by their definition. Along the way, we have useful identities for ϕ:

$$\phi(d, h) = \frac{\operatorname{sign}(d)}{M}\left(h - \sqrt{h^2 + 2M|d|}\right) = \frac{-2d}{h + \sqrt{h^2 + 2M|d|}}, \tag{34}$$

and

$$\frac{M}{2}|\phi(d, h)|\phi(d, h) = -d - h\phi(d, h). \tag{35}$$

Note that (34) shows that ϕ is well-defined. We first describe the perturbation of ϕ, and how P behaves near its minimum.

Lemma 6 (Perturbation of ϕ) *Let $d, d' \in \mathbb{R}$ have the same sign and $h, h' \geq 0$. Defining $S = \sqrt{h^2 + 2M|d|}$ and $S' = \sqrt{(h')^2 + 2M|d'|}$,*

$$|\phi(d, h) - \phi(d', h')| \leq \frac{|h - h'|}{M} + \frac{2|d - d'|}{S + S'}. \tag{36}$$

Proof Because d and d' have the same sign, from (34), we obtain that $\phi(d,h)$ and $\phi(d',h')$ have the same sign and so $|\phi(d,h)-\phi(d',h')| = \frac{1}{M}|S-S'-(h-h')|$, whence

$$\begin{aligned}|\phi(d,h)-\phi(d',h')| &= \frac{1}{M}\left|\left(S-S'\right)\frac{S+S'}{S+S'}-(h-h')\right| = \frac{1}{M}\left|\frac{S^2-(S')^2}{S+S'}-(h-h')\right| \\ &= \frac{1}{M}\left|\frac{(h-h')(h+h')}{S+S'}+\frac{2M(|d|-|d'|)}{S+S'}-(h-h')\right| \\ &\le \frac{1}{M}\left(1-\frac{h+h'}{S+S'}\right)|h-h'|+\frac{2|d-d'|}{S+S'} \quad \le \frac{|h-h'|}{M}+\frac{2|d-d'|}{S+S'},\end{aligned}$$

where the last inequality comes from that $0 \le h+h' \le S+S'$. □

We now analyze the effect of perturbations to $\alpha_{\min}$ on $P(\alpha)$, under the assumption that the perturbed value of α has the same sign as $\alpha_{\min}$.

Lemma 7 (Perturbation of $P(\alpha)$ near minimum) *Let $d \in \mathbb{R}$ and $h \ge 0$. Define $\alpha_{\min} = \phi(d,h)$, and let $\alpha' \in \mathbb{R}$ have $\operatorname{sign}(\alpha') = \operatorname{sign}(\alpha_{\min})$. Then*

$$0 \le P(\alpha';d,h)-P(\alpha_{\min};d,h) \le \frac{1}{2}(\alpha_{\min}-\alpha')^2(h+M|\alpha_{\min}|+\frac{M}{3}|\alpha_{\min}-\alpha'|). \quad (37)$$

Proof Let $\sigma = \operatorname{sign}(\alpha_{\min}) = \operatorname{sign}(\alpha')$. We write $P(\alpha_{\min})$, resp. $P(\alpha)$, for $P(\alpha_{\min};d,h)$, resp. $P(\alpha';d,h)$. Then,

$$\begin{aligned}&P(\alpha')-P(\alpha_{\min}) \\ &= d(\alpha'-\alpha_{\min})+\frac{h}{2}(\alpha'-\alpha_{\min})(\alpha'+\alpha_{\min})+\frac{\sigma M}{6}(\alpha'-\alpha_{\min})((\alpha')^2+\alpha_{\min}^2+\alpha_{\min}\alpha') \\ &= (\alpha'-\alpha_{\min})\left(d+\frac{h}{2}(\alpha'+\alpha_{\min})+\frac{\sigma M}{6}((\alpha')^2+\alpha_{\min}^2+\alpha_{\min}\alpha')\right).\end{aligned}$$

Using (35), we get

$$\begin{aligned}P(\alpha')-P(\alpha_{\min}) &= (\alpha'-\alpha_{\min})\left(\frac{h}{2}(\alpha'-\alpha_{\min})+\frac{\sigma M}{6}((\alpha')^2-2\alpha_{\min}^2+\alpha_{\min}\alpha')\right) \\ &= \frac{1}{2}(\alpha'-\alpha_{\min})^2\left(h+\frac{M}{3}|\alpha'+2\alpha_{\min}|\right) \\ &\le \frac{1}{2}(\alpha'-\alpha_{\min})^2\left(h+M|\alpha_{\min}|+\frac{M}{3}|\alpha'-\alpha_{\min}|\right).\end{aligned}$$

Noting that $P(\alpha')-P(\alpha_{\min}) \ge 0$ as $\alpha_{\min}$ minimizes $P(\alpha)$ we obtain the desired statement. □

From (34) we see that if $\operatorname{sign}(d_{r_k}) = \operatorname{sign}(d_0)$ then $\operatorname{sign}(\hat{\alpha}_k) = \operatorname{sign}(\alpha_k^+)$, whence we may use the perturbation bounds of Lemmas 6 and 7. If $\operatorname{sign}(d_{r_k}) = -\operatorname{sign}(d_0)$ then $\operatorname{sign}(\hat{\alpha}_k) = \operatorname{sign}(\alpha_k^-)$ and the conclusions of Lemmas 6 and 7 still apply. We conclude that at least one of α_k^+ and α_k^- is a good approximation for $\hat{\alpha}_k$, and formalize this as Lemma 4.

