Fixed Dimension Matters: Tight Complexity and Computational Hardness for Higher-Order Smooth Nonconvex Stationarity

Anonymous authors
Paper under double-blind review

Abstract

We study approximate first-order stationarity for nonconvex functions whose derivatives of orders $1, \dots, p$ have prescribed Lipschitz constants $L_1, \dots, L_p$, and whose initial function-value gap is at most Δ. For every **fixed** dimension d and fixed order p, we establish a tight oracle bound $\Theta(\epsilon^{-(p+1)/p})$ of finding ϵ-stationary points with deterministic exact gradient-only oracles. We also show that, for every fixed finite $p \geq 1$ and fixed $d \geq 2$, stationary-point search is PLS-complete for bounded functions with bounded derivatives through order p and Lipschitz p-th derivative.

1 Introduction

Finding a point $x \in \mathbb{R}^d$ with $\|\nabla f(x)\| \leq \epsilon$ is a basic task in smooth nonconvex optimization. Unlike global optimality, this local criterion can be certified by a single gradient evaluation. Its complexity nevertheless depends on what information an algorithm may request: a *gradient-only* oracle returns $\nabla f(x)$, whereas a *value-and-gradient* oracle also returns $f(x)$. Classical gradient descent needs only the former and attains $O(\epsilon^{-2})$ queries under normalized gradient-Lipschitz and initial-gap bounds (Nesterov, 2004). In sufficiently large dimension, matching lower bounds hold even with richer derivative information (Carmon et al., 2020).

Why isolate gradient information? Our motivation is to determine what stationarity guarantees follow from first-derivative information alone, rather than from an additional ability to compare objective values. This distinction matters for methods whose updates and stopping rules use gradients but no value-based acceptance test. It is also a genuine information-theoretic distinction, not merely a change of implementation: in one dimension, Chewi et al. (2023) prove deterministic complexity $\Theta(\epsilon^{-2})$ with gradients only, but $\Theta(\log(1/\epsilon))$ when values are also available. Thus a low-dimensional value-and-gradient upper bound cannot simply be transferred to a value-free algorithm. Conversely, a lower bound for the richer oracle also applies to gradients only, whereas the reverse implication is invalid.

Higher-order smoothness makes this question particularly informative. Nearby exact gradient evaluations can approximate higher derivative tensors, but a regularized model must still produce progress without checking objective values. Known smoothness budgets allow such progress to be certified analytically. We therefore study exact gradient queries with known budgets, and charge only the queries, not internal model minimization. This model isolates the information requirement; it does not claim that values are unavailable in every application, nor that high-order model subproblems are computationally free in practice. Objective-function-free regularization is an established algorithmic framework (Gratton et al., 2025); our question concerns its sharp fixed-dimensional information complexity under simultaneous smoothness constraints.

Why fix the dimension? For an objective whose derivatives of orders $1, \dots, p$ are all Lipschitz, Carmon et al. (2021) establish deterministic first-order lower bounds $\Omega(\epsilon^{-12/7})$ for $p = 2$ and $\Omega(\epsilon^{-8/5})$ for fixed $p \geq 3$. The recent preprint of Zhou (2026) reports the sharper exponents $7/4$ and $5/3$, respectively. The former matches the upper bound of Li & Lin

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

(2022); for $p \geq 3$, the known upper bound retains a logarithmic factor (Carmon et al., 2017). These lower-bound constructions allow the dimension to grow with the required accuracy. They do not characterize the complexity when d is fixed and constants depending on d are allowed. Low-dimensional value-and-gradient results already demonstrate the importance of this distinction (Chewi et al., 2023; Hollender & Zampetakis, 2025). In two dimensions, the bounded Hessian-Lipschitz construction in Kontogiannis et al. (2024) also gives an $\Omega(\epsilon^{-1})$ first-order-stationarity lower bound. We ask:

What is the tight deterministic gradient-only complexity of approximate stationarity under simultaneous higher-order smoothness in fixed dimension?

A complementary question concerns *computation* rather than information. When value and gradient evaluators are given as finite descriptions, Hollender & Zampetakis (2025) prove PLS-completeness in every fixed dimension at least two. The Hessian-Lipschitz case is also already covered by Kontogiannis et al. (2024). We investigate whether this barrier persists at every fixed smoothness order, while preserving polynomial-time evaluation of the constructed functions. This white-box question is separate from our exact-gradient oracle model.

1.1 Our Contributions

1. **Tight gradient-only query complexity.** Our oracle characterization tracks all simultaneous smoothness budgets. Let
$$R_\epsilon = \Delta \min_{1 \leq q \leq p} L_q^{1/q} \epsilon^{-(q+1)/q}.$$
Theorem 3.1 proves upper and lower bounds $A_{d,p}(1+R_\epsilon)$ and $a_p R_\epsilon - b_p$. Hence the complexity is $\Theta_{d,p}(R_\epsilon)$ above an order-dependent constant threshold, and is $\Theta_{d,p}(\Delta L_p^{1/p} \epsilon^{-(p+1)/p})$ as $\epsilon \downarrow 0$ with positive budgets fixed. The upper bound reconstructs approximate derivative tensors from a constant-size gradient stencil at each iteration and selects the best order q. Global model minimization is permitted as uncharged internal computation. The normalized upper exponent is consistent with the existing objective-function-free theory (Gratton et al., 2025); our direct proof makes the full parameter dependence explicit.
2. **A universal one-dimensional lower bound.** The matching lower bound is one-dimensional and applies to *every* deterministic gradient-query rule, including algorithms that query distant points or output unqueried points. We realize a finite constant-gradient transcript using a smooth plateau whose support measure controls the initial gap. Its width enforces every derivative-Lipschitz budget simultaneously. The entire construction and its proof appear together in Appendix C.2.
3. **Computational hardness at every fixed smoothness order.** For the finite-description problem, Theorem 4.2 establishes PLS-completeness of a totalized stationary-point problem, with hardness on the promised bounded higher-smoothness class for every fixed p and $d \geq 2$. Starting from the hard family of Hollender & Zampetakis (2025), a polynomial kernel preserves stationary-point decoding and permits exact rational evaluation of the smoothed function and its gradient. The contribution beyond the known $p = 1, 2$ cases is this effective extension to arbitrary fixed finite order. Appendix D provides the analytic and encoding details.

Compared with the other bounds in Table 1, our result exhibits a continuing benefit from higher-order smoothness: the tight fixed-dimensional gradient-only exponent $1 + 1/p$ strictly decreases with p, rather than remaining unchanged for $p \geq 3$ as in the growing-dimensional bounds shown there.

