Instance-Dependent Bounds for Zeroth-order Lipschitz Optimization with Error Certificates

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of zeroth-order (black-box) optimization of a Lipschitz function f defined on a compact subset $\mathcal{X}$ of $\mathbb{R}^d$, with the additional constraint that algorithms must certify the accuracy of their recommendations. We characterize the optimal number of evaluations of any Lipschitz function f to find and certify an approximate maximizer of f at accuracy ε. Under a weak assumption on $\mathcal{X}$, this optimal sample complexity is shown to be nearly proportional to the integral $\int_{\mathcal{X}} \mathrm{d}\boldsymbol{x}/(\max(f) - f(\boldsymbol{x}) + \varepsilon)^d$. This result, which was only (and partially) known in dimension $d = 1$, solves an open problem dating back to 1991. In terms of techniques, our upper bound relies on a slightly improved analysis of the DOO algorithm that we adapt to the certified setting and then link to the above integral. Our instance-dependent lower bound differs from traditional worst-case lower bounds in the Lipschitz setting and relies on a local worst-case analysis that could likely prove useful for other learning tasks.

1 Introduction

The problem of optimizing a black-box function f with as few evaluations of f as possible arises in many scientific and industrial fields such as computer experiments (Jones et al., 1998; Richet et al., 2013) or automatic selection of hyperparameters in machine learning (Bergstra et al., 2011). For safety-critical applications, e.g., in aircraft or nuclear engineering, using sample-efficient methods is not enough. *Certifying* the accuracy of the output of the optimization method can be a crucial additional requirement (Vanaret et al., 2013). As a concrete example, Azzimonti et al. (2021) describe a black-box function in nuclear engineering whose output is a k-effective multiplication factor, for which a higher value corresponds to a higher nuclear hazard. Certifying the optimization error is a way to certify the worst-case k-effective factor, which may be required by safety authorities.

In this paper, we formally study the problem of finding and certifying an ε-approximate maximizer of a Lipschitz function f of d variables and characterize the optimal number of evaluations of any such function f to achieve this goal. We start by formally defining the setting.

1.1 Setting: Zeroth-order Lipschitz Optimization with Error Certificates

Let $f\colon \mathcal{X} \to \mathbb{R}$ be a function on a compact non-empty subset $\mathcal{X}$ of $\mathbb{R}^d$ and $\boldsymbol{x}^\star \in \mathcal{X}$ a maximizer.

Lipschitz assumption. We assume that f is Lipschitz with respect to a norm $\|\cdot\|$, that is, there exists $L \geq 0$ such that $\big|f(\boldsymbol{x}) - f(\boldsymbol{y})\big| \leq L\,\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. Furthermore, we assume such a Lipschitz bound L to be known. Even though the smallest Lipschitz constant $\mathrm{Lip}(f) := \min\{L' \geq 0 : f \text{ is } L'\text{-Lipschitz}\}$ is well defined mathematically, it is rarely known exactly in practical black-box problems. As a theoretical curiosity, we will briefly discuss the case $L = \mathrm{Lip}(f)$ (i.e., when the best

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Lipschitz constant of the unknown black-box function f is known exactly) in Section 4, but for most of our results, we will make the following more realistic assumption.

Assumption 1. *For some known Lipschitz constant L, the function $f\colon \mathcal{X} \to \mathbb{R}$ belongs to*

$$\mathcal{F}_L := \big\{ g\colon \mathcal{X} \to \mathbb{R} \mid g \text{ is Lipschitz and } \mathrm{Lip}(g) < L \big\} \ . \tag{1}$$

The Lipschitzness of f implies the weaker property that $f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) \le L \, \|\boldsymbol{x}^\star - \boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathcal{X}$, sometimes referred to as *Lipschitzness around a maximizer* $\boldsymbol{x}^\star \in \mathcal{X}$. Although this is not the focus of our work, we will mention when our results hold under this weaker assumption.

Online learning protocol. We study the case in which f is black-box, i.e., except for the *a priori* knowledge of L, we can only access f by sequentially querying its values at a sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \in \mathcal{X}$ of points of our choice. At every round $n \ge 1$, the query point $\boldsymbol{x}_n$ can be chosen as a deterministic function of the values $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_{n-1})$ observed so far. At the end of round n, using all the values $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$, the learner outputs two quantities:

- a recommendation $\boldsymbol{x}_n^\star \in \mathcal{X}$, with the goal of minimizing the *optimization error* (a.k.a. *simple regret*): $\max(f) - f(\boldsymbol{x}_n^\star)$;
- an *error certificate* $\xi_n \ge 0$, with the constraint to correctly upper bound the optimization error for any L-Lipschitz function $f\colon \mathcal{X} \to \mathbb{R}$, i.e., so that $\max(f) - f(\boldsymbol{x}_n^\star) \le \xi_n$.

We call *certified algorithm* any algorithm for choosing such a sequence $(\boldsymbol{x}_n, \boldsymbol{x}_n^\star, \xi_n)_{n \ge 1}$.

Our goal is to quantify the smallest number of evaluations of f that certified algorithms need in order to find and certify an approximate maximizer of f at accuracy ε. This objective motivates the following definition. For any accuracy $\varepsilon > 0$, we define the *sample complexity* of a certified algorithm A for an L-Lipschitz function f as

$$\sigma(A, f, \varepsilon) := \inf\big\{ n \ge 1 : \xi_n \le \varepsilon \big\} \in \{1, 2, \ldots\} \cup \{+\infty\} \, . \tag{2}$$

This corresponds to the first time when we can stop the algorithm while being sure to have an ε-optimal recommendation $\boldsymbol{x}_n^\star$.

1.2 Main Contributions and Outline of the Paper

The main result of this paper is a tight characterization (up to a $\log$ factor) of the optimal sample complexity of certified algorithms in any dimension $d \ge 1$, solving a three-decade old open problem raised by Hansen et al. (1991). More precisely, we prove the following instance-dependent upper and lower bounds, which we later state formally in Theorem 3 of Section 4 (see also discussions therein, as well as Propositions 2 and 3 for the limit case $L = \mathrm{Lip}(f)$).

Theorem (Informal statement)**.** *Under a mild geometric assumption on $\mathcal{X}$, there exists a computationally tractable algorithm A (e.g.,* c.DOO*, Algorithm 1) such that, for some constant $C > 0$, any Lipschitz function $f \in \mathcal{F}_L$ (see* (1)*) and any accuracy ε,*

$$\sigma(A, f, \varepsilon) \le C \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} \ , \tag{3}$$

while any certified algorithm A' must satisfy, for all $f \in \mathcal{F}_L$ and $c \approx (1 - \mathrm{Lip}(f)/L)^d / \log(1/\varepsilon)$,

$$c \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} \le \sigma(A', f, \varepsilon) \ . \tag{4}$$

In particular, this result extends to any dimension $d \ge 1$ the upper bound proportional to $\int_0^1 \mathrm{d}x / (f(x^\star) - f(x) + \varepsilon)$ that Hansen et al. (1991) derived in dimension $d = 1$ using arguments specific to the geometry of the real line.

Detailed contributions and outline of the paper. We make the following contributions.