Proof (of Lemma 4) First, assume that $\operatorname{sign}(d_0) = \operatorname{sign}(d_{r_k})$, so $\operatorname{sign}(\hat{\alpha}_k) = \operatorname{sign}(\alpha_k^+)$ by (34). Thus, by Lemma 7,

$$|P(\hat{\alpha}_k)-P(\alpha_k^+)| \le \frac{1}{2}(\alpha_k^+-\hat{\alpha}_k)^2\left(h_0+M|\hat{\alpha}_k|+\frac{M}{3}|\alpha_k^+-\hat{\alpha}_k|\right). \quad (38)$$

Since $h_0 = u_k^\top H_k u_k \le L$, it only remains to find appropriate bounds for $|\alpha_k^+ - \hat{\alpha}_k|$ and $\hat{\alpha}_k$. For notational convenience, define $S_r := \sqrt{h_r^2 + 2M|d_r|}$ for $r \ge 0$. As f is convex we know that $|d_0| \le \|g_k\| \le \sqrt{2L(f(x_k) - f_\star)}$, see [7, Prop. B.3] and so

$$M|\hat{\alpha}_k| \overset{(34)}{=} |S_0 - h_0| \le \sqrt{S_0^2 - h_0^2} = \sqrt{2M|d_0|} \le \sqrt{2M\|g_k\|}$$
$$\le \sqrt{2M\sqrt{2L(f(x_k) - f_\star)}} = \sqrt{\frac{2}{R^3}\left(MR^3\right)\sqrt{\frac{2}{R^2}(LR^2)(f(x_k) - f_\star)}} \le \frac{2^{3/4}B}{R^2},$$

using the definition of $B = \max(LR^2, MR^3, f(x_k) - f_\star)$. Defining the finite difference errors $e_k^d = d_{r_k} - d_0$ and $e_k^h = h_{r_k} - h_0$, Lemma 6 implies

$$|\hat{\alpha}_k - \alpha_k^+| \le \frac{|e_k^h|}{M} + \frac{2|e_k^d|}{S_0 + S_{r_k}}. \tag{39}$$

As $\|u_k\| = 1$ and H is assumed Lipschitz continuous (i.e. $a = 1$), from (30), we have $|e_k^h| \le \frac{Mr_k}{3}$ and the first term on the right-hand side of (39) is bounded by $\frac{r_k}{3}$. Appealing to (29) we obtain $|e_k^d| \le \frac{Mr_k^2}{6}$. We use this and the fact that $\operatorname{sign}(d_0) = \operatorname{sign}(d_k)$ to bound the second term on the right-hand side of (39):

$$\frac{2|e_k^d|}{S_0 + S_{r_k}} = \frac{2|d_k - d_0|}{S_0 + S_{r_k}} \le \frac{2\left(\sqrt{|d_0|} + \sqrt{|d_k|}\right)\left|\sqrt{|d_0|} - \sqrt{|d_k|}\right|}{\sqrt{2M|d_0|} + \sqrt{2M|d_k|}}$$
$$= \frac{2\left|\sqrt{|d_0|} - \sqrt{|d_k|}\right|}{\sqrt{2M}} \le \frac{2\sqrt{|d_0 - d_k|}}{\sqrt{2M}} \le 2\sqrt{\frac{1}{2M}\frac{Mr_k^2}{6}} = \frac{r_k}{\sqrt{3}}.$$

This provides a nice bound independent of L, M, and R; $|\hat{\alpha}_k - \hat{\alpha}_k| \le (1/3 + 1/\sqrt{3})r_k < r_k$. Combining everything with (38), we get

$$\left|P_0(\hat{\alpha}_k) - P_0(\alpha_k^+)\right| < \frac{1}{2}r_k^2\left(L + \frac{2^{3/4}B}{R^2} + \frac{M}{3}r_k\right) \le \frac{1}{2}r_k^2\left(\frac{B}{R^2} + \frac{2^{3/4}B}{R^2} + \frac{B}{3R^2}\right)$$
$$\le \frac{Br_k^2}{R^2}\left(\frac{2}{3} + \frac{1}{2^{1/4}}\right) \le \frac{2B}{R^2}r_k^2.$$

If $\operatorname{sign}(d_0) = -\operatorname{sign}(d_{r_k})$ then $\operatorname{sign}(\hat{\alpha}_k) = \operatorname{sign}(\alpha_k^-)$, again by (34). Lemmas 6 and 7 now yield

$$|\hat{\alpha}_k - \alpha_k^-| \le \frac{|e_k^h|}{M} + \frac{2|d_0 - (-d_{r_k})|}{S_0 + S_{r_k}}$$
$$|P(\hat{\alpha}_k) - P(\alpha_k^-)| \le \frac{1}{2}(\alpha_k^- - \hat{\alpha}_k)^2\left(h_0 + M|\hat{\alpha}_k| + \frac{M}{3}|\alpha_k^- - \hat{\alpha}_k|\right). \tag{40}$$

The first term in (40) can be bounded as before. Because $|d_0 + d_{r_k}| \le |d_0 - d_{r_k}| \le |e_k^d|$ as d_0 and d_{r_k} have opposite signs, the second term in (40) is bounded by $r_k/\sqrt{3}$ as before. Following the proof of the $\operatorname{sign}(d_0) = \operatorname{sign}(d_{r_k})$ case we conclude that,

$$\left|P_0(\hat{\alpha}_k) - P_0(\alpha_k^-)\right| \le \frac{2B}{R^2}r_k^2,$$

thus proving the theorem. □

The proofs of Theorems 2 and 3 are closely related to the proofs of [34, Lemma 5.7] and [34, Theorem 5.8]. However, unlike in [34], extra care must be taken in accounting for the error arising from using finite difference approximations to the (first and second) derivative. Moreover, the proof of Theorem 2 streamlines that of [34, Lemma 5.7]—as we are only focused on one-dimensional subspaces. The proof of Theorem 3 provides additional flexibility as compared to that of [34, Theorem 5.8] in the form of a tunable exponent s in the auxiliary sequence γ_k. In [34] this exponent is fixed to be 2. As we shall see, this flexibility proves vital in providing simple ways to account for the error induced by finite differencing.