1.2 Related work

High-order models and objective-function-free algorithms. Cubic and higher-order regularization yield the exponent $(p+1)/p$ when sufficiently accurate derivative models are available (Nesterov & Polyak, 2006; Birgin et al., 2017; Cartis et al., 2020). Finite-difference implementations and approximate Hessians have been studied extensively (Cartis et al., 2012; Grapiglia et al., 2022). The inexact objective-function-free framework of Gratton

Domain / oracle	Order	Upper bound	Lower bound
$\mathbb{R}$; gradients only	$p = 1$	$O(\epsilon^{-2})$ (Nesterov, 2004)	$\Omega(\epsilon^{-2})$ (Chewi et al., 2023)
$\mathbb{R}$; values and gradients	$p = 1$	$O(\log(1/\epsilon))$ (Chewi et al., 2023)	$\Omega(\log(1/\epsilon))$ (Chewi et al., 2023)
$\mathbb{R}^2$; bounded objective; values and gradients	$p = 1$	$O(\epsilon^{-1})$ (Hollender & Zampetakis, 2025)	$\Omega(\epsilon^{-1})$ (Hollender & Zampetakis, 2025)
$\mathbb{R}^2$; bounded objective; values and gradients	$p = 2$	$O(\epsilon^{-1})$ (Hollender & Zampetakis, 2025)	$\Omega(\epsilon^{-1})$ (Kontogiannis et al., 2024)
Growing dimension; values and gradients	$p = 1$	$O(\epsilon^{-2})$ (Nesterov, 2004)	$\Omega(\epsilon^{-2})$ (Carmon et al., 2020)
Growing dimension; values and gradients	$p = 2$	$O(\epsilon^{-7/4})$ (Li & Lin, 2022)	Earlier: $\Omega(\epsilon^{-12/7})$ (Carmon et al., 2021); sharper: $\Omega(\epsilon^{-7/4})$ (Zhou, 2026)
Growing dimension; values and gradients	fixed $p \geq 3$	$\widetilde{O}(\epsilon^{-5/3})$ (Carmon et al., 2017)	Earlier: $\Omega(\epsilon^{-8/5})$ (Carmon et al., 2021); sharper: $\Omega(\epsilon^{-5/3})$ (Zhou, 2026)
$[0, 1]^2$; values and gradients	$p = 1$	$O(\epsilon^{-1/2})$ (Hollender & Zampetakis, 2025)	$\Omega(\epsilon^{-1/2})$ (Vavasis, 1993) Bubeck & Mikulincer (2020)
Fixed dimension; gradients only (**ours**)	fixed $p \geq 1$	$O_{d,p}(\epsilon^{-(p+1)/p})$ (Prop. 3.3)	$\Omega_p(\epsilon^{-(p+1)/p})$ (Prop. 3.5)

Table 1: Deterministic query bounds as $\epsilon \downarrow 0$. Unconstrained initial-gap rows use fixed normalized smoothness and gap budgets; the bounded $\mathbb{R}^2$ rows additionally assume uniformly bounded values (their hard families are also Lipschitz). In the $p = 2$ row, the cited proof first establishes the first-order-stationarity lower bound and then deduces the second-order result; its local evaluator also supplies Hessians, so the bound applies to the weaker value-and-gradient oracle. The constrained $[0, 1]^2$ row concerns a different, boundary-aware target and is not an unconstrained lower bound. Growing-dimensional lower bounds allow d to depend on ϵ. The tilde retains logarithmic factors. The sharper higher-order lower bounds are those reported in Zhou (2026).

et al. (2025) permits deterministic derivative approximations; in fixed dimension, a fixed-size gradient stencil gives the normalized upper exponent. Doikov & Grapiglia (2026) study lower-order implementations of higher-order methods, using an oracle that supplies values and derivatives through order $p - 1$. Dimension-dependent gradient/Hessian tradeoffs are also studied by Adil et al. (2025). These oracle and dimension distinctions prevent an upper bound from being transferred without checking its information assumptions.

Lower bounds and computational hardness. The one-dimensional gradient-only construction of Chewi et al. (2023) is the direct $p = 1$ predecessor of our lower bound. The one-dimensional higher-order sharp examples in Cartis et al. (2020) concern specified model-based algorithm classes; our lower bound has no such restriction. Table 1 and Appendix A separate these statements from value-and-gradient and compact-domain results (Carmon et al., 2021; 2020; Vavasis, 1993; Bubeck & Mikulincer, 2020; Hollender & Zampetakis, 2025; Kontogiannis et al., 2024). For finite descriptions, PLS formalizes discrete local search with efficient potential and neighborhood evaluation (Johnson et al., 1988). Our reduction extends the smooth hard family of Hollender & Zampetakis (2025) to every fixed finite order without reclaiming the Hessian-Lipschitz case already established by Kontogiannis et al. (2024).

2 Preliminaries

All vector norms are Euclidean; $|S|$ denotes the cardinality of a finite set S. For a scalar-valued function, $D^q f(x)$ denotes its symmetric q-linear derivative tensor, with operator norm $\|D^q f(x)\|_{\mathrm{op}} = \sup_{\|u_1\|=\cdots=\|u_q\|=1} |D^q f(x)[u_1, \ldots, u_q]|$. We identify $D^1 f$ with ∇f, set

$D^0 f = f$, and write $D^q f(x)[s]^q = D^q f(x)[s, \dots, s]$. For a tensor field A, $\|A\|_\infty$ denotes the supremum of its pointwise norm (absolute value for scalars, Euclidean norm for vectors, and operator norm for tensors). We use $f^{(q)}$ for univariate derivatives. Throughout the oracle analysis, the initial point is translated to $x_0 = 0$.

Definition 2.1 (Simultaneous higher-order smoothness)**.** Fix $d \geq 1$ and positive numbers Δ, L_q. The class $\mathcal{F}_q(\Delta, L_q)$ consists of the functions $f \in C^q(\mathbb{R}^d)$ such that

$$f(0) - \inf_{x \in \mathbb{R}^d} f(x) \leq \Delta, \qquad \|D^q f(x) - D^q f(y)\|_{\mathrm{op}} \leq L_q \|x - y\| \quad (x, y \in \mathbb{R}^d).$$

For an integer $p \geq 1$, set

$$\mathcal{F}_{1:p}(\Delta, L_1, \dots, L_p) = \bigcap_{q=1}^{p} \mathcal{F}_q(\Delta, L_q).$$

Dimension is implicit when clear from the context. We write $C^{q,1}$ for C^q functions with a globally Lipschitz q-th derivative.

If $f \in \mathcal{F}_{1:p}$, then $\|D^r f(x)\|_{\mathrm{op}} \leq L_{r-1}$ for $2 \leq r \leq p$, by differentiating the Lipschitz bound for $D^{r-1} f$. Membership in $\mathcal{F}_q$ alone does not impose the separate lower-order Lipschitz bounds when $q > 1$.

A deterministic exact gradient-only oracle algorithm receives $\nabla f(x)$ at each adaptively chosen point $x \in \mathbb{R}^d$. The parameters $d, p, \epsilon, \Delta, L_1, \dots, L_p$ are known. It receives neither function values nor higher derivatives. It may carry out arbitrary exact internal computations at no query cost, and may output a point that it has not queried. Its worst-case complexity is the maximum number of gradient queries needed to return an ϵ-stationary point for every function in the class. Let $T^{\nabla}_{d,p}(\epsilon; \Delta, L_1, \dots, L_p)$ denote the infimum of this worst-case cost over successful deterministic algorithms. An algorithm that fails on an admissible function has infinite worst-case cost.