- As a warmup, we show in Section 2 how to add error certificates to the DOO algorithm (well-known in the more classical zeroth-order Lipschitz optimization setting *without* error certificates, see Perevozchikov 1990; Munos 2011). We then upper bound its sample complexity by the quantity $S_{\mathrm{C}}(f, \varepsilon)$ defined in (5) below. This bound matches a recent bound derived by Bouttier et al. (2020) for a computationally much more expensive algorithm. In passing, we also slightly improve the packing arguments that Munos (2011) used in the non-certified setting.

- In Section 3 we show that, under a mild geometric assumption on $\mathcal{X}$, the complexity measure $S_{\mathrm{C}}(f,\varepsilon)$ is actually proportional to the compact integral $\int_{\mathcal{X}} \mathrm{d}\boldsymbol{x}/\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d$, which implies (3) above. This extends the bound of Hansen et al. (1991) ($d = 1$) to any dimension d.
- Finally, in Section 4, we prove the instance-dependent lower bound (4), which differs from traditional worst-case lower bounds in the Lipschitz setting. Our proof relies on a *local* worst-case analysis that could likely prove useful for other learning tasks.

Some of the proofs are deferred to the Supplementary Material, where we also recall useful results on packing and covering numbers (Section A), as well as provide a slightly improved sample complexity bound on the DOO algorithm in the more classical non-certified setting (Section E).

Limitations and open problems. There are some interesting directions that would be worth investigating in the future but we did not cover in this paper, such as noisy observations (see, e.g, Bubeck et al. 2011; Kleinberg et al. 2019) or adaptivity to smoothness (e.g., Munos 2011; Bartlett et al. 2019; we consider L-Lipschitz functions f with L known, although our lower bound suggests that no adaptivity could be possible for certified algorithms). Finally, even if the results of Section 2 could be easily extended to pseudo-metric spaces as in Munos (2014) and related works, our other results are finite-dimensional and exploit the normed space structure.

1.3 Related Works

We detail below some connections with the global optimization and the bandit optimization literatures.

Zeroth-order Lipschitz optimization with error certificates. The problem of optimizing a function with error certificates has been studied in different settings over the past decades. For instance, in convex optimization, an example of error certificate is given by the *duality gap* between primal and dual feasible points (see, e.g., Boyd and Vandenberghe 2004).

In our setting, namely, global zeroth-order Lipschitz optimization with error certificates, most of the attention seems to have been on the very natural (yet computationally expensive) algorithm introduced by Piyavskii (1972) and Shubert (1972).[1] In dimension $d = 1$, Hansen et al. (1991) proved that its sample complexity $\sigma(\mathrm{PS}, f, \varepsilon)$ for L-Lipschitz functions $f\colon [0,1] \to \mathbb{R}$ is at most proportional to the integral $\int_0^1 \big(f(x^\star) - f(x) + \varepsilon\big)^{-1} \mathrm{d}x$, and left the question of extending the results to arbitrary dimensions open, stating that the task of "*Extending the results of this paper to the multivariate case appears to be difficult*". Recently, writing $\mathcal{X}_\varepsilon := \{\boldsymbol{x} \in \mathcal{X} : \max(f) - f(\boldsymbol{x}) \le \varepsilon\}$ for the set of ε-optimal points, $\mathcal{X}_{(a,b]} := \{\boldsymbol{x} \in \mathcal{X} : a < \max(f) - f(\boldsymbol{x}) \le b\}$ for the set of points in between a and b optimal, and $\mathcal{N}(E, r)$ for the packing number of a set E at scale r (see Section 1.4), Bouttier et al. (2020, Theorem 2) proved a bound valid in any dimension $d \ge 1$ roughly of this form:

$$S_{\mathrm{C}}(f,\varepsilon) := \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) , \tag{5}$$

where the number of terms in the sum is $m_\varepsilon := \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$ (with $\varepsilon_0 := L \sup_{\boldsymbol{x},\boldsymbol{y} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|$) and the associated scales are given by $\varepsilon_{m_\varepsilon} := \varepsilon$ and $\varepsilon_k := \varepsilon_0 2^{-k}$ for all $k \in \{0, 1, \dots, m_\varepsilon - 1\}$.

The equivalence we prove in Section 3 between $S_{\mathrm{C}}(f,\varepsilon)$ and $\int_{\mathcal{X}} \mathrm{d}\boldsymbol{x}/(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon)^d$ solves in particular the question left open by Hansen et al. (1991). The upper bound we prove for the certified DOO algorithm in Section 2 also indicates that the bound $S_{\mathrm{C}}(f,\varepsilon)$ and the equivalent integral bound can be achieved with a computationally much more tractable algorithm. Indeed, the Piyavskii-Shubert algorithm requires at every step n to solve an inner global Lipschitz optimization problem close to the computation of a Voronoi diagram (see discussion in Bouttier et al. 2020, Section 1.1), hence a running time believed to grow as $n^{\Omega(d)}$ after n function evaluations. On the contrary, as detailed in Remark 1 (Section 2), the running time of our certified version of the DOO algorithm is only of the order of $n \log(n)$ after n of evaluations of f.

[1] For the interested reader who is unfamiliar with this classic algorithm, we added some details in Section D.2 of the Supplementary Material.

Connections with the bandit optimization literature: upper bounds. Our work is also strongly connected to the bandit optimization literature, in which multiple authors studied the global Lipschitz optimization problem with zeroth-order (or *bandit*) feedback, either with perfect (deterministic) or noisy (stochastic) observations. In the deterministic setting considered here, these papers show that though the number $(L/\varepsilon)^d$ of evaluations associated to a naive grid search is optimal for worst-case Lipschitz functions (e.g., Thm 1.1.2 by Nesterov 2003), sequential algorithms can approximately optimize more benign functions with a much smaller number of evaluations. Examples of algorithms with such guarantees in the deterministic setting are the branch-and-bound algorithm by Perevozchikov (1990), the DOO algorithm by Munos (2011) or the LIPO algorithm by Malherbe and Vayatis (2017). Examples of algorithms in the stochastic setting are the HOO algorithm by Bubeck et al. (2011) or the (generic yet computationally challenging) Zooming algorithm by Kleinberg et al. (2008, 2019). More examples and references can be found in the textbooks by Munos (2014) and Slivkins (2019).

Note however that, except for the work of Bouttier et al. (2020) mentioned earlier, these bandit optimization papers did not address the problem of *certifying* the accuracy of the recommendations $\boldsymbol{x}_n^\star$. Indeed, all bounds are related to a more classical notion of sample complexity, namely, the minimum number of queries made by an algorithm A before outputting ε-optimal recommendations only:

$$\zeta(A, f, \varepsilon) := \inf\{n \geq 1 : \boldsymbol{x}_n^\star \in \mathcal{X}_\varepsilon\} \in \{1, 2, \ldots\} \cup \{+\infty\} \,. \tag{6}$$

Though $\zeta(A, f, \varepsilon)$ is always upper bounded by $\sigma(A, f, \varepsilon)$ defined in (2), these two quantities can differ significantly, as shown by the simple example of constant functions f, for which $\zeta(A, f, \varepsilon) = 1$ but $\sigma(A, f, \varepsilon) \approx (L/\varepsilon)^d$ since the only way to *certify* that the output is ε-optimal is essentially to perform a grid-search with step-size roughly ε/L, so as to be sure there is no hidden bump of height more than ε. At a high level, the more "constant" a function is, the easier it is to recommend an ε-optimal point, but the harder it is to certify that such recommendation is actually a good recommendation. See the Supplementary Material (Section E) for a comparison of bounds.