Proof (of Theorem 2) First, fix $u_k \in \mathbb{R}^d$ drawn from $\mathcal{D}_k$. Then, for $\sigma = -\text{sign}(d_0(x_k; u_k))$ and any $z \in \mathbb{R}^d$,

$$\begin{aligned}
&f(x_{k+1}) - f(x_k) \\
&\quad \le f(x_k + \alpha_k^\sigma u) - f(x_k) \quad \le P(\alpha_k^\sigma; d_0(x_k; u_k), h_0(x_k; u_k)) && \text{(Eq. (16))} \\
&\quad \le P(\hat{\alpha}_k; d_0, h_0) + \frac{2B}{R^2} r_k^2 && \text{(Lemma 4)} \\
&\quad \le P(u_k^\top z; d_0, h_0) + \frac{2B}{R^2} r_k^2 && \text{(minimality of } \hat{\alpha}_k) \\
&\quad = (z^\top u_k)(u_k^\top g_k) + \frac{1}{2}(z^\top u_k)(u_k^\top H_k u_k)(u_k^\top z) + \frac{M}{6}|u_k^\top z|^3 + \frac{2B}{R^2} r_k^2
\end{aligned}$$

holds. Now taking the expectation and using the isotropy condition:

$$\mathbb{E}\left[f(x_{k+1}) \mid x_k\right] - f(x_k) \le \frac{1}{d} z^\top g_k + \frac{1}{2} z^\top \mathbb{E}\left[u_k u_k^\top H_k u_k u_k^\top\right] z + \frac{M}{6}\mathbb{E}\left[|u_k^\top z|^3\right] + \frac{2B}{R^2} r_k^2.$$

Note that the expectations above satisfy $\frac{1}{2} z^\top \mathbb{E}\left[u_k u_k^\top H_k u_k u_k^\top\right] z \le \frac{1}{2} z^\top \mathbb{E}\left[L u_k u_k^\top\right] z = \frac{L}{2d}\|z\|^2$ and $\mathbb{E}\left[|u_k^\top z|^3\right] \le \mathbb{E}\left[|u_k^\top z|^2\right]\|z\| = \frac{1}{d}\|z\|^3$, respectively. Therefore,

$$\mathbb{E}\left[f(x_{k+1}) \mid x_k\right] - f(x_k) \le \frac{1}{d} z^\top g_k + \frac{L}{2d}\|z\|^2 + \frac{M}{6d}\|z\|^3 + \frac{2B}{R^2} r_k^2. \tag{41}$$

Now, using convexity of f, namely $f(x_k + z) - f(x_k) \ge z^\top g_k$, we obtain (19). □

Proof (of Theorem 3) Let $\delta(x) := f(x) - f_\star$ denote the optimality gap while $\delta_k := \mathbb{E}[\delta(x_k)]$ denotes the expected optimality gap. Since Algorithm 2 has non-increasing δ_k, we may assume $\delta_0 > \varepsilon$. Note that δ is convex. Letting $x_\star$ be any fixed minimizer (i.e. $f(x_\star) = f_\star$), we note that $\delta(x_\star) = 0$. For any $t_k \in (0, 1)$, setting $z = t_k(x_\star - x_k)$ in Theorem 2 and defining $\Delta_k = \|x_\star - x_k\|$ yields

$$\begin{aligned}
&\mathbb{E}\left[f(x_{k+1}) \mid x_k\right] - f_\star \\
&\quad \le \left(1 - \frac{1}{d}\right) f(x_k) + \frac{1}{d} f((1 - t_k)x_k + t_k x_\star) - f_\star + \frac{L}{2d} t_k^2 \Delta_k^2 + \frac{M}{6d} t_k^3 \Delta_k^3 + \frac{2B}{R^2} r_k^2
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\left[\delta(x_{k+1}) \mid x_k\right] &\le \left(1 - \frac{1}{d}\right) f(x_k) + \frac{1 - t_k}{d} f(x_k) + \frac{t_k}{d} f_\star - f_\star \\
&\quad + \frac{L}{2d} t_k^2 \Delta_k^2 + \frac{M}{6d} t_k^3 \Delta_k^3 + \frac{2B}{R^2} r_k^2 \qquad (42) \\
\mathbb{E}\left[\delta(x_{k+1}) \mid x_k\right] &\le (1 - \frac{1}{d} + \frac{1}{d} - \frac{t_k}{d}) f(x_k) - \left(1 - \frac{t_k}{d}\right) f_\star
\end{aligned}$$

$$+\frac{L}{2d}t_k^2\Delta_k^2+\frac{M}{6d}t_k^3\Delta_k^3+\frac{2B}{R^2}r_k^2 \tag{43}$$

$$\delta_{k+1}\le\left(1-\frac{t_k}{d}\right)\delta_k+\frac{L}{2d}t_k^2\,\mathbb{E}\left[\Delta_k^2\right]+\frac{M}{6d}t_k^3\,\mathbb{E}\left[\Delta_k^3\right]+\frac{2B}{R^2}r_k^2, \tag{44}$$

where in (42) we use the convexity of f, in (43) we use $f(x_\star)=f_\star$, and in (44) we take the total expectation and use the definition of δ_k. We adopt an auxiliary sequence $\{\beta_k\}$ to make (44) telescoping. Let $s>1$, and define $\gamma_k=k^s$ and $\beta_k=\beta_0+\sum_{j=1}^k\gamma_j$ with $\beta_0=s^s d^{s+1}/(s+1)$, then $t_k=d\frac{\gamma_{k+1}}{\beta_{k+1}}\in(0,1)$, and $1-\frac{t_k}{d}=\frac{\beta_k}{\beta_{k+1}}$. We further note that:

$$\frac{k^{s+1}}{s+1}\le\beta_0+\int_1^k\frac{1}{x^s}dx\le\beta_k\le\beta_0+\int_2^{k+1}\frac{1}{x^s}dx=\beta_0+\frac{(k+1)^{s+1}}{s+1} \tag{45}$$

Then by multiplying β_{k+1} on both sides of (44), we get

$$\beta_{k+1}\delta_{k+1}\le\beta_k\delta_k+\frac{Ld}{2}\frac{\gamma_{k+1}^2}{\beta_{k+1}}\,\mathbb{E}\left[\Delta_k^2\right]+\frac{Md^2}{6}\frac{\gamma_{k+1}^3}{\beta_{k+1}^2}\,\mathbb{E}\left[\Delta_k^3\right]+\frac{2B}{R^2}\beta_{k+1}r_k^2,$$

and summing up from $k=0$ to $K-1$, we have

$$\delta_K\le\frac{\beta_0}{\beta_K}\delta_0+\frac{Ld}{2\beta_K}\sum_{k=1}^K\frac{\gamma_k^2}{\beta_k}\,\mathbb{E}\left[\Delta_k^2\right]+\frac{Md^2}{6\beta_K}\sum_{k=1}^K\frac{\gamma_k^3}{\beta_k^2}\,\mathbb{E}\left[\Delta_k^3\right]+\frac{2B}{R^2\beta_K}\sum_{k=1}^K\beta_k r_{k-1}^2. \tag{46}$$