Define the intrinsic length and work scales by

$$\rho_\epsilon = \max_{1 \leq q \leq p} (\epsilon / L_q)^{1/q}, \qquad R_\epsilon = \frac{\Delta}{\epsilon \rho_\epsilon} = \Delta \min_{1 \leq q \leq p} L_q^{1/q} \epsilon^{-(q+1)/q}. \tag{1}$$

The length ρ_ϵ is, up to order-dependent constants, the width required for a gradient plateau of height ϵ to respect all the smoothness constraints. Constants in $O_{d,p}$, $\Omega_{d,p}$, and $\Theta_{d,p}$ may depend on d, p, but not on $\epsilon, \Delta, L_1, \dots, L_p$.

Definition 2.2 (Bounded higher-order smoothness)**.** For positive finite budgets $M_0, \dots, M_p, \Lambda_p$, write

$$\mathcal{B}^{p,1}(M_0, \dots, M_p; \Lambda_p)$$

for the functions $f \in C^p(\mathbb{R}^d)$ satisfying

$$\sup_x |f(x)| \leq M_0, \qquad \sup_x \|D^r f(x)\|_{\mathrm{op}} \leq M_r \quad (1 \leq r \leq p),$$

and $\|D^p f(x) - D^p f(y)\|_{\mathrm{op}} \leq \Lambda_p \|x - y\|$. The unparameterized class $\mathcal{B}^{p,1}(\mathbb{R}^d)$ is the union of these classes over all finite budgets.

Remark 2.3 (Parameter-dependent inclusion). For every fixed choice of budgets,

$$\mathcal{B}^{p,1}(M_0, \dots, M_p; \Lambda_p) \subseteq \mathcal{F}_{1:p}(2M_0, M_2, \dots, M_p, \Lambda_p),$$

where the right side means $\mathcal{F}_1(2M_0, \Lambda_1)$ when $p = 1$. Indeed, boundedness gives a gap at most $2M_0$, and the fundamental theorem of calculus gives $\mathrm{Lip}(D^q f) \leq M_{q+1}$ for $q < p$. The unparameterized union $\mathcal{B}^{p,1}(\mathbb{R}^d)$ is not contained in a class with one predetermined finite parameter tuple. This distinction matters when relating oracle bounds to finite-description inputs whose numerical budgets vary with the instance.

3 Tight oracle complexity in fixed dimension

The next theorem states the result uniformly in the gap and all smoothness budgets. The additive terms account for the small-work regime, in which an unqualified multiplicative Θ statement need not hold.

Theorem 3.1 (Fixed-dimensional gradient-only complexity)**.** *For every fixed pair of integers $d, p \geq 1$, there exist constants $a_p > 0$, $b_p \geq 0$, and $A_{d,p} > 0$ such that, for all positive $\epsilon, \Delta, L_1, \ldots, L_p$,*

$$a_p R_\epsilon - b_p \leq T^{\nabla}_{d,p}(\epsilon; \Delta, L_1, \ldots, L_p) \leq A_{d,p}(1 + R_\epsilon). \tag{2}$$

In particular, for a constant $\kappa_p > 0$ depending only on p and every $R_\epsilon \geq \kappa_p$,

$$T^{\nabla}_{d,p}(\epsilon; \Delta, L_1, \ldots, L_p) = \Theta_{d,p}\left(\Delta \min_{1 \leq q \leq p} L_q^{1/q} \epsilon^{-(q+1)/q}\right).$$

For fixed positive $\Delta, L_1, \ldots, L_p$, this specializes, as $\epsilon \downarrow 0$, to

$$T^{\nabla}_{d,p}(\epsilon; \Delta, L_1, \ldots, L_p) = \Theta_{d,p}\big(\Delta L_p^{1/p} \epsilon^{-(p+1)/p}\big).$$

The constants in the lower bound can be chosen independently of d.

Remark 3.2 (Information and computation). Theorem 3.1 is a statement about exact gradient information. It places no restriction on the distance of a query from the origin, on query precision, or on the cost of solving an internal regularized model. The number of stencil points depends on the fixed dimension and order. Consequently, the upper bound is neither a dimension-free guarantee nor a polynomial bit-time bound. The finite-description problem in Section 4 uses a different computational model.

3.1 Upper bound: finite-difference high-order regularization

We obtain an upper bound by combining high-order regularization with finite-difference reconstruction of derivative tensors from exact gradient queries. The regularization framework includes cubic regularization (Nesterov & Polyak, 2006) and its higher-order extensions (Birgin et al., 2017). Finite-difference implementations of cubic regularization have a substantial history (Cartis et al., 2012; Grapiglia et al., 2022), and lower-order implementations of higher-order methods have also been studied (Doikov & Grapiglia, 2026). In particular, the objective-function-free inexact-derivative framework of Gratton et al. (2025), specialized to deterministic derivatives and combined with fixed-size gradient stencils, already implies the normalized fixed-dimensional upper exponent. Our purpose here is to give a self-contained gradient-only argument with the uniform parameter dependence needed for Theorem 3.1.

For a selected order q, the method reconstructs the tensors $D^r f(x)$, $2 \leq r \leq q$, to accuracies determined by ϵ and L_q. It then minimizes an approximate Taylor model regularized by a term of degree $q+1$. Known smoothness constants make function-value evaluations unnecessary: every step whose endpoint is not already ϵ-stationary has a guaranteed objective decrease.

Proposition 3.3 (Gradient-only upper bound)**.** *For every pair of fixed integers $d, p \geq 1$, there is a constant $C_{d,p} > 0$, depending only on d, p, such that*

$$T^{\nabla}_{d,p}(\epsilon; \Delta, L_1, \ldots, L_p) \leq C_{d,p}\left(1 + \Delta \min_{1 \leq q \leq p} L_q^{1/q} \epsilon^{-(q+1)/q}\right).$$

Algorithm 1 attains this bound using exact gradient queries. Consequently, when $\Delta/(\epsilon\rho_\epsilon) \geq 1$, the additive constant can be absorbed into $O_{d,p}\big(\Delta/(\epsilon\rho_\epsilon)\big)$.

The proof is given in Appendix C.1. Its two ingredients are the regularized-step estimate in Lemma C.2 and the finite-difference estimate in Lemma C.3. Lemma C.4 combines them while controlling the approximation errors uniformly over the smoothness parameters.

The algorithm is stated in the exact-real oracle model. Its model minimization is internal computation and is not charged as an oracle query. The regularized model is coercive, so its global minimizer exists; we use a fixed deterministic tie-breaking rule when it is not unique.

The oracle bound alone does not give a bound on the arithmetic work or the bit complexity of this internal minimization.

Algorithm 1 Finite-difference high-order regularization (FD-HOR)

Require: $\epsilon > 0$, $L_1, \dots, L_p > 0$, $x_0 = 0$, and an exact gradient oracle $\mathcal{O}_\nabla$.
Ensure: A point x satisfying $\|\nabla f(x)\| \le \epsilon$.