Despite this important difference, the bandit viewpoint (using packing numbers instead of more specific one-dimensional arguments) is key to obtain our multi-dimensional integral characterization.

Comparison with existing lower bounds. Several worst-case (minimax) lower bounds were derived in the bandit Lipschitz optimization setting. When rewritten in terms of the accuracy ε and translated into our deterministic setting, the lower bounds of Horn (2006) (when $d^\star = d/2$) and of Bubeck et al. (2011) (for any $d^\star$) are of the form $\inf_A \sup_{f \in \mathcal{G}_{d^\star}} \zeta(A, f, \varepsilon) \geq \psi_{d^\star}(\varepsilon)$, where $\mathcal{G}_{d^\star}$ is the subset of L-Lipschitz functions with near-optimality dimension at most $d^\star$.

On the contrary, our instance-dependent lower bound (4) quantifies the minimum number of evaluations to certify an ε-optimal point *for each function* $f \in \mathcal{F}_L$. Our proof relies on a *local* worst-case analysis in the same spirit as for distribution-dependent lower bounds in stochastic multi-armed bandits (see, e.g., Theorem 16.2 in Lattimore and Szepesvári 2020), yet for continuous instead of finite action sets. We believe this lower bound technique should prove useful for other learning tasks.

1.4 Recurring Notation

This short section contains a summary of all the notation that we use in the paper and can be used by the reader for easy referencing. We denote the set of positive integers $\{1, 2, \ldots\}$ by $\mathbb{N}^*$ and let $\mathbb{N} := \mathbb{N}^* \cup \{0\}$. For all $n \in \mathbb{N}^*$, we denote by $[n]$ the set of the first n integers $\{1, \ldots, n\}$. We denote the Lebesgue measure of a (Lebesgue-measurable) set E by $\mathrm{vol}(E)$ and refer to it simply as its *volume*. For all $\rho > 0$ and $\boldsymbol{x} \in \mathbb{R}^d$, we denote by $B_\rho(\boldsymbol{x})$ the closed ball with radius ρ centered at $\boldsymbol{x}$, with respect to the arbitrary norm $\|\cdot\|$ that is fixed throughout the paper. We also write B_ρ for the ball with radius ρ centered at the origin and denote by v_ρ its volume.

$\mathrm{Lip}(f)$ denotes the smallest Lipschitz constant of our target L-Lipschitz function $f \colon \mathcal{X} \to \mathbb{R}$. The set of its ε-optimal points is denoted by $\mathcal{X}_\varepsilon := \{\boldsymbol{x} \in \mathcal{X} : \max(f) - f(\boldsymbol{x}) \leq \varepsilon\}$, its complement (i.e., the set of ε-*suboptimal points*) by $\mathcal{X}_\varepsilon^c$, and for all $0 \leq a < b$, the $(a, b]$-*layer* (i.e., the set of points that are b-optimal but a-suboptimal) by $\mathcal{X}_{(a,b]} := \mathcal{X}_a^c \cap \mathcal{X}_b = \big\{\boldsymbol{x} \in \mathcal{X} : a < f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) \leq b\big\}$. Since f is L-Lipschitz, every point in $\mathcal{X}$ is ε_0-optimal, with ε_0 defined by $\varepsilon_0 := L \max_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|$. In other words, $\mathcal{X}_{\varepsilon_0} = \mathcal{X}$. For this reason, without loss of generality, we will only consider values of accuracy ε smaller than or equal to ε_0.

For any bounded set $E \subset \mathbb{R}^d$ and all $r > 0$, the r-*packing number* of E is the largest cardinality of an r-packing of E, that is, $\mathcal{N}(E, r) := \sup\big\{k \in \mathbb{N}^* : \exists \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in E, \min_{i \neq j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| > r\big\}$ if E

is nonempty, zero otherwise; the *r-covering number* of E is the smallest cardinality of an r-covering of E, i.e., $\mathcal{M}(E, r) := \min\big\{k \in \mathbb{N}^* : \exists \boldsymbol{x}_1, \dots, \boldsymbol{x}_k \in \mathbb{R}^d, \forall \boldsymbol{x} \in E, \exists i \in [k], \|\boldsymbol{x} - \boldsymbol{x}_i\| \le r\big\}$ if E is nonempty, zero otherwise. Well-known and useful properties of packing (and covering) numbers are recalled in Section A of the Supplementary Material.

2 Warmup: Certified DOO Has Sample Complexity $S_{\mathrm{C}}(f, \varepsilon)$

In this section, we start by adapting the well-known DOO algorithm (Perevozchikov, 1990; Munos, 2011) to the certified setting. We then bound its sample complexity by the quantity $S_{\mathrm{C}}(f, \varepsilon)$ defined in Eq. (5). In passing, we slightly improve the packing arguments used by Munos (2011) in the non-certified setting (Supplementary Material, Section E). In Section 4, we will prove that this rate is optimal (up to logarithmic factors) for certified algorithms.

The certified DOO algorithm (c.DOO, Algorithm 1) is defined for a fixed $K \in \mathbb{N}^\star$, by an infinite sequence of subsets of $\mathcal{X}$ of the form $(X_{h,i})_{h \in \mathbb{N}, i=0,\dots,K^h-1}$, called *cells*. For each $h \in \mathbb{N}$, the cells $X_{h,0}, \dots, X_{h,K^h-1}$ are non-empty, pairwise disjoint, and their union contains $\mathcal{X}$. The sequence $(X_{h,i})_{h \in \mathbb{N}, i=0,\dots,K^h-1}$ is associated with a K-ary tree in the following way. For any $h \in \mathbb{N}$ and $j \in \{0, \dots, K^h - 1\}$, there exist K distinct $i_1, \dots, i_K \in \{0, \dots, K^{h+1} - 1\}$ such that $X_{h+1,i_1}, \dots, X_{h+1,i_K}$ form a partition of $X_{h,j}$. We call $(h+1, i_1), \dots, (h+1, i_K)$ the *children* of (h, j). To each cell $X_{h,i}$ ($h \in \mathbb{N}$, $i \in \{0, \dots, K^h - 1\}$) is associated a *representative* $\boldsymbol{x}_{h,i} \in X_{h,i}$, which can be thought of as the "center" of the cell. We assume that feasible cells have feasible representatives, i.e., that $X_{h,i} \cap \mathcal{X} \neq \varnothing$ implies $\boldsymbol{x}_{h,i} \in \mathcal{X}$. The two following assumptions prescribe a sufficiently good behavior of the sequences of cells and representatives.