First, $\frac{\beta_0}{\beta_K}\le\frac{\beta_0}{\beta_K-\beta_0}\le\frac{s^s}{(K/d)^{s+1}}$. Because the sequence $f(x_k)$ is non-increasing, $x_k\in\mathcal{Q}$ for all $k\ge0$ and so $\Delta_k\le\mathcal{R}$ (see Definition 3). Using $(1+\frac{1}{K})^s\le e^{s/K}$,

$$\begin{aligned}\frac{1}{\beta_K}\sum_{k=1}^K\frac{\gamma_k^2}{\beta_k}\Delta_{k-1}^2&\overset{(45)}{\le}\frac{\mathcal{R}^2(s+1)^2}{K^{s+1}}\sum_{k=1}^K\frac{k^{2s}}{k^{s+1}}=\frac{\mathcal{R}^2(s+1)^2}{K^{s+1}}\sum_{k=1}^K k^{s-1}\\&\le\frac{\mathcal{R}^2(s+1)^2}{K^{s+1}}\frac{(K+1)^s}{s}\le\frac{\mathcal{R}^2e^{s/K}(s+1)^2}{sK}\end{aligned}$$

and

$$\begin{aligned}\frac{1}{\beta_K}\sum_{k=1}^K\frac{\gamma_k^3}{\beta_k^2}\Delta_{k-1}^3&\overset{(45)}{\le}\frac{\mathcal{R}^3(s+1)^3}{K^{s+1}}\sum_{k=1}^K\frac{k^{3s}}{k^{2s+2}}=\frac{\mathcal{R}^3(s+1)^3}{K^{s+1}}\sum_{k=1}^K k^{s-2}\\&\le\frac{\mathcal{R}^3(s+1)^3}{K^{s+1}}\frac{(K+1)^{s-1}}{s-1}\le\frac{\mathcal{R}^3e^{(s-1)/K}(s+1)^3}{(s-1)K^2}.\end{aligned}$$

Lastly, the error due to the finite difference is controlled by the sampling radius:

$$\begin{aligned}\frac{2B}{R^2\beta_K}\sum_{k=1}^K\beta_k r_{k-1}^2&\overset{(45)}{\le}\frac{2(s+1)B\varepsilon\rho^2}{R^2K^{s+1}}\sum_{k=1}^K\frac{(k+1)^s}{s+1}+\frac{2B\varepsilon\rho^2\beta_0}{R^2\beta_K}\sum_{k=1}^K\frac{1}{(k+1)}\\&\le\frac{\varepsilon e^{2(s+1)/K}}{s+1}+\frac{\varepsilon\beta_0\log(K+2)}{\beta_K}.\end{aligned}$$

Combining the above with $\varepsilon<\delta_0$ we get

$$\delta_K\le\frac{s^s\delta_0(1+\log(K+2))}{(K/d)^{s+1}}+\frac{e^{s/K}(s+1)^2L\mathcal{R}^2}{2s(K/d)}$$

$$+\frac{e^{(s-1)/K}(s+1)^3M\mathcal{R}^3}{6(s-1)(K/d)^2}+\frac{e^{2(s+1)/K}}{s+1}\varepsilon. \tag{47}$$

For simplicity, we now choose $s=4$, whence

$$\delta_K \le \frac{256\delta_0(1+\log(K+2))}{(K/d)^5}+\frac{25e^{4/K}L\mathcal{R}^2}{8K/d}+\frac{125e^{3/K}M\mathcal{R}^3}{18(K/d)^2}+\frac{e^{10/K}}{5}\varepsilon. \tag{48}$$

We may choose K sufficiently large such that each term in (48) is less then $\varepsilon/4$. Indeed, if $K \ge 50$ then $\frac{e^{10/K}}{5}\varepsilon \le \frac{\varepsilon}{4}$. This also simplifies other exponential factors:

$$\frac{125e^{3/K}}{18} \le 8, \quad \frac{25e^{4/K}}{8} \le 4$$

Using these bounds,

1. If $\frac{K}{d} \ge \sqrt{\frac{32\,M\mathcal{R}^3}{\varepsilon}}$ then $\frac{125e^{3/K}M\mathcal{R}^3}{18(K/d)^2} \le \frac{\varepsilon}{4}$.
2. If $\frac{K}{d} \ge \frac{16L\mathcal{R}^2}{\varepsilon}$ then $\frac{25e^{4/K}L\mathcal{R}^2}{8K/d} \le \frac{\varepsilon}{4}$.
3. If $K \ge \left(\frac{1024\delta_0 d^5}{\varepsilon}\right)^{1/4}$ then $\frac{256\delta_0(1+\log(K+2))}{(K/d)^5} \le \frac{\varepsilon}{4}$ (using the simple upper bound $1+\log(K+2) \le K$, which is valid for $K \ge 50$).

Combining these terms, and rearranging where appropriate to emphasize the dependence on d and ε:

$$K \ge \max\left\{16L\mathcal{R}^2\frac{d}{\varepsilon}, \sqrt{32M\mathcal{R}^3}\frac{d}{\sqrt{\varepsilon}}, (1024\delta_0)^{1/4}\frac{d^{5/4}}{\varepsilon^{1/4}}, 50\right\}$$

Finally, we recall that $\delta_0 := f(x_0) - f_\star$ to obtain the stated bound. □

Remark 4 We have not attempted to optimize s. A better choice may yield improved constants.

5 Experimental Results

For a detailed description of all experimental settings and hyperparameters, see Appendix A. The code for all the experiments can be found online at https://github.com/bumsu-kim/CARS.