Choose $q_\star \in \arg\min_{1 \le q \le p} L_q^{1/q} \epsilon^{-(q+1)/q}$, breaking ties deterministically; set $q \leftarrow q_\star$.
if $q = 1$ **then**
  for $t = 0, 1, 2, \dots$ **do**
    Query $g_t \leftarrow \mathcal{O}_\nabla(x_t)$.
    if $\|g_t\| \le \epsilon$ **then**
      return x_t.
    end if
    Set $x_{t+1} \leftarrow x_t - L_1^{-1} g_t$.
  end for
end if
Choose $\nu_{d,q} > 0$ as in Lemma C.4; set $h \leftarrow \nu_{d,q}(\epsilon/L_q)^{1/q}$ and $\sigma \leftarrow 2L_q$.
for $t = 0, 1, 2, \dots$ **do**
  Query $g_t \leftarrow \mathcal{O}_\nabla(x_t)$.
  if $\|g_t\| \le \epsilon$ **then**
    return x_t.
  end if
  $(G_{2,t}, \dots, G_{q,t}) \leftarrow$ FD-Tensors$(x_t, q, h, \mathcal{O}_\nabla)$ using Algorithm 2.
  Form the model

$$\widetilde{m}_t(s) = \langle g_t, s \rangle + \sum_{r=2}^{q} \frac{1}{r!} G_{r,t}[s]^r + \frac{\sigma}{(q+1)!} \|s\|^{q+1}.$$

  Compute $s_t \in \arg\min_{s \in \mathbb{R}^d} \widetilde{m}_t(s)$.
  Set $y_t \leftarrow x_t + s_t$ and query $g_t^+ \leftarrow \mathcal{O}_\nabla(y_t)$.
  if $\|g_t^+\| \le \epsilon$ **then**
    return y_t.
  end if
  Set $x_{t+1} \leftarrow y_t$.
end for

Here $G_{r,t} = G_r(x_t)$ denotes the reconstruction in Appendix B. The last gradient query explicitly checks the candidate endpoint. It can be reused as the current gradient at the next iteration. For fixed d, q, both this check and the entire finite-difference stencil have constant query cost. Selecting $q_\star$ gives the best dependence on the available smoothness bounds, up to constants depending only on d, p.

3.2 Lower bound: a smooth plateau resisting oracle

We construct a one-dimensional resisting oracle and then embed it into $\mathbb{R}^d$. The constant-reply mechanism extends Chewi et al. (2023), while the covering scale enforces all derivative budgets simultaneously. Unlike the high-order examples for specified regularization schemes in Cartis et al. (2020), the construction applies to every deterministic gradient-only query rule. Queries may be arbitrarily far apart, and the final output need not have been queried. The following sketches explain why neither freedom defeats the lower bound; complete proofs are in Appendix C.2.

Proposition 3.4 (One-dimensional gradient-only lower bound)**.** *For each fixed integer $p \ge 1$, there exist constants $c_p > 0$ and $C_p \ge 0$, depending only on p, such that every deterministic gradient-only algorithm which finds an ϵ-stationary point of every function in $\mathcal{F}_{1:p}(\Delta, L_1, \dots, L_p)$ on $\mathbb{R}$ has worst-case gradient-query complexity at least*

$$c_p \frac{\Delta}{\epsilon \rho_\epsilon} - C_p, \qquad \rho_\epsilon = \max_{1 \le q \le p} (\epsilon/L_q)^{1/q}.$$

In particular, there is a constant $\kappa_p > 0$, depending only on p, such that, whenever $\Delta/(\epsilon\rho_\epsilon) \geq \kappa_p$, the lower bound is

$$\Omega_p\left(\frac{\Delta}{\epsilon\rho_\epsilon}\right) = \Omega_p\left(\Delta \min_{1\leq q\leq p} L_q^{1/q}\epsilon^{-(q+1)/q}\right).$$

The algorithm may output a point which it has not queried.

Proof sketch of Proposition 3.4. Hard-instance idea: charge for local coverage, not for travel distance. Fix a budget N and answer every derivative query by 2ϵ. These replies determine a finite transcript. If the algorithm outputs within the budget, include its output in the covered set S, even if unqueried; otherwise stop just before its $(N+1)$-st answer. In either case, $|S| \leq N+1$. We only need to realize this finite transcript, not an infinite constant derivative, which would violate the gap condition.

Lemma C.5 supplies a smooth plateau ϕ, equal to one on S, with values in $[0,1]$, support measure $O_p((N+2)\rho)$, and $\left\|\phi^{(q)}\right\|_\infty \leq K_p\rho^{-q}$. It convolves the indicator of a union of neighborhoods with a probability kernel. Using a union rather than summing individual bumps prevents overlaps from amplifying the derivative bounds. Choose $\rho = \rho_\epsilon^* = C_p^*\rho_\epsilon$, with C_p^* a sufficiently large order-dependent constant, and set

$$f(x) = 2\epsilon\int_{-\infty}^{x}\phi(t)\,dt, \qquad \left\|f^{(q+1)}\right\|_\infty \leq 2\epsilon K_p(\rho_\epsilon^*)^{-q} \leq L_q \quad (1\leq q\leq p).$$

The maximum defining ρ_ϵ is precisely what makes one plateau respect *all* the smoothness budgets. Since $f \geq 0$ and vanishes sufficiently far to the left, its initial gap is at most its total increase:

$$f(0) - \inf f \leq 2\epsilon\int_{\mathbb{R}}\phi(t)\,dt \leq B_p(N+2)\epsilon\rho_\epsilon, \qquad B_p = 2K_pC_p^*.$$

Crucially, large empty gaps between queries contribute nothing to this integral. Thus each covered point costs only $O_p(\epsilon\rho_\epsilon)$, regardless of its location. Whenever this total fits within Δ, the constructed function is admissible and reproduces the transcript exactly. An early output has derivative 2ϵ and fails; otherwise the algorithm exceeds N queries on a valid objective. Taking N proportional to R_ϵ, with the constant-size correction in (17), proves the lower bound.

The same scale explains the match with the upper bound. At order q, a nonterminal regularized step decreases the objective by a constant multiple of $\epsilon\tau_q$, where $\tau_q = (\epsilon/L_q)^{1/q}$, by Lemma C.4. Selecting the best order maximizes τ_q, giving ρ_ϵ. The lower construction spends that same amount of gap per covered point. Thus the upper and lower bounds are controlled by the same geometric length, not merely by coincident powers of ϵ. The construction only fixes derivative replies: matching additional function-value replies would impose new integral constraints, so this argument does not establish the same lower bound for a value-and-gradient oracle. □

Proposition 3.5 (Dimension-d lower bound)**.** *For every integer $d \geq 1$ and each fixed integer $p \geq 1$,*

$$T_{d,p}^{\nabla}(\epsilon;\Delta,L_1,\ldots,L_p) \geq c_p\frac{\Delta}{\epsilon\rho_\epsilon} - C_p,$$

where the constants may be taken to be those of Proposition 3.4, independently of d. Consequently, for $\Delta/(\epsilon\rho_\epsilon) \geq \kappa_p$,

$$T_{d,p}^{\nabla}(\epsilon;\Delta,L_1,\ldots,L_p) = \Omega_p\left(\Delta \min_{1\leq q\leq p} L_q^{1/q}\epsilon^{-(q+1)/q}\right).$$