Assumption 2. *There exist two positive constants $\delta \in (0, 1)$ and $R > 0$ such that, for any cell $X_{h,i}$ ($h \in \mathbb{N}$, $i = 0, \dots, K^h - 1$) and all $\boldsymbol{u}, \boldsymbol{v} \in X_{h,i}$, it holds that $\|\boldsymbol{u} - \boldsymbol{v}\| \le R\delta^h$.*

Assumption 3. *There exists $\nu > 0$ such that, with δ as in Assumption 2, for any $h \in \mathbb{N}$, $i = 0, \dots, K^h - 1$, $h' \in \mathbb{N}$, $i' = 0, \dots, K^{h'} - 1$, with $(h, i) \neq (h', i')$, $\|\boldsymbol{x}_{h,i} - \boldsymbol{x}_{h',i'}\| \ge \nu\delta^{\max(h,h')}$.*

The classic Assumption 2 is simply stating that diameters of cells decrease geometrically with the depth of the tree. Assumption 3, which is key for our improved analysis, is slightly stronger than the corresponding one in Munos (2011), yet very easy to satisfy. Indeed, one can prove that for any compact $\mathcal{X}$, it is always possible to find a sequence of cells and representatives satisfying Assumptions 2 and 3. For instance, if $\mathcal{X}$ is the unit hypercube $[0, 1]^d$ and $\|\cdot\|$ is the supremum norm $\|\cdot\|_\infty$, we can define cells by bisection, letting $K = 2^d$, $X_{h,i}$ be a hypercube of edge-length 2^{-h}, and $\boldsymbol{x}_{h,i}$ be its center (for $h \in \mathbb{N}$ and $i = 0, \dots, 2^{dh} - 1$). In this case, Assumptions 2 and 3 are satisfied with $R = 1$ and $\delta = \nu = 1/2$.

Algorithm 1: Certified DOO (c.DOO)

input: $\mathcal{X}$, L, K, δ, R, cells $(X_{h,i})_{h \in \mathbb{N}, i=0,\dots,K^h-1}$, representatives $(\boldsymbol{x}_{h,i})_{h \in \mathbb{N}, i=0,\dots,K^h-1}$
initialization: let $n \leftarrow 1$ and $\mathcal{L}_1 \leftarrow \{(0, 0)\}$
pick the first query point $\boldsymbol{x}_1 \leftarrow \boldsymbol{x}_{0,0}$
observe the value $f(\boldsymbol{x}_1)$
output recommendation $\boldsymbol{x}_1^\star \leftarrow \boldsymbol{x}_1$ and error certificate $\xi_1 \leftarrow LR$
for *iteration* $k = 1, 2, \dots$ **do**
  let $(h^\star, i^\star) \in \operatorname{argmax}_{(h,i) \in \mathcal{L}_n} \big\{f(\boldsymbol{x}_{h,i}) + LR\delta^h\big\}$ `// ties broken arbitrarily`
  let $\mathcal{L}_+$ be the set of the K children of $(h^\star, i^\star)$
  for each *child* $(h^\star + 1, j) \in \mathcal{L}_+$ *of* $(h^\star, i^\star)$ **do**
    if $X_{h^\star+1,j} \cap \mathcal{X} \neq \varnothing$ **then**
      let $n \leftarrow n + 1$ and $\mathcal{L}_n \leftarrow \mathcal{L}_{n-1} \cup \{(h^\star + 1, j)\}$
      pick the next query point $\boldsymbol{x}_n \leftarrow \boldsymbol{x}_{h^\star+1,j}$
      observe the value $f(\boldsymbol{x}_n)$
      output a recommendation $\boldsymbol{x}_n^\star \in \operatorname{argmax}_{\boldsymbol{x} \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}} f(\boldsymbol{x})$
      output the error certificate $\xi_n \leftarrow f(\boldsymbol{x}_{h^\star,i^\star}) + LR\delta^{h^\star} - f(\boldsymbol{x}_n^\star)$
  remove $(h^\star, i^\star)$ from $\mathcal{L}_n$

Our certified version of the DOO algorithm (c.DOO, Algorithm 1) maintains a set of indices of *active* cells $\mathcal{L}_n$ throughout rounds n. During each iteration k, it selects the index of the most promising active cell $(h^\star, i^\star)$ (Line 5) and splits it into its K children $\mathcal{L}_+$ (Line 6). Then, it sequentially picks the representatives of the cells corresponding to each of these children (Line 10), observes the value of the target function f at these points (Line 11), recommends the point with the highest observed value of f (Line 12), and outputs a certificate $\xi_n = \big(f(\boldsymbol{x}_{h^\star,i^\star}) + LR\delta^{h^\star}\big) - f(\boldsymbol{x}_n^\star)$ (Line 13) that is the difference between an upper bound on f and the currently recommended value $f(\boldsymbol{x}_n^\star)$. In the meantime, all children in $\mathcal{L}_+$ are added to the set of active indices $\mathcal{L}_n$ (Line 9), and the current iteration is concluded by removing $(h^\star, i^\star)$ from $\mathcal{L}_n$ (Line 14), now that it has been replaced by its refinement $\mathcal{L}_+$.

Remark 1. *The running-time of* c.DOO *(ignoring the cost of calling the function f) is driven by the computation of the recommendation $\boldsymbol{x}_n^\star \in \operatorname{argmax}_{\boldsymbol{x}\in\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_n\}} f(\boldsymbol{x})$ (Line 12) and the search of the index of the most promising active cell $(h^\star, i^\star) \in \operatorname{argmax}_{(h,i)\in\mathcal{L}_n}\{f(\boldsymbol{x}_{h,i}) + LR\delta^h\}$ (Line 5). The recommendation $\boldsymbol{x}_n^\star$ can be computed sequentially in constant time (by comparing the new value $f(\boldsymbol{x}_n)$ with the current maximum). In Line 5, the leaf $(h^\star, i^\star)$ to be split at iteration k can be computed sequentially in logarithmic time (using a max-heap structure). Therefore, the running time of* c.DOO *is of order $n\log(n)$ in the number n of evaluations of the function f.*

The next proposition shows that the sample complexity of the certified DOO algorithm is upper bounded (up to constants) by the instance-dependent quantity $S_{\mathrm{C}}(f,\varepsilon)$ introduced in Eq. (5).

Proposition 1. *If Assumptions 2 and 3 hold, then Algorithm 1 is a certified algorithm and letting $a := 1 + K\big(\mathbf{1}_{\nu/R\geq 1} + \mathbf{1}_{\nu/R<1}(4R/\nu)^d\big)$, its sample complexity satisfies, for all Lipschitz functions $f \in \mathcal{F}_L$ and any accuracy $\varepsilon \in (0, \varepsilon_0]$,*

$$\sigma(\mathrm{c.DOO}, f, \varepsilon) \leq a S_{\mathrm{C}}(f,\varepsilon)\ .$$

The proof is postponed to Section B of the Supplementary Material and shares some arguments with those of Perevozchikov (1990) and Munos (2011), originally written for the non-certified setting. The key change is to partition the values of f, instead of its domain $\mathcal{X}$ at any depth h of the tree (see Munos 2011), when counting the representatives selected at all levels. The idea of using layers $\mathcal{X}_{(\varepsilon_i,\ \varepsilon_{i-1}]}$ was already present in Kleinberg et al. (2008, 2019) and Bouttier et al. (2020) for more computationally challenging algorithms (see discussion in Section 1.3).

3 Characterization of $S_{\mathrm{C}}(f,\varepsilon)$

Earlier, we mentioned that the quantity $S_{\mathrm{C}}(f,\varepsilon)$ introduced in Eq. (5) upper bounds the sample complexity of several certified algorithms, such as c.DOO or Piyavskii-Shubert. In this section, we provide a characterization of this quantity in terms of a much cleaner and compact integral expression.