5.1 Convex Functions

As an illustrative example, we compared the performance of CARS and CARS-CR to STP [5], SMTP [28], Nesterov-Spokoiny [55], SPSA [59], 2SPSA [58], and AdaDGS [62] on the following convex quartic function:

$$f(x) = \alpha\sum_{i=1}^{d} x_i^4 + \frac{1}{2}x^\top Ax + \beta\|x\|^2,$$

where $\alpha, \beta > 0$ and $A = G^\top G$ with $G_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$ for $i, j = 1, 2, \ldots, d$. We show in Fig. 1 the objective function value versus the number of function queries. Note that CARS selected $x_{k+1} = x_{\mathrm{CARS},k}$ on over 95% of the iterations (see line 9 of Algorithm 1) while CARS-CR selected either $x_{k+1} = x_{\mathrm{CR}+,k}$ or $x_{k+1} = x_{\mathrm{CR}-,k}$ in 100% of iterations.

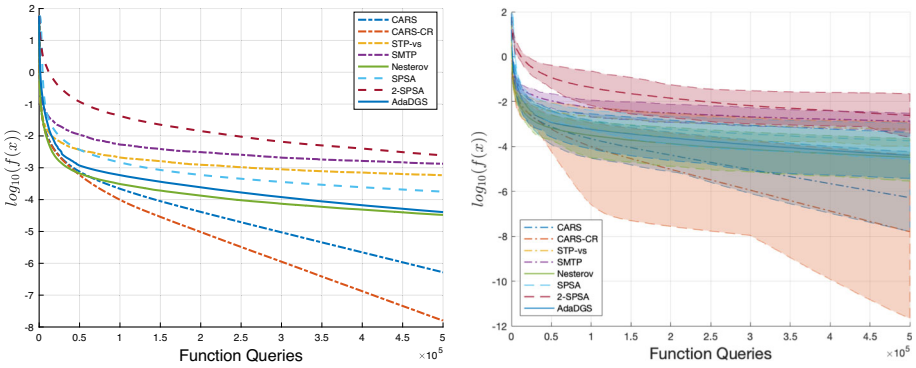

Fig. 1 Performance of each algorithm on a convex quartic function $f(x) = 0.1\sum_{i=1}^{d} x_i^4 + \frac{1}{2}x^\top Ax + 0.01\|x\|^2$, where $A = G^\top G$ with $G_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0,1)$ and $x \in \mathbb{R}^{30}$. **Left:** Mean over 20 trials, each with an independently sampled G. **Right:** Shading between the 10th and 90th percentiles, for the same 20 trials

5.2 Benchmark Problem Sets with Non-convex Functions

The test results in this section are presented in the form of performance profiles [21], which is a commonly used tool for comparing the performance of multiple algorithms over a suite of test problems. Performance profiles tend to be more informative than single-dimensional summaries (e.g. average number of iterations required to solve a problem). Formally, consider fixed sets of problems $\mathcal{P}$ and algorithms $\mathcal{S}$. For each $p \in \mathcal{P}$ and $s \in \mathcal{S}$ the *performance ratio* $r_{p,s}$ is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min_{s' \in \mathcal{S}} t_{p,s'}},$$

where $t_{p,s}$ is the number of function queries required for s to solve p. This is the relative performance of s on p compared to the best algorithm in $\mathcal{S}$ for p. The *performance profile* of s, $\rho_s : [1, \infty) \to [0, 1]$ is defined as

$$\rho_s(\tau) = \frac{|\{p \in \mathcal{P} : r_{p,s} \leq \tau\}|}{|\mathcal{P}|}.$$

Therefore, $\rho_s(1)$ is the fraction of problems for which s performs the best, while $\rho_s(\tau)$ for large τ measures the robustness of s. For all τ, a *higher value of* $\rho_s(\tau)$ *is better*. We use a log-scale on the horizontal access when plotting $\rho_s(\tau)$.

Moré–Garbow–Hillstrom Problems. We tested the same set of algorithms using the well-known non-convex Moré–Garbow–Hillstrom 34 test problems [50].

For each target accuracy ε, a problem is considered solved when we have $f(x_k) - f_\star \leq \varepsilon(f(x_0) - f_\star)$ within the budget of 20,000 queries. We used the recommended starting point x_0 as in [50] for all the tested algorithm, and repeated each test 10 times. The results are presented in Fig. 2.

CUTEst Problems. We further assessed the performance of CARS and CARS-CR to the same suite of algorithms on the CUTEst [29] problem set, which contains various convex and non-convex problems. As before, we compared the methods using performance profiles for the 146 problems with dimension less than or equal to 50. The query budget for each problem was set to be 20,000 times the problem dimension. The target accuracies were again set to $\varepsilon(f(x_0) - f_\star)$. The results are reported in Fig. 2.

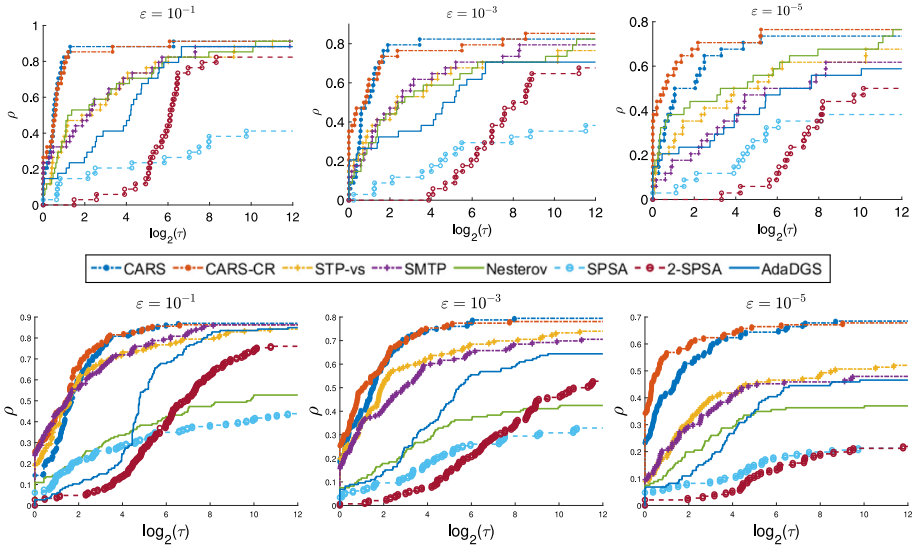

Fig. 2 Performance profiles on Moré–Garbow–Hillstrom problems (**upper**) and CUTEst problems (**lower**), for various target accuracies $\varepsilon = 10^{-1}$ (**left**), 10^{-3} (**middle**), and 10^{-5} (**right**). Our results demonstrate that CARS and CARS-CR consistently outperform other methods in terms of both efficiency (ρ at low τ values) and robustness (ρ at high τ values.) at all levels of accuracy