Proof sketch of Proposition 3.5. Extra coordinates need not reveal extra information. For a one-dimensional objective φ, set

$$f(x) = \varphi(x_1), \qquad \nabla f(x) = \varphi'(x_1)e_1, \qquad \|\nabla f(x)\| = |\varphi'(x_1)|.$$

The initial gap is unchanged. Moreover, $D^q f(x) = \varphi^{(q)}(x_1) e_1^{\otimes q}$; the tensor $e_1^{\otimes q}$ has operator norm one, so every Lipschitz budget L_q is preserved without a dimension factor. Simulate any deterministic d-dimensional algorithm by one scalar derivative query at x_1 per gradient query, appending zeros to the answer. Its choices of the other coordinates provide no new information about φ. Projecting its final output onto the first coordinate preserves stationarity, including for an unqueried output. A smaller budget would therefore contradict Proposition 3.4. This explains why the lower-bound constants are independent of d, even though the upper-bound stencil size is not. □

Combining Propositions 3.3 and 3.5 proves the two-sided bound in Theorem 3.1. Above κ_p, the additive constants are absorbed. With all positive budgets fixed, the $q = p$ term eventually minimizes R_ϵ, giving exponent $(p+1)/p$; when budgets vary with ϵ, the full expression in (1) is essential.

4 White-box computational hardness

We now distinguish query complexity from computation with a finite description. We use the Stationary model of Hollender & Zampetakis (2025), restricted to exact rational evaluators. An input supplies clocked polynomial-time machines returning a rational value and gradient at each rational query point. Their time and output bit length are polynomial in the description and query lengths; no real-number oracle is hidden in the representation. Agreement with a differentiable real function is a semantic promise, while clocks also enforce the running-time bound on invalid inputs.

We inherit the base problem's polynomial output-length bound, verification rule, and accepted certificates of violated boundedness, Lipschitz continuity, or value–gradient consistency. Stationarity is checked by comparing the squared norm of the exact rational gradient with ϵ^2. Rather than requiring a verifier to decide higher differentiability of an arbitrary description, we impose higher smoothness on the hard instances of this totalized problem. Every constructed hard instance has exact rational evaluators, and none admits a valid base violation certificate.

Definition 4.1 (The totalized problem $\mathsf{B\text{-}Stationary}_{d,p}$)**.** Fix integers $d, p \geq 1$. An input consists of value and gradient evaluator descriptions in the preceding base model, a rational accuracy $\epsilon > 0$, and positive rational declarations

$$M_0, M_1, \ldots, M_p, \Lambda_p.$$

Set

$$L_{\mathrm{grad}} := \begin{cases} \Lambda_p, & p = 1, \\ M_2, & p \geq 2, \end{cases} \qquad B := M_0, \qquad L := \max\{M_1, L_{\mathrm{grad}}\}. \tag{3}$$

The associated base instance is Stationary with these evaluator descriptions and parameters ϵ, B, L. A solution is exactly an accepted finite solution of this base instance. Syntactically malformed inputs are treated using the base encoding convention.

The *higher-smoothness promise* is that the evaluators correctly represent a function $f \in \mathcal{B}^{p,1}(\mathbb{R}^d)$ satisfying the declarations in Definition 2.2: its value and derivatives of orders $1, \ldots, p$ are bounded by $M_0, M_1, \ldots, M_p$, respectively, and its p-th derivative is Λ_p-Lipschitz. On this promise, the required output is a rational approximate stationary point x, with $\|\nabla f(x)\| \leq \epsilon$. The declarations of higher derivative bounds are semantic promises, not syntactically verifiable certificates. On inputs that fail this promise, the solution relation remains the base solution relation; it does not require a certificate witnessing every possible failure of higher smoothness.

Indeed, a valid higher-smoothness input is a valid base input. The derivative bound M_1 makes f globally M_1-Lipschitz. For $p \geq 2$, the bound M_2 makes its gradient M_2-Lipschitz by the fundamental theorem of calculus; for $p = 1$, the gradient Lipschitz bound is supplied directly. Thus the choice in (3) certifies the base analytic properties on the promise, and no base violation solution exists there.

Theorem 4.2 (PLS-completeness with higher-smoothness promises)**.** *For every fixed finite integer $p \geq 1$ and every fixed dimension $d \geq 2$, the totalized problem* $\mathsf{B\text{-}Stationary}_{d,p}$ *is* PLS*-complete. Its* PLS*-hardness holds even when the higher-smoothness promise in Definition 4.1 is satisfied. Hence, a polynomial-time algorithm that finds approximate stationary points on all valid promised inputs would yield a polynomial-time algorithm for every problem in* PLS*.*

Proof sketch of Theorem 4.2. Preserve the computational encoding while smoothing the geometry. Membership and hardness require different arguments. For membership, Proposition D.1 maps the totalized input to its associated Stationary instance without changing its solution relation. The inherited potential, and neighborhood rule establish PLS membership, including on invalid inputs. A continuous descent inequality alone would not do so.

For hardness, Lemma D.2 supplies a periodic, cellwise bicubic function F_I encoding an Iter instance I. Every rational point with $\|\nabla F_I\| \leq \epsilon_0$ decodes a source solution, where $\epsilon_0 = 1/100$ is fixed. Smooth F_I with the polynomial probability kernel from (19):

$$\eta_I = \min\{1/8, \epsilon_0/(8L_I)\}, \qquad G_I = F_I * \psi_{p,\eta_I}, \qquad \epsilon_I = \epsilon_0/4.$$

There are three properties to preserve. First, transfer derivatives onto the *known kernel*, rather than differentiate the merely C^1 source repeatedly. The proof of Lemma D.5 gives

$$D^j G_I = F_I * D^j \psi_{p,\eta_I}, \qquad \left\|D^j G_I\right\|_\infty \leq B_I A_{p,j} \eta_I^{-j} \quad (1 \leq j \leq p+1).$$

The kernel's boundary vanishing makes these derivatives continuous globally, not just inside each cell. The $(p+1)$-st derivative bound also controls $\mathrm{Lip}(D^p G_I)$; the rational common bound Q_I in (24) therefore certifies all declarations with polynomial encoding length. Second, Lemmas D.6–D.7 give the uniform implication

$$\|\nabla G_I(x)\| \leq \epsilon_I \quad \Longrightarrow \quad \|\nabla F_I(x)\| \leq \epsilon_I + \sqrt{2} L_I \eta_I \leq \epsilon_0/2 < \epsilon_0.$$

This estimate holds across cell boundaries and periodic seams, so smoothing creates no approximate stationary point that escapes the source decoder.

Third, smoothing must preserve *exact finite evaluation*, not just regularity. The support square meets at most four source cells with positive area. On each resulting rational rectangle, both the source function and the kernel are fixed-degree polynomials. Their product can therefore be integrated exactly by rational arithmetic; the same applies to the gradient. Lemma D.8 shows that the evaluator size and running time are polynomial for fixed p. This is why we use a polynomial kernel, rather than invoke an unspecified smooth mollifier.