This result is inspired by Hansen et al. (1991), that in dimension $d = 1$, derive an elegant bound on the sample complexity $\sigma(\mathrm{PS}, f, \varepsilon)$ of the certified Piyavskii-Shubert algorithm for any Lipschitz function f and accuracy ε. They proved that $\sigma(\mathrm{PS}, f, \varepsilon)$ is upper bounded by $\int_0^1 \mathrm{d}x/(f(x^\star) - f(x) + \varepsilon)$ up to constants. However, the authors rely heavily on the one-dimensional assumption and the specific form of the Piyavskii-Shubert algorithm in this setting, stating that the task of "*Extending the results of this paper to the multivariate case appears to be difficult*". In this section, we show an equivalence between $S_{\mathrm{C}}(f,\varepsilon)$ and this type of integral bound in any dimension d. Putting this together with a recent result of Bouttier et al. (2020) (which proves that up to constants, $\sigma(\mathrm{PS}, f, \varepsilon) \leq S_{\mathrm{C}}(f,\varepsilon)$) solves the long-standing problem raised by Hansen et al. (1991) three decades ago.

To tame the wild spectrum of shapes that compact subsets may have, we will assume that $\mathcal{X}$ satisfies the following mild geometric assumption. At a high level, it says that a constant fraction of each (sufficiently small) ball centered at a point in $\mathcal{X}$ is included in $\mathcal{X}$. This removes sets containing isolated points or cuspidal corners and includes most domains that are typically used, such as non-degenerate polytopes, ellipsoids, finite unions of them, etc. This natural assumption is weaker than the classic rolling ball assumption from the statistics literature (Cuevas et al., 2012; Walther, 1997) and has already been proved useful in the past (Hu et al., 2020).

Assumption 4. *There exist two constants $r_0 > 0, \gamma \in (0,1]$ such that, for any $\boldsymbol{x} \in \mathcal{X}$ and all $r \in (0, r_0)$, $\mathrm{vol}\big(B_r(\boldsymbol{x}) \cap \mathcal{X}\big) \geq \gamma v_r$.*

We can now state the main result of this section. Its proof relies on some additional technical results that are deferred to the Supplementary Material.

Theorem 1. *For any Lipschitz function $f \in \mathcal{F}_L$, if Assumption 4 holds with $r_0 > \varepsilon_0/2L, \gamma \in (0,1]$,*[2] *then there exist $c, C > 0$ (e.g., $c := 1/v_{1/L}$ and $C := 1/(\gamma v_{1/128L})$) such that, for all $\varepsilon \in (0, \varepsilon_0]$,*

$$c \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} \leq S_{\mathrm{C}}(f, \varepsilon) \leq C \int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} \,.$$

Proof. Fix any $\varepsilon \in (0, \varepsilon_0]$ and recall that $m_\varepsilon := \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$, $\varepsilon_{m_\varepsilon} := \varepsilon$, and for all $k \leq m_\varepsilon - 1$, $\varepsilon_k := \varepsilon_0 2^{-k}$. Partition the domain of integration $\mathcal{X}$ into the following $m_\varepsilon + 1$ sets: the set of ε-optimal points $\mathcal{X}_\varepsilon$ and the m_ε layers $\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]}$, for $k \in [m_\varepsilon]$. We begin by proving the first inequality:

$$\begin{aligned}
\int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} &\leq \frac{\mathrm{vol}(\mathcal{X}_\varepsilon)}{\varepsilon^d} + \sum_{k=1}^{m_\varepsilon} \frac{\mathrm{vol}\big(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]}\big)}{(\varepsilon_k + \varepsilon)^d} \\
&\leq \frac{\mathcal{M}\big(\mathcal{X}_\varepsilon,\ \frac{\varepsilon}{L}\big) \cdot v_1\big(\frac{\varepsilon}{L}\big)^d}{\varepsilon^d} + \sum_{k=1}^{m_\varepsilon} \frac{\mathcal{M}\big(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]},\ \frac{\varepsilon_k}{L}\big) \cdot v_1\big(\frac{\varepsilon_k}{L}\big)^d}{\varepsilon_k^d} \\
&\leq \frac{v_1}{L^d} \left(\mathcal{N}\left(\mathcal{X}_\varepsilon,\ \frac{\varepsilon}{L}\right) + \sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]},\ \frac{\varepsilon_k}{L}\right) \right) \,,
\end{aligned}$$

where the first inequality follows by lower bounding $f(\boldsymbol{x}^\star) - f$ with its infimum on the partition, the second one by dropping $\varepsilon > 0$ from the second denominator and upper bounding the volume of a set with the volume of the balls of a smallest ε_k/L-cover, and the last one by the fact that covering numbers are always smaller than packing numbers (we recall this known result in the Supplementary Material, Lemma 1, Eq. (8)). This proves the first part of the theorem.

For the second one, we have

$$\begin{aligned}
\int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} &\geq \frac{\mathrm{vol}(\mathcal{X}_\varepsilon)}{(\varepsilon + \varepsilon)^d} + \sum_{k=1}^{m_\varepsilon} \frac{\mathrm{vol}\big(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]}\big)}{(\varepsilon_{k-1} + \varepsilon)^d} \\
&\geq \frac{1}{2^d} \frac{\mathrm{vol}(\mathcal{X}_\varepsilon)}{\varepsilon^d} + \frac{1}{4^d} \sum_{k=1}^{m_\varepsilon} \frac{\mathrm{vol}\big(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]}\big)}{\varepsilon_k^d} \geq \frac{1}{32^d} \left(\frac{\mathrm{vol}(\mathcal{X}_{2\varepsilon})}{\varepsilon^d} + \sum_{k=1}^{m_\varepsilon} \frac{\mathrm{vol}\left(\mathcal{X}_{\left(\frac{1}{2}\varepsilon_k,\, 2\varepsilon_{k-1}\right]}\right)}{\varepsilon_{k-1}^d} \right) \\
&\geq \frac{\mathcal{N}\left(\mathcal{X}_\varepsilon,\ \frac{\varepsilon}{L}\right) \mathrm{vol}\left(\frac{\varepsilon}{2L} B_1\right)}{(32\,\varepsilon)^d/\gamma} + \sum_{k=1}^{m_\varepsilon} \frac{\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]},\ \frac{\varepsilon_k}{L}\right) \mathrm{vol}\left(\frac{\varepsilon_k}{2L} B_1\right)}{(32\,\varepsilon_{k-1})^d/\gamma} \\
&\geq \gamma v_{1/64L} \mathcal{N}\left(\mathcal{X}_\varepsilon,\ \frac{\varepsilon}{L}\right) + \gamma v_{1/128L} \sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\, \varepsilon_{k-1}]},\ \frac{\varepsilon_k}{L}\right) \,,
\end{aligned}$$

where the first inequality follows by upper bounding $f(\boldsymbol{x}^\star) - f$ with its supremum on the partition, the second one by $\varepsilon \leq \varepsilon_{k-1}$ (for all $k \in [m_\varepsilon + 1]$) and $\varepsilon_{k-1} \leq 2\varepsilon_k$ (for all $k \in [m_\varepsilon]$), the third one by lower bounding the sum of disjoint layers with that of overlapping ones (proved in the Supplementary Material, Lemma 3), and the fourth one by the elementary inclusions $\mathcal{X}_{2\varepsilon} \supseteq \mathcal{X}_{\frac{3}{2}\varepsilon}$ and $\mathcal{X}_{\left(\frac{1}{2}\varepsilon_k,\, 2\varepsilon_{k-1}\right]} \supseteq \mathcal{X}_{\left(\frac{1}{2}\varepsilon_k,\, \frac{3}{2}\varepsilon_{k-1}\right]}$ (for all $k \in [m_\varepsilon]$) followed by a relationship between packing numbers and volumes (proved in the Supplementary Material, Proposition 4) that holds under Assumption 4. □