5.3 Black-Box Adversarial Attacks

Suppose $\mathcal{N}$ is an image classifier. The problem of generating small perturbations x that, when added to a natural image x_{nat}, fool the classifier (i.e. $\mathcal{N}(x_{\text{nat}} + x) \neq \mathcal{N}(x_{\text{nat}})$) is known as finding an *adversarial attack* [27]. As described in [16], when no access to the internal workings of the classifier is available, this problem becomes a black-box, or derivative-free, optimization problem. In order to ensure the attacked image $x_{\text{nat}} + x$ appears natural, a pixel-wise bound $\|x\|_\infty \leq \varepsilon_{\text{atk}}$ is usually enforced. CARS showed state-of-the-art performance in generating black-box adversarial attacks for $\mathcal{N}$ trained on the MNIST digit classification dataset [45]. We note that the dimension of the optimization problem here is $d = 784$.

In our experiments, $\mathcal{N}$ is a two-layer CNN achieving 99% test accuracy on unperturbed images. We use $\varepsilon_{\text{atk}} = 0.2$ and consider all 10,000 images from the test set of MNIST. We consider an attack a success if it fools $\mathcal{N}$ before a budget of 10,000 queries is met. The success rates, median and average queries for successful attacks are shown in Table 2. The results from ZOO [16], PGD-NES [36], and ZOHA-type algorithms [65] are cited from [65]. As pointed out in Sect. 2.1, the choice of sampling directions for CARS is not restrictive. Hence we used a similar initialization and distribution $\mathcal{D}$ as the Square Attack [1], which is known to be particularly well-suited for attacking CNN models. Visualization of attacked images is partly shown in Fig. 3. Detailed settings can be found in Appendix A.

Table 2 Comparison of success rates, and median and average function queries for the successful black-box adversarial attacks on MNIST with ℓ_∞-perturbation bound 0.2

Algorithm	Success rate (%)	Median queries	Average queries
ZOO*	93.95	11,700	11,804
PGD-NES*	88.39	2,450	4,584
ZOHA-Gauss*	91.69	1,400	2,586
ZOHA-Diag*	91.06	1,656	3,233
STP	53.64	2,193	3,141
SMTP	65.68	1,415	2,250
Nesterov	67.72	1,105	2,044
Square Attack	**98.21**	1,060	1,297
CARS (Square)	97.09	**717**	**1,169**

CARS, equipped with the Square Attack's distribution, shows the best performance in successful attacks, while reaching the second best success rate. The results marked with * are cited from [65]

Fig. 3 Adversarial examples with misclassified labels on MNIST generated with CARS

6 Concluding Remarks

We proposed two query-efficient and lightweight DFO algorithms: CARS and CARS-CR. Our analysis establishes their convergence on strongly convex functions and convex functions. Specifically, we develop a novel and rigorous analysis on the finite difference errors and the probability of significant descents of the objective function. CARS can incorporate various distributions, making it highly adaptable to a range of problem-specific distributions. We demonstrate the efficacy of CARS and CARS-CR through benchmark tests, where it outperforms existing methods in minimizing non-convex functions as well.

Appendix A: More on Numerical Experiments

In this section, we list the hyperparameters we used for each experiment. The code for all experiments can be found in https://github.com/bumsu-kim/CARS. We ran experiments on

two machines to distribute the load. A laptop equipped with Intel i5-9400F and Nvidia RTX 2060 and a workstation equipped with i9-9940X and two Nvidia RTX 2080 are used.

Moré–Garbow–Hillstrom and CUTEst Problems. The Moré–Garbow–Hillstrom Problem set consists of 34 non-convex smooth functions, where the problem dimension lies between 2 and 100. This experiment is conducted in Matlab. On the other hand, we used 146 unconstrained problems in the CUTEst Problem set, which have dimension not greater than 50. We used Julia for the CUTEst experiment.

We consider a problem solved when $f(x_k) - f_\star \leq \varepsilon(f(x_0) - f_\star)$. The target accuracies used here are $\varepsilon = 10^{-1}, 10^{-3}$ and 10^{-5}. For CARS, we used the sampling radius $r_k = 0.5/(k+2)$, $\hat{L} = 2$. For CARS-CR, we used the same sampling radius, and $M = 0.1$. For STP [5] and Nesterov-Spokoiny [55] we used the same hyperparameters as given in [5, Section 8.1]. We also used the same decreasing step-size for Stochastic Momentum Three Points method (SMTP) [28]. For the momentum parameter β for SMTP, we followed [28] and used $\beta = 0.5$. Namely, following the notations in [5] and [28], $\mathcal{D} = \text{Unif}(\mathbb{S}^{d-1})$ and $\alpha_k = \frac{1}{\sqrt{k+1}}$ (STP), $\alpha_k = \frac{1}{4(n+4)}$ and $\mu_k = 10^{-4}$, (Nesterov-Spokoiny), and $\gamma_k = \frac{1}{\sqrt{k+1}}$ and $\beta = 0.5$ (SMTP). For SPSA [59] and 2SPSA [58], we used the Rademacher distribution (i.e. $(u_k)_i = \pm 1$ with probability 0.5) for $\mathcal{D}$, $\alpha = 0.602$, $\gamma = 0.101$, $A = 100$, $a = 0.16$, and $c = 10^{-4}$. For AdaDGS [62], we used the code provided by the authors, by implementing the original Python code in Matlab. Some modifications on hyperparameters are made due to the difference in the scale of problem dimension, and the lack of domain width. First, the original AdaDGS code performs experiments on high dimensional problems (e.g. $d = 1000$), whereas $2 \leq d \leq 100$ in this experiment. Also, the problems are unconstrained, and $\|x_0 - x_\star\|$ varies from order of 10^0 to 10^6. Thus we used the following modified hyperparameters (following the notation of [62]):

1. The number of points used for line search $S = 100$, since the suggested value $0.05d(M-1)$ is too small for our experiments.
2. The initial smoothing(sampling) radius $\sigma_0 = 10^{-2}$. We tested $\sigma_0 = 5, 1, 10^{-1}, 10^{-2}$ and 10^{-3}, and chose the best value. When $\sigma_0 \leq 10^{-1}$ then the results were similar.