Proposition D.9 combines these properties: all declarations are valid, so every accepted output is a stationary point and decodes a solution of I. Proposition D.10 extends the reduction to any fixed $d \geq 2$ by ignoring the extra coordinates. Numerical bounds such as Q_I may be exponentially large but have polynomial bit length; normalizing them must also rescale ϵ_I, as explained in Remark D.11. □

Remark 4.3 (Relation to known hardness results)*.* The cases $p = 1$ and $p = 2$ are already covered by Hollender & Zampetakis (2025) and Kontogiannis et al. (2024), respectively. The latter's bounded, gradient-Lipschitz, Hessian-Lipschitz functions also satisfy the bounded-derivative requirements here: if $|f| \leq B$ and ∇f is L-Lipschitz, Lemma C.1 and differentiability give $\|\nabla f(x)\| \leq 2\sqrt{BL}$ and $\left\|D^2 f(x)\right\|_{\mathrm{op}} \leq L$. Our contribution is an effective smoothing extension to every fixed finite p, not a new hardness claim for these two known cases.

5 Conclusion

For fixed d, p, deterministic exact gradient-only stationarity has complexity $\Theta_{d,p}(R_\epsilon)$ above a constant threshold. Finite-difference regularization and a one-dimensional smooth resisting oracle give matching bounds while tracking every smoothness budget and allowing arbitrary query locations and unqueried outputs. Separately, effective polynomial smoothing preserves PLS-completeness of the totalized finite-description problem on valid higher-smoothness inputs. These conclusions distinguish information from computation: the oracle model does not charge internal model minimization, and normalizing the hardness construction necessarily rescales its accuracy.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the grammar and reorganizing the paper's structure. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All assumptions and theoretical claims are stated in the paper, with complete proofs provided in the appendix.

REFERENCES

Deeksha Adil, Brian Bullins, Aaron Sidford, and Chenyi Zhang. Balancing gradient and Hessian queries in non-convex optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 43299–43346, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3dc910c921507a5d51d3dac0202b2618-Abstract-Conference.html.

Ernesto G. Birgin, John L. Gardenghi, José M. Martínez, Sandra A. Santos, and Philippe L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1–2):359–368, 2017. doi: 10.1007/s10107-016-1065-8.

Sébastien Bubeck and Dan Mikulincer. How to trap a gradient flow. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 940–960, 2020. URL https://proceedings.mlr.press/v125/bubeck20b.html.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. "Convex Until Proven Guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 654–663. PMLR, 2017. URL https://proceedings.mlr.press/v70/carmon17a.html.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184:71–120, 2020. URL https://arxiv.org/abs/1710.11606.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *Mathematical Programming*, 185(1–2):315–355, 2021. doi: 10.1007/s10107-019-01431-x.

Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM Journal on Optimization*, 22(1):66–86, 2012. doi: 10.1137/100812276.

Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM Journal on Optimization*, 30(1):513–541, 2020. doi: 10.1137/17M1144854.

Sinho Chewi, Sébastien Bubeck, and Adil Salim. On the complexity of finding stationary points of smooth functions in one dimension. In *Proceedings of the 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pp. 358–374. PMLR, 2023. URL https://proceedings.mlr.press/v201/chewi23a.html.

Nikita Doikov and Geovani Nunes Grapiglia. On the complexity of lower-order implementations of higher-order methods. *Mathematical Programming*, 2026. doi: 10.1007/s10107-026-02386-6. Published online July 8, 2026.

Geovani Nunes Grapiglia, Max L. N. Gonçalves, and G. N. Silva. A cubic regularization of Newton's method with finite difference Hessian approximations. *Numerical Algorithms*, 90 (2):607–630, 2022. doi: 10.1007/s11075-021-01200-y.

Serge Gratton, Sadok Jerad, and Philippe L. Toint. A stochastic objective-function-free adaptive regularization method with optimal complexity. *Open Journal of Mathematical Optimization*, 6:1–24, 2025. doi: 10.5802/ojmo.41. URL https://ojmo.centre-mersenne.org/articles/10.5802/ojmo.41/. Article 5.

Alexandros Hollender and Manolis Zampetakis. The computational complexity of finding stationary points in non-convex optimization. *Mathematical Programming*, 213:281–341, 2025. doi: 10.1007/s10107-024-02139-3. URL https://arxiv.org/abs/2310.09157v2. Appendix references in this manuscript refer to arXiv:2310.09157v2.

David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi: 10.1016/0022-0000(88)90046-3.

Andreas Kontogiannis, Vasilis Pollatos, Sotiris Kanellopoulos, Panayotis Mertikopoulos, Aris Pagourtzis, and Ioannis Panageas. The computational complexity of finding second-order stationary points. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25191–25225. PMLR, 2024. URL https://proceedings.mlr.press/v235/kontogiannis24a.html.

Huan Li and Zhouchen Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $O(\epsilon^{-7/4})$ complexity. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12901–12916. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22o.html.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004. doi: 10.1007/978-1-4419-8853-9.

Yurii Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006. doi: 10.1007/s10107-006-0706-8.

Stephen A. Vavasis. Black-box complexity of local minimization. *SIAM Journal on Optimization*, 3(1):60–80, 1993. doi: 10.1137/0803004.

Dongruo Zhou. Sharp first-order lower bounds for higher-order smooth nonconvex optimization, 2026. URL https://arxiv.org/abs/2606.05438v2. arXiv:2606.05438v2, September 21, 2026.

APPENDIX

A Scope of the literature comparison

The comparisons in Table 1 depend on the information returned, the function class, and the quantifiers over algorithms and dimension. We summarize these distinctions without repeating the table.

Information and domain. A value-and-gradient lower bound also applies to gradients only, but its upper bound need not. In one dimension this distinction changes the deterministic complexity from $\Theta(\log(1/\epsilon))$ to $\Theta(\epsilon^{-2})$ (Chewi et al., 2023). For bounded objectives on $\mathbb{R}^2$, Hollender & Zampetakis (2025) obtain matching ϵ^{-1} bounds with values and gradients. The Hessian-Lipschitz lower-bound entry is justified by the *first-order* argument in Kontogiannis et al. (2024, Appendix F), before its second-order corollary; it is not inferred from SOSP hardness. Compact-domain bounds concern a different, boundary-aware target: the planar lower bound of Vavasis (1993) and the upper bound of Hollender & Zampetakis (2025, Theorem 4.2) match at $\epsilon^{-1/2}$. See also Bubeck & Mikulincer (2020) for the lower construction. They do not yield an unconstrained gradient-norm bound.

Smoothness and parameter dependence. On $\mathcal{F}_p(\Delta, L_p)$, a degree-$p$ regularized Taylor model yields $O(1+\Delta L_p^{1/p}\epsilon^{-(p+1)/p})$ when the required derivatives are available (Birgin et al., 2017; Cartis et al., 2020). On $\mathcal{F}_{1:p}$, the objective-function-free inexact-derivative theory of Gratton et al. (2025, Theorem 13 and Section 4.1), combined with fixed-size gradient stencils, yields the normalized upper exponent in fixed dimension. The derivative bounds $\|D^r f\|_\infty \le L_{r-1}$, $2 \le r \le p$, supply its lower-model bounds. This normalized statement does not alone identify constants uniform in every L_q. Proposition 3.3 directly analyzes each order using only its top-order Lipschitz bound, then selects the most favorable one. In contrast, the lower construction must satisfy *all* budgets at once, which determines ρ_ϵ and R_ϵ.