4 Optimality of $S_{\mathrm{C}}(f, \varepsilon)$

We begin this section by proving an f-dependent lower bound on the sample complexity of certified algorithms that matches the upper bound $S_{\mathrm{C}}(f, \varepsilon)$ on the sample complexity of the c.DOO algorithm (Proposition 1), up to a $\log(1/\varepsilon)$ term.

[2] We actually prove a stronger result. The first inequality holds more generally for any f that is L-Lipschitz around a maximizer and Lebesgue-measurable, and does not require $\mathcal{X}$ to satisfy Assumption 4.

The proof of this result differs from those of traditional worst-case lower bounds. The idea is to build a *local* worst-case analysis, introducing a weaker notion of sample complexity τ that is smaller than $\sigma(A, f, \varepsilon)$. Then we further lower bound this quantity by finding adversarial perturbations of the target function f with the following opposing properties. First, these perturbations have to be similar enough to f so that running A on them would return the same recommendations for sufficiently many rounds. Second, they have to be different enough from f so that enough rounds have to pass before being able to certify sub-ε accuracy. We recall that $m_\varepsilon := \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$.

Theorem 2. *The sample complexity of any certified algorithm A satisfies*

$$\sigma(A, f, \varepsilon) > \frac{c}{1 + m_\varepsilon} S_{\mathrm{C}}(f, \varepsilon)$$

for any Lipschitz function $f \in \mathcal{F}_L$ and all $\varepsilon \in (0, \varepsilon_0]$, where $c := 2^{-2}\big(2^{-7}\big(1 - \mathrm{Lip}(f)/L\big)\big)^d$.

Proof. Fix any $f \in \mathcal{F}_L$ and an accuracy $\varepsilon \in (0, \varepsilon_0]$. We begin by defining the tightest error certificate that a certified algorithm A could return based on its first n observations of f. Formally,

$$\mathrm{err}_n(A) := \sup\big\{\max(g) - g(\boldsymbol{x}_n^\star) : g \text{ is } L\text{-Lipschitz and } f(\boldsymbol{x}) = g(\boldsymbol{x}) \text{ for all } x \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}\big\}$$

where $\boldsymbol{x}_i = \boldsymbol{x}_i(A, f)$ and $\boldsymbol{x}_i^\star = \boldsymbol{x}_i^\star(A, f)$ are the query and recommendation points chosen by A at time i when run on f (to lighten the notation, we omit the explicit dependencies on A and f of $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^\star$). In particular, $\max(f) - f(\boldsymbol{x}_n^\star) \le \mathrm{err}_n(A)$. Based on this quantity, we define a corresponding notion of optimal sample complexity τ as the smallest number of rounds n that the best certified algorithm A' needs in order to guarantee that $\mathrm{err}_n(A') \le \varepsilon$. Formally,

$$\tau := \min\big\{n \in \mathbb{N}^* : \inf_{A'}\big(\mathrm{err}_n(A')\big) \le \varepsilon\big\} \ , \tag{7}$$

where the infimum is over all certified algorithms A'. It is immediate to prove that τ is finite, by considering an algorithm that queries a dense sequence of points (independently of the observed function values) and outputs a recommendation that maximizes the observed values.

Crucially, τ lower bounds the sample complexity $\sigma(A, f, \varepsilon)$ of any certified algorithm A. At a high level, this makes intuitive sense because τ guarantees a weaker property: that the best certificate of the best algorithm is small, while $\sigma(A, f, \varepsilon)$ controls the certificate of the specific algorithm A. We defer a formal proof of this statement to the Supplementary Material, Section D.1.

Since $\sigma(A, f, \varepsilon) \ge \tau$, to prove the theorem it is sufficient to show that

$$\tau > \frac{c}{1 + m_\varepsilon} S_{\mathrm{C}}(f, \varepsilon) \ .$$

Let $K := 16L/\big(L - \mathrm{Lip}(f)\big)$. If $S_{\mathrm{C}}(f, \varepsilon)/(1 + m_\varepsilon) < 3(8K)^d$, then $\tau \ge 1 > 3/4 > cS_{\mathrm{C}}(f, \varepsilon)/(1 + m_\varepsilon)$. Assume from now on that $S_{\mathrm{C}}(f, \varepsilon)/(1 + m_\varepsilon) \ge 3(8K)^d$.

The idea now is to upper bound the sum of $1 + m_\varepsilon$ packing numbers that define $S_{\mathrm{C}}(f, \varepsilon)$ in (5) with the largest one multiplied by $1 + m_\varepsilon$. This way, $S_{\mathrm{C}}(f, \varepsilon)/(1 + m_\varepsilon)$ can be controlled by (an upper bound of) the largest summand in $S_{\mathrm{C}}(f, \varepsilon)$. Let $\tilde{\varepsilon}$ be the scale achieving the maximum in (5), that is

$$\tilde{\varepsilon} = \begin{cases} \varepsilon \ , & \text{if } \mathcal{N}\left(\mathcal{X}_\varepsilon,\ \frac{\varepsilon}{L}\right) \ge \max_{i \in \{1, \dots, m_\varepsilon\}} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_i,\ \varepsilon_{i-1}]},\ \frac{\varepsilon_i}{L}\right) \ , \\ \varepsilon_{i^\star - 1} \ , & \text{otherwise, where } i^\star \in \mathrm{argmax}_{i \in \{1, \dots, m_\varepsilon\}} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_i,\ \varepsilon_{i-1}]},\ \frac{\varepsilon_i}{L}\right) \ . \end{cases}$$

Since $\mathcal{N}\left(\mathcal{X}_\varepsilon,\ \varepsilon/L\right) \le \mathcal{N}\left(\mathcal{X}_\varepsilon,\ \varepsilon/2L\right)$ and $\mathcal{N}\left(\mathcal{X}_{(\varepsilon_i,\ \varepsilon_{i-1}]},\ \varepsilon_i/L\right) \le \mathcal{N}\left(\mathcal{X}_{\varepsilon_{i-1}},\ \varepsilon_{i-1}/2L\right)$, we then have $S_{\mathrm{C}}(f, \varepsilon) \le (m_\varepsilon + 1)\mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ \tilde{\varepsilon}/2L\right)$. Let now $n \le cS_{\mathrm{C}}(f, \varepsilon)/(1 + m_\varepsilon)$. We then have $n \le c\mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ \tilde{\varepsilon}/2L\right)$. From a known property of packing numbers (Supplementary Material, Lemma 2),

$$\mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ \tfrac{K\tilde{\varepsilon}}{L}\right) \ge \left(\tfrac{1}{8K}\right)^d \mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ \tfrac{\tilde{\varepsilon}}{2L}\right) \ge \left(\tfrac{1}{8K}\right)^d \tfrac{S_{\mathrm{C}}(f, \varepsilon)}{m_\varepsilon + 1} \ge 3 \ ,$$

by our initial assumption. Then we have $n \le c(8K)^d \mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ K\tilde{\varepsilon}/L\right)$. Since $c(8K)^d = 1/4$, we thus obtain $n \le \mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ K\tilde{\varepsilon}/L\right) - 2$.