For plotting the performance profile, we set the performance ratio $r_{p,s} = r_M$ when p is not solved by s. Having $r_M = \infty$ is ideal, but setting it by a sufficiently large number does not make any difference. We used $r_M = 10^{20}$.

Black-box Adversarial Attacks. In this section, we explain the experiment setting for black-box adversarial attacks and also provide the hyperparameters that we used. The CNN model we attack has two 5×5 convolutional layers with 6 and 16 output channels, followed by a 4×4 convolutional layer with 120 output channels. Then two fully connected layers with 84 and 10 units follows. Between layers we use ReLU, and between convolutional layers we use 2×2 max-pooling as well. Finally we apply log softmax to the output layer. The test accuracy of the trained model is 98.99%.

For this particular experiment, we make three modifications to CARS. First, since the problem is highly non-convex ($h_r < 0$ at around 50% of the iteration), we do not compute x_{CARS} when $h_r < 0$ at k-th iteration. The second modification is due to the constraint of the problem. Let $\mathcal{F} = \{x \in [0,1]^d : \|x - x_0\| \leq \varepsilon_{\text{atk}}\}$ denote the feasible set. Inspired by [1], we also compute $x_{\text{bdry}} = x_k - t_{\max} d_r u_k$, where $t_{\max} = \max\{t > 0 : x_k - t d_r u_k \in \mathcal{F}\}$. To sum up,

$$x_{k+1} = \begin{cases} \arg\min\{f(x_k \pm r_k u_k),\ f(x_{\text{CARS}}),\ f(x_{\text{bdry}})\} & \text{if } h_r > 0; \\ \arg\min\{f(x_k \pm r_k u_k),\ f(x_{\text{bdry}})\} & \text{otherwise.} \end{cases}$$

We use the same sampling distribution as Square Attack [1], which is known to be particularly well-suited for attacking CNN models. Lastly, we perturbed x_0 by adding horizontal stripes. This choice of initialization is found to be very effective in [1].

Funding The work of HanQin Cai is partially supported by NSF DMS 2304489.

Code availability The software code of this paper can be accessed through https://github.com/bumsu-kim/CARS.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: European Conference on Computer Vision, pp. 484–501. Springer (2020)
2. Balasubramanian, K., Ghadimi, S.: Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. Foundations of Computational Mathematics pp. 1–42 (2021)
3. Berahas, A.S., Cao, L., Choromanski, K., Scheinberg, K.: A theoretical and empirical comparison of gradient approximations in derivative-free optimization. Found. Comput. Math. 1–54 (2021)
4. Berahas, A.S., Cao, L., Scheinberg, K.: Global convergence rate analysis of a generic line search algorithm with noise. SIAM J. Optim. **31**(2), 1489–1518 (2021)
5. Bergou, E.H., Gorbunov, E., Richtarik, P.: Stochastic three points method for unconstrained smooth minimization. SIAM J. Optim. **30**(4), 2726–2749 (2020)
6. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(1), 281–305 (2012)
7. Bertsekas, D.P.: Nonlinear programming. J. Oper. Res. Soc. **48**(3), 334–334 (1997)
8. Bibi, A., Bergou, E.H., Sener, O., Ghanem, B., Richtarik, P.: A stochastic derivative-free optimization method with importance sampling: Theory and learning to control. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3275–3282 (2020)
9. Cai, H., Lou, Y., McKenzie, D., Yin, W.: A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In: Proceedings of the 38th International Conference on Machine Learning, pp. 1193–1203. PMLR (2021)
10. Cai, H., Mckenzie, D., Yin, W., Zhang, Z.: A one-bit, comparison-based gradient estimator. Appl. Comput. Harmon. Anal. **60**, 242–266 (2022)
11. Cai, H., Mckenzie, D., Yin, W., Zhang, Z.: Zeroth-order regularized optimization (ZORO): approximately sparse gradients and adaptive sampling. SIAM J. Optim. **32**(2), 687–714 (2022)
12. Cartis, C., Massart, E., Otemissov, A.: Global optimization using random embeddings. Math. Program. 1–49 (2022)
13. Cartis, C., Otemissov, A.: A dimensionality reduction technique for unconstrained global optimization of functions with low effective dimensionality. Inf. Inference: A J. IMA **11**(1), 167–201 (2022)
14. Cartis, C., Roberts, L.: Scalable subspace methods for derivative-free nonlinear least-squares optimization. arXiv preprint arXiv:2102.12016 (2021)
15. Cartis, C., Roberts, L.: Scalable subspace methods for derivative-free nonlinear least-squares optimization. Math. Program. 1–64 (2022)
16. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 15–26 (2017)
17. Cheng, M., Singh, S., Chen, P., Chen, P.Y., Liu, S., Hsieh, C.J.: Sign-opt: A query-efficient hard-label adversarial attack. arXiv preprint arXiv:1909.10773 (2019)
18. Choromanski, K., Pacchiano, A., Parker-Holder, J., Tang, Y., Jain, D., Yang, Y., Iscen, A., Hsu, J., Sindhwani, V.: Provably robust blackbox optimization for reinforcement learning. In: Conference on Robot Learning, pp. 683–696 (2020)

19. Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., Weller, A.: Structured evolution with compact architectures for scalable policy optimization. In: International Conference on Machine Learning, pp. 970–978. PMLR (2018)
20. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to derivative-free optimization (2009)
21. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
22. Dong, Y., Cheng, S., Pang, T., Su, H., Zhu, J.: Query-efficient black-box adversarial attacks guided by a transfer-based prior. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9536–9548 (2021)
23. Fabian, V.: Stochastic approximation. In: Optimizing methods in statistics, pp. 439–470 (1971)
24. Fazel, M., Ge, R., Kakade, S., Mesbahi, M.: Global convergence of policy gradient methods for the linear quadratic regulator. In: International Conference on Machine Learning, pp. 1467–1476. PMLR (2018)
25. Ghadimi, S., Lan, G.: Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM J. Optim. **23**(4), 2341–2368 (2013)
26. Glasmachers, T., Krause, O.: The hessian estimation evolution strategy. In: International Conference on Parallel Problem Solving from Nature, pp. 597–609. Springer (2020)
27. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
28. Gorbunov, E., Bibi, A., Sener, O., Bergou, E.H., Richtárik, P.: A stochastic derivative free optimization method with momentum. arXiv preprint arXiv:1905.13278 (2019)
29. Gould, N.I., Orban, D., Toint, P.L.: Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput. Optim. Appl. **60**(3), 545–557 (2015)
30. Gower, R., Koralev, D., Lieder, F., Richtárik, P.: RSN: Randomized subspace newton. In: Advances in Neural Information Processing Systems, pp. 616–625 (2019)
31. Grippo, L., Lampariello, F., Lucidi, S.: Global convergence and stabilization of unconstrained minimization methods without derivatives. J. Optim. Theory Appl. **56**(3), 385–406 (1988)
32. Grippo, L., Rinaldi, F.: A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations. Comput. Optim. Appl. **60**(1), 1–33 (2015)
33. Grippo, L., Sciandrone, M.: Nonmonotone derivative-free methods for nonlinear equations. Comput. Optim. Appl. **37**(3), 297–328 (2007)
34. Hanzely, F., Doikov, N., Nesterov, Y., Richtarik, P.: Stochastic subspace cubic Newton method. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 4027–4038 (2020)
35. Heaton, H., Chen, X., Wang, Z., Yin, W.: Safeguarded learned convex optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 7848–7855 (2023)
36. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: International Conference on Machine Learning, pp. 2137–2146. PMLR (2018)
37. Jamieson, K.G., Nowak, R., Recht, B.: Query complexity of derivative-free optimization. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
38. Karimireddy, S.P., Stich, S.U., Jaggi, M.: Global linear convergence of Newton's method without strong-convexity or lipschitz gradients. arXiv preprint arXiv:1806.00413 (2018)
39. Karmanov, V.: Convergence estimates for iterative minimization methods. USSR Comput. Math. Math. Phys. **14**(1), 1–13 (1974)
40. Karmanov, V.: On convergence of a random search method in convex minimization problems. Theory Probab. Appl. **19**(4), 788–794 (1975)
41. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**(3), 385–482 (2003)
42. Kozak, D., Becker, S., Doostan, A., Tenorio, L.: A stochastic subspace approach to gradient-free optimization in high dimensions. Comput. Optim. Appl. **79**(2), 339–368 (2021)
43. Krutikov, V.: On the rate of convergence of the minimization method along vectors in a given directional system. USSR Comput. Math. Math. Phys. **23**(1), 154–155 (1983)
44. Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numer. **28**, 287–404 (2019)
45. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs. http://yann.lecun.com/exdb/mnist **2** (2010)
46. Liu, S., Chen, P.Y., Kailkhura, B., Zhang, G., Hero, A.O., III., Varshney, P.K.: A primer on zeroth-order optimization in signal processing and machine learning: principals, recent advances, and applications. IEEE Signal Process. Mag. **37**(5), 43–54 (2020)
47. Liu, S., Kailkhura, B., Chen, P.Y., Ting, P., Chang, S., Amini, L.: Zeroth-order stochastic variance reduction for nonconvex optimization. Advances in Neural Information Processing Systems **31** (2018)
48. Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J.D., Chen, D., Arora, S.: Fine-tuning language models with just forward passes. arXiv preprint arXiv:2305.17333 (2023)

49. Mania, H., Guy, A., Recht, B.: Simple random search of static linear policies is competitive for reinforcement learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 1805–1814 (2018)
50. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17–41 (1981)
51. Mutsenieks, V., Rastrigin, L.: Extremal control of continuous multi-parameter systems by the method of random search. Akademiia Nauk SSSR, Izvestiia1, Tekhnichekaia Kibernetika pp. 101–110 (1964)
52. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
53. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)
54. Nesterov, Y., Polyak, B.T.: Cubic regularization of newton method and its global performance. Math. Program. **108**(1), 177–205 (2006)
55. Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. Found. Comput. Math. **17**(2), 527–566 (2017)
56. Salimans, T., Ho, J., Chen, X., Sidor, S., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864 (2017)
57. Schrack, G., Choit, M.: Optimized relative step size random searches. Math. Program. **10**(1), 230–244 (1976)
58. Spall, J.C.: Adaptive stochastic approximation by the simultaneous perturbation method. IEEE Trans. Autom. Control **45**(10), 1839–1853 (2000)
59. Spall, J.C., et al.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control **37**(3), 332–341 (1992)
60. Stich, S.U., Muller, C.L., Gartner, B.: Optimization of convex functions with random pursuit. SIAM J. Optim. **23**(2), 1284–1309 (2013)
61. Sun, T., Shao, Y., Qian, H., Huang, X., Qiu, X.: Black-box tuning for language-model-as-a-service. In: International Conference on Machine Learning, pp. 20841–20855. PMLR (2022)
62. Tran, H., Zhang, G.: AdaDGS: An adaptive black-box optimization method with a nonlocal directional Gaussian smoothing gradient. arXiv preprint arXiv:2011.02009 (2020)
63. Wang, Y., Du, S., Balakrishnan, S., Singh, A.: Stochastic zeroth-order optimization in high dimensions. In: International Conference on Artificial Intelligence and Statistics, pp. 1356–1365. PMLR (2018)
64. Xiao, Q., Ling, Q., Chen, T.: Lazy queries can reduce variance in zeroth-order optimization. IEEE Trans. Signal Process. (2023)
65. Ye, H., Huang, Z., Fang, C., Li, C.J., Zhang, T.: Hessian-aware zeroth-order optimization for black-box adversarial attack. arXiv preprint arXiv:1812.11377 (2018)
66. Zhu, J.: Hessian inverse approximation as covariance for random perturbation in black-box problems. arXiv preprint arXiv:2011.13166 (2020)
67. Zhu, J., Wang, L., Spall, J.C.: Efficient implementation of second-order stochastic approximation algorithms in high-dimensional problems. IEEE Trans. Neural Netw. Learn. Syst. **31**(8), 3087–3099 (2019)