Algorithmic quantifiers and dimension. The one-dimensional lower bound of Chewi et al. (2023) covers arbitrary deterministic gradient queries at $p = 1$. At higher order, the sharp examples in Cartis et al. (2020) concern specified regularization and model-based schemes. Proposition 3.4 instead permits any deterministic query rule and unqueried outputs. The unrestricted high-dimensional bounds of Carmon et al. (2020), the first-order bounds of Carmon et al. (2021), and the sharper exponents reported in Zhou (2026) allow dimension to grow with inverse accuracy. They therefore do not establish a fixed-dimensional, simultaneous-budget lower bound; the oracle and function-class restrictions must also match.

Finite-description hardness. The known $p = 1, 2$ cases are Hollender & Zampetakis (2025) and Kontogiannis et al. (2024). Bounded values and a Lipschitz gradient already imply bounded first and second derivatives in the latter setting, as shown in Remark 4.3; the notation $\mathcal{B}^{2,1}$ is not itself an additional hardness contribution. Theorem 4.2 extends this barrier to every fixed finite order while preserving exact rational evaluation and decoding. Its total-search convention is Definition 4.1, and normalization rescales accuracy as in Remark D.11. This computational result is separate from the exact-gradient query characterization.

B Finite-difference reconstruction

This appendix specifies the finite-difference reconstruction used by Algorithm 1. Fix $d \geq 1$ and $q \geq 2$. We fix the stencil width to be q, rather than introducing an additional free parameter. Define

$$\mathcal{K}_q := \{-q, -q+1, \ldots, q-1, q\}, \qquad S_{d,q} := \mathcal{K}_q^d.$$

Thus $|S_{d,q}| = (2q+1)^d$, and every $z \in S_{d,q}$ satisfies $\|z\| \leq q\sqrt{d}$.

For each $a \in \mathcal{K}_q$, let ℓ_a be the one-dimensional Lagrange basis polynomial

$$\ell_a(t) := \prod_{\substack{b \in \mathcal{K}_q \\ b \neq a}} \frac{t-b}{a-b}.$$

For $z = (z_1, \ldots, z_d) \in S_{d,q}$, define the tensor-product basis polynomial

$$L_z(u) := \prod_{j=1}^{d} \ell_{z_j}(u_j), \qquad u = (u_1, \ldots, u_d) \in \mathbb{R}^d.$$

Write $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$. For every multi-index $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$, write $|\alpha| := \sum_{j=1}^d \alpha_j$. The finite-difference weight is

$$\omega_{\alpha,z}^{(d,q)} := \partial^\alpha L_z(0) = \prod_{j=1}^{d} \ell_{z_j}^{(\alpha_j)}(0). \tag{4}$$

These weights depend only on d, q, α, z.

Let $\mathrm{Sym}^r(\mathbb{R}^d)$ denote the space of symmetric r-linear tensors. We next specify the symmetric-tensor reconstruction operator

$$\mathcal{W}_{r,z}^{(d,q)} : \mathbb{R}^d \longrightarrow \mathrm{Sym}^r(\mathbb{R}^d), \qquad 2 \leq r \leq q, \quad z \in S_{d,q}.$$

Write $[d] = \{1, \ldots, d\}$, and let $I = (i_1, \ldots, i_r) \in [d]^r$ be an ordered index tuple. For $\ell \in \{1, \ldots, r\}$, define

$$\alpha_{I,\ell} := \sum_{\substack{s=1 \\ s \neq \ell}}^{r} e_{i_s} \in \mathbb{N}_0^d,$$

where $e_1, \ldots, e_d$ are the standard basis vectors. The multi-index $\alpha_{I,\ell}$ records the multiplicities of all indices of I except its ℓ-th index. For $Y \in \mathbb{R}^d$, set

$$\big(\mathcal{W}_{r,z}^{(d,q)}[Y]\big)_{i_1,\ldots,i_r} := \frac{1}{r} \sum_{\ell=1}^{r} \omega_{\alpha_{I,\ell},z}^{(d,q)} Y_{i_\ell}. \tag{5}$$

Permuting the indices of I only permutes the summands, so this definition gives a fully symmetric tensor.

Given a point $x \in \mathbb{R}^d$ and a radius $h > 0$, the estimator is

$$G_r(x) := \frac{1}{h^{r-1}} \sum_{z \in S_{d,q}} \mathcal{W}_{r,z}^{(d,q)}\big[\nabla f(x + hz)\big], \qquad r = 2, \ldots, q. \tag{6}$$

Equivalently, for $I = (i_1, \ldots, i_r)$,

$$\big(G_r(x)\big)_{i_1,\ldots,i_r} = \frac{1}{r h^{r-1}} \sum_{\ell=1}^{r} \sum_{z \in S_{d,q}} \omega_{\alpha_{I,\ell},z}^{(d,q)} \partial_{i_\ell} f(x + hz).$$

In particular, the second-order reconstruction is

$$\big(\mathcal{W}_{2,z}^{(d,q)}[Y]\big)_{ij} = \frac{1}{2}\left(\omega_{e_j,z}^{(d,q)} Y_i + \omega_{e_i,z}^{(d,q)} Y_j\right),$$

and therefore

$$\big(G_2(x)\big)_{ij} = \frac{1}{2h} \sum_{z \in S_{d,q}} \Big(\omega^{(d,q)}_{e_j,z} \partial_i f(x + hz) + \omega^{(d,q)}_{e_i,z} \partial_j f(x + hz) \Big).$$

Algorithm 2 FD-Tensors: symmetric derivative tensors from gradient queries

Require: $x \in \mathbb{R}^d$, integer $q \geq 2$, radius $h > 0$, and an exact gradient oracle $\mathcal{O}_\nabla$.
Ensure: The tensors $G_2(x), \ldots, G_q(x)$ in (6).
1: **for** each $z \in S_{d,q}$ **do**
2: Query $Y_z \leftarrow \mathcal{O}_\nabla(x + hz)$.
3: **end for**
4: **for** $r = 2, \ldots, q$ **do**
5: Set $G_r(x) \leftarrow h^{-(r-1)} \sum_{z \in S_{d,q}} \mathcal{W}^{(d,q)}_{r,z}[Y_z]$, using (5).
6: **end for**
7: **return** $(G_2(x), \ldots, G_q(x))$.

All tensors use the same $(2q+1)^d$ gradient evaluations. The stencil contains $z = 0$, so a previously obtained gradient at x may be reused. Lemma C.3 proves the approximation guarantee using only Lipschitz continuity of $D^q f$.

C Proofs of the oracle complexity bounds

Proof of Theorem 3.1. Propositions 3.3 and 3.5 give (2) for all positive parameters, with lower-bound constants independent of d. The proof of Proposition 3.4 chooses a threshold $\kappa_p \geq 1$ at which the lower bound is a positive constant multiple of R_ϵ; the upper additive constant is absorbed because $1 + R_\epsilon \leq 2R_\epsilon$ there. The identity of the two parameter expressions follows from (1).