Consider a certified algorithm A for L-Lipschitz functions. Fix a $(K\tilde{\varepsilon}/L)$-packing $\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_N$ of $\mathcal{X}_{\tilde{\varepsilon}}$ with cardinality $N = \mathcal{N}\left(\mathcal{X}_{\tilde{\varepsilon}},\ K\tilde{\varepsilon}/L\right)$. Then the open balls of centers $\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_N$ and radius $K\tilde{\varepsilon}/2L$ are disjoint and two of them, with centers, say, $\tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_2$, do not contain any of the points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ queried by A when it is run on f. Let, for $\boldsymbol{x} \in \mathcal{X}$,

$$g_{\tilde{\varepsilon}}(\boldsymbol{x}) := \big(8\tilde{\varepsilon} - \tfrac{16L}{K}\,\|\boldsymbol{x} - \tilde{\boldsymbol{x}}_1\|\big)\mathbb{I}\big\{\boldsymbol{x} \in \mathcal{X} \cap B_{K\tilde{\varepsilon}/2L}(\tilde{\boldsymbol{x}}_1)\big\} \ .$$

Then $g_{\tilde{\varepsilon}}(x)$ is $16L/K = L - \mathrm{Lip}(f)$ Lipschitz. Hence $f + g_{\tilde{\varepsilon}}$ and $f - g_{\tilde{\varepsilon}}$ are L-Lipschitz. Note that f, $f + g_{\tilde{\varepsilon}}$ and $f - g_{\tilde{\varepsilon}}$ coincide on the points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ that A queries when it is run on f. As a consequence, A queries the same points and returns the same recommendation $\boldsymbol{x}_n^\star$ when it is run on any of the three functions f, $f + g_{\tilde{\varepsilon}}$, and $f - g_{\tilde{\varepsilon}}$.

Consider first the case $\boldsymbol{x}_n^\star \in B(\tilde{\boldsymbol{x}}_1, K\tilde{\varepsilon}/4L)$. Then, we have, by definition of $g_{\tilde{\varepsilon}}$ and the fact that $\tilde{\boldsymbol{x}}_2 \in \mathcal{X}_{\tilde{\varepsilon}}$, $f(\tilde{\boldsymbol{x}}_2) - g_{\tilde{\varepsilon}}(\tilde{\boldsymbol{x}}_2) - f(\boldsymbol{x}_n^\star) + g_{\tilde{\varepsilon}}(\boldsymbol{x}_n^\star) \geq -\tilde{\varepsilon} + 8\tilde{\varepsilon} - \frac{16L}{K}\frac{K\tilde{\varepsilon}}{4L} = 3\tilde{\varepsilon}$.

Now consider the case $\boldsymbol{x}_n^\star \notin B(\tilde{\boldsymbol{x}}_1, K\tilde{\varepsilon}/4L)$. Then, we have, by definition of $g_{\tilde{\varepsilon}}$ and the fact that $\tilde{\boldsymbol{x}}_1 \in \mathcal{X}_{\tilde{\varepsilon}}$, $f(\tilde{\boldsymbol{x}}_1) + g_{\tilde{\varepsilon}}(\tilde{\boldsymbol{x}}_1) - f(\boldsymbol{x}_n^\star) - g_{\tilde{\varepsilon}}(\boldsymbol{x}_n^\star) \geq -\tilde{\varepsilon} + 8\tilde{\varepsilon} - 8\tilde{\varepsilon} + \frac{16L}{K}\frac{K\tilde{\varepsilon}}{4L} = 3\tilde{\varepsilon}$.

Therefore, in both cases $\mathrm{err}_n(A) \geq 3\tilde{\varepsilon} > \varepsilon$. Repeating the same construction from any other certified algorithm A', we obtain that $\inf_{A'}\big(\mathrm{err}_n(A')\big) > \varepsilon$. Since this has been shown for any $n \leq cS_{\mathrm{C}}(f,\varepsilon)/(1+m_\varepsilon)$, by definition of τ we can conclude that $\tau > cS_{\mathrm{C}}(f,\varepsilon)/(1+m_\varepsilon)$. □

Putting together Proposition 1 and Theorem 2 shows that the sample complexity of c.DOO applied to any Lipschitz function $f \in \mathcal{F}_L$ is of order $S_{\mathrm{C}}(f,\varepsilon)$ and no certified algorithm A can do better, *not even if A knows f exactly*. This is a striking difference with the classical non-certified setting in which the best algorithm for each fixed function f has trivially sample complexity 1. In particular, for non-pathological sets $\mathcal{X}$, combining Proposition 1 and Theorem 2 with Theorem 1 gives the following compact characterization of the optimal sample complexity of certified algorithms.

Theorem 3. *Let $L > 0$ and suppose that Assumption 4 holds with $r_0 > \varepsilon_0/2L$. Then, there exist two constants $c', C > 0$ such that, for all Lipschitz functions $f \in \mathcal{F}_L$ and any accuracy $\varepsilon \in (0, \varepsilon_0]$, letting $c := c'(1 - \mathrm{Lip}(f)/L)^d/(1 + \log(\varepsilon_0/\varepsilon))$, the optimal sample complexity of any certified algorithm A satisfies*

$$c\int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d} \leq \inf_A \sigma(A, f, \varepsilon) \leq \sigma(\text{c.DOO}, f, \varepsilon) \leq C\int_{\mathcal{X}} \frac{\mathrm{d}\boldsymbol{x}}{\big(f(\boldsymbol{x}^\star) - f(\boldsymbol{x}) + \varepsilon\big)^d}\,.$$

For the sake of completeness, we conclude the section by briefly discussing the boundary case in which $L = \mathrm{Lip}(f)$, i.e., when the best Lipschitz constant of the target function is known exactly by the algorithm. As we mentioned earlier, this case is not of great relevance in practice, as one can hardly think of a scenario in which f is unknown but the learner has somehow perfect knowledge of its smallest Lipschitz constant $\mathrm{Lip}(f)$. It could, however, be of some theoretical interest.

The dependence of the sample complexity on $1 - L/\mathrm{Lip}(f)$ in Theorem 2 suggests that the problem becomes easier the closer L gets to $\mathrm{Lip}(f)$, and, when $L = \mathrm{Lip}(f)$, constant sample complexity could be achieved. The next result gives some formal evidence for this intuition in dimension $d \geq 2$.

Proposition 2. *Let $d \geq 2$, $\mathcal{X} := B_1$, and $\|\cdot\|$ be a norm. The certified Piyavskii-Shubert algorithm* PS *with initial guess $\boldsymbol{x}_1 := \mathbf{0}$ is a certified algorithm satisfying, for the L-Lipschitz function $f := L\,\|\cdot\|$ and any $\varepsilon \in (0, \varepsilon_0)$, $\sigma(\mathrm{PS}, f, \varepsilon) = 2 \ll c/\varepsilon^{d-1} \leq S_{\mathrm{C}}(f,\varepsilon)$, for some constant $c > 0$.*

For a full proof and a refresher on the PS algorithm, see Section D.2 of the Supplementary Material. The next result shows that, surprisingly, the sample complexity in dimension $d = 1$ behaves differently. In this case, $S_{\mathrm{C}}(f,\varepsilon)$ remains a lower bound even when $L = \mathrm{Lip}(f)$. More precisely, the following proposition holds. Its proof is deferred to Section D.3 of the Supplementary Material.

Proposition 3. *If $d = 1$, let $c = 2^{-8}/3$. Then, the sample complexity of any certified algorithm A satisfies, for any L-Lipschitz function f and all $\varepsilon \in (0, \varepsilon_0]$, $\sigma(A, f, \varepsilon) > cS_{\mathrm{C}}(f,\varepsilon)/(1+m_\varepsilon)$.*

5 Conclusion

In this paper, we studied the sample complexity of certified zeroth-order Lipschitz optimization. We first showed that the sample complexity of the computationally tractable c.DOO algorithm scales with the quantity $S_{\mathrm{C}}(f,\varepsilon)$ introduced in Eq. (5) (Proposition 1). We then characterized this quantity in terms of the compact integral expression $\int_{\mathcal{X}} \mathrm{d}\boldsymbol{x}/(\max(f) - f(\boldsymbol{x}) + \varepsilon)^d$ (Theorem 1), solving as a corollary a long-standing open problem in Hansen et al. (1991). Finally we proved an instance-dependent lower bound (Theorem 2) showing that this integral characterizes (up to $\log$ factors) the optimal sample complexity of certified zeroth-order Lipschitz optimization in any dimensions $d \geq 1$ whenever the smallest Lipschitz constant of f is not known exactly by the algorithm (Theorem 3).

References

Dario Azzimonti, David Ginsbourger, Clément Chevalier, Julien Bect, and Yann Richet. 2021. Adaptive design of experiments for conservative estimation of excursion sets. *Technometrics* 63, 1 (2021), 13–26.

Peter L. Bartlett, Victor Gabillon, and Michal Valko. 2019. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory (Proceedings of Machine Learning Research, Vol. 98)*, Aurélien Garivier and Satyen Kale (Eds.). PMLR, Chicago, Illinois, 184–206.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *25th annual conference on neural information processing systems (NIPS 2011)*, Vol. 24. Neural Information Processing Systems Foundation, Neural Information Processing Systems Foundation, Granada, Spain, 2546–2554.

Clément Bouttier, Tommaso R Cesari, and Sébastien Gerchinovitz. 2020. Regret analysis of the Piyavskii-Shubert algorithm for global Lipschitz optimization. (Feb. 2020). Preprint.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge University Press, Cambridge.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. 2011. X-armed bandits. *Journal of Machine Learning Research* 12, May (2011), 1655–1695.

Antonio Cuevas, Ricardo Fraiman, and Beatriz Pateiro-López. 2012. On statistical properties of sets fulfilling rolling-type conditions. *Advances in Applied Probability* 44, 2 (2012), 311–329.

Pierre Hansen, Brigitte Jaumard, and S-H Lu. 1991. On the number of iterations of Piyavskii's global optimization algorithm. *Mathematics of Operations Research* 16, 2 (1991), 334–350.

Matthias Horn. 2006. Optimal algorithms for global optimization in case of unknown Lipschitz constant. *Journal of Complexity* 22, 1 (2006), 50–70.

Yichun Hu, Nathan Kallus, and Xiaojie Mao. 2020. Smooth Contextual Bandits: Bridging the Parametric and Non-differentiable Regret Regimes. In *Proceedings of Thirty Third Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 125)*, Jacob Abernethy and Shivani Agarwal (Eds.). PMLR, Online, 2007–2010.

Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. 2008. Multi-Armed Bandits in Metric Spaces. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing* (Victoria, British Columbia, Canada) *(STOC '08)*. Association for Computing Machinery, New York, NY, USA, 681–690.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. 2019. Bandits and experts in metric spaces. *Journal of the ACM (JACM)* 66, 4 (2019), 1–77.

Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press, Cambridge.

Cédric Malherbe and Nicolas Vayatis. 2017. Global optimization of Lipschitz functions. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. PMLR 70. JMLR.org, Sydney, NSW, Australia, 2314–2323.

Rémi Munos. 2011. Optimistic Optimization of a Deterministic Function without the Knowledge of Its Smoothness. In *Proceedings of the 24th International Conference on Neural Information Processing Systems* (Granada, Spain) *(NIPS'11)*. Curran Associates Inc., Red Hook, NY, USA, 783–791.

Rémi Munos. 2014. From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning. *Foundations and Trends in Machine Learning* 7, 1 (2014), 1–130.

Yurii Nesterov. 2003. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, New York, US.

A.G. Perevozchikov. 1990. The complexity of the computation of the global extremum in a class of multi-extremum problems. *U. S. S. R. Comput. Math. and Math. Phys.* 30, 2 (1990), 28–33.

S.A. Piyavskii. 1972. An algorithm for finding the absolute extremum of a function. *U. S. S. R. Comput. Math. and Math. Phys.* 12, 4 (1972), 57–67.

Yann Richet, Grégory Caplin, Jérôme Crevel, David Ginsbourger, and Victor Picheny. 2013. Using the efficient global optimization algorithm to assist nuclear criticality safety assessment. *Nuclear Science and Engineering* 175, 1 (2013), 1–18.

Bruno O Shubert. 1972. A sequential method seeking the global maximum of a function. *SIAM J. Numer. Anal.* 9, 3 (1972), 379–388.

Aleksandrs Slivkins. 2019. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286. `https://doi.org/10.1561/2200000068`

Charlie Vanaret, Jean-Baptiste Gotteland, Nicolas Durand, and Jean-Marc Alliot. 2013. Preventing premature convergence and proving the optimality in evolutionary algorithms. In *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, Springer International Publishing, Cham, 29–40.

Martin J. Wainwright. 2019. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, Cambridge, United Kingdom.

Guenther Walther. 1997. Granulometric smoothing. *The Annals of Statistics* 25, 6 (1997), 2273–2299.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The outline in Section 1.2 provides pointers to where the claimed contributions of the paper are provided.
   (b) Did you describe the limitations of your work? [Yes] In particular, the outline Section 1.2 contains a paragraph entitled "Limitations and open problems".
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This is a theoretical/foundation work that adds to the theory and methodology of optimization. As for any such contributions, the positive or negative societal impact will depend on the application case. We do not promote any harmful use of this theory, but we expand on the existing knowledge.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Se previous question.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All our results explicitly refer to their required assumptions. Some general assumptions that hold throughout the paper are also stated in the introduction.
   (b) Did you include complete proofs of all theoretical results? [Yes] Some technical details are deferred to the Supplementary Material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] We did not ran experiments.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] We did not use existing assets (code, data or models) nor cure/release new assets (code, data or models).
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing nor conducted research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]