For completeness, if the positive smoothness budgets are fixed, then for each $q < p$,

$$\frac{L_q^{1/q} \epsilon^{-(q+1)/q}}{L_p^{1/p} \epsilon^{-(p+1)/p}} = \frac{L_q^{1/q}}{L_p^{1/p}} \epsilon^{-(1/q - 1/p)} \longrightarrow \infty.$$

There are finitely many such q, so the minimum is eventually attained at p (trivially so when $p = 1$). With $\Delta > 0$ fixed, $R_\epsilon \to \infty$, which proves the final asymptotic assertion in the nontrivial regime. □

C.1 Upper-bound proofs

We begin with the first-order descent inequality, which also justifies the bounded-gradient observation in Remark 4.3.

Lemma C.1 (Descent lemma)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be continuously differentiable with an L-Lipschitz gradient, where $L > 0$. Then, for all $x, y \in \mathbb{R}^d$,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2.$$

In particular, $x^+ = x - L^{-1} \nabla f(x)$ satisfies

$$f(x^+) \leq f(x) - \frac{1}{2L} \|\nabla f(x)\|^2.$$

Proof. The fundamental theorem of calculus along the segment joining x and y gives

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x \rangle \, dt \\ &\leq \int_0^1 Lt \|y - x\|^2 \, dt = \frac{L}{2} \|y - x\|^2. \end{aligned}$$

Substitution of $y = x - L^{-1} \nabla f(x)$ proves the second assertion. □

We next record the decrease and endpoint-gradient estimates for an exact regularized Taylor step.

Lemma C.2 (Exact high-order regularized step)**.** *Let $q \geq 1$, let $f \in C^q(\mathbb{R}^d)$, and suppose that $D^q f$ is L_q-Lipschitz. At $x \in \mathbb{R}^d$, define*

$$T_q(x; s) := f(x) + \sum_{r=1}^{q} \frac{1}{r!} D^r f(x)[s]^r.$$

Let $\sigma > L_q$, and let

$$s \in \arg\min_{u \in \mathbb{R}^d} \left\{ T_q(x; u) + \frac{\sigma}{(q+1)!} \|u\|^{q+1} \right\}.$$

Then

$$f(x) - f(x+s) \geq \frac{\sigma - L_q}{(q+1)!} \|s\|^{q+1}, \tag{7}$$

$$\|\nabla f(x+s)\| \leq \frac{\sigma + L_q}{q!} \|s\|^q. \tag{8}$$

In particular, for $\sigma = 2L_q$, if $\|\nabla f(x+s)\| > \epsilon$, then

$$f(x) - f(x+s) \geq c_q L_q^{-1/q} \epsilon^{(q+1)/q},$$

where $c_q > 0$ depends only on q.

Proof. The regularized model is continuous and coercive, since its positive term of degree $q+1$ dominates its Taylor polynomial of degree at most q. Its global minimum is therefore attained. Taylor's theorem with a Lipschitz highest derivative gives

$$f(x+s) \leq T_q(x; s) + \frac{L_q}{(q+1)!} \|s\|^{q+1}.$$

Global minimality and feasibility of $u = 0$ give

$$T_q(x; s) + \frac{\sigma}{(q+1)!} \|s\|^{q+1} \leq T_q(x; 0) = f(x).$$

Combining these inequalities proves (7).

The first-order optimality condition for the regularized model is

$$\nabla_s T_q(x; s) + \frac{\sigma}{q!} \|s\|^{q-1} s = 0.$$

The gradient Taylor remainder satisfies

$$\|\nabla f(x+s) - \nabla_s T_q(x; s)\| \leq \frac{L_q}{q!} \|s\|^q.$$

The triangle inequality proves (8). When $\sigma = 2L_q$ and $\|\nabla f(x+s)\| > \epsilon$, it follows that

$$\|s\| > \left(\frac{q!\epsilon}{3L_q} \right)^{1/q}.$$

Substitution in (7) gives the claimed decrease. □

We next prove the approximation guarantee for the reconstruction operators in Appendix B.

Lemma C.3. *Fix integers $d \geq 1$ and $q \geq 2$. Let $f \in C^q(\mathbb{R}^d)$ have an L_q-Lipschitz q-th derivative. For every $x \in \mathbb{R}^d$ and $h > 0$, Algorithm 2 constructs symmetric tensors $G_r(x)$, $r = 2, \ldots, q$, satisfying*

$$\|G_r(x) - D^r f(x)\|_{\mathrm{op}} \leq C_{d,q}^{\mathrm{FD}} L_q h^{q-r+1}. \tag{9}$$

It uses at most $N_{d,q} := (2q+1)^d$ gradient queries in the closed ball $\overline{B}(x, q\sqrt{d}\, h)$. The constants $N_{d,q}$ and $C_{d,q}^{\mathrm{FD}}$ depend only on d, q.

Proof. We first establish a scalar estimate. Let $\phi \in C^{q-1,1}(\mathbb{R}^d)$, and let $\alpha \in \mathbb{N}_0^d$ satisfy $|\alpha| = s \le q-1$. Define

$$\mathrm{FD}_{\alpha,h}\phi(x) := \frac{1}{h^s} \sum_{z \in S_{d,q}} \omega_{\alpha,z}^{(d,q)} \phi(x + hz).$$

We claim that

$$|\mathrm{FD}_{\alpha,h}\phi(x) - \partial^\alpha \phi(x)| \le C_{d,q}\, \mathrm{Lip}(D^{q-1}\phi)\, h^{q-s}. \tag{10}$$

Here and below constants denoted by $C_{d,q}$ may be enlarged, but depend only on d, q.

Consider the rescaled function $\Phi(u) := \phi(x + hu)$, and let P_{q-1} be its Taylor polynomial at 0 of total degree $q-1$. Every monomial of total degree at most $q-1$ has degree at most $2q$ in each coordinate. It therefore belongs to the tensor-product interpolation space on $S_{d,q}$. Consequently,

$$P_{q-1}(u) = \sum_{z \in S_{d,q}} P_{q-1}(z) L_z(u),$$

and differentiation at 0 gives

$$\partial^\alpha P_{q-1}(0) = \sum_{z \in S_{d,q}} \omega_{\alpha,z}^{(d,q)} P_{q-1}(z).$$

Since $s \le q-1$,

$$\partial^\alpha P_{q-1}(0) = \partial^\alpha \Phi(0) = h^s \partial^\alpha \phi(x).$$

The Lipschitz Taylor remainder and the bound $\|z\| \le q\sqrt{d}$ imply

$$|\Phi(z) - P_{q-1}(z)| \le C_{d,q}\, \mathrm{Lip}(D^{q-1}\phi)\, h^q \qquad (z \in S_{d,q}).$$

Thus

$$\begin{aligned}
\left| \sum_{z \in S_{d,q}} \omega_{\alpha,z}^{(d,q)} \Phi(z) - \partial^\alpha \Phi(0) \right| &= \left| \sum_{z \in S_{d,q}} \omega_{\alpha,z}^{(d,q)} \big(\Phi(z) - P_{q-1}(z)\big) \right| \\
&\le \sum_{z \in S_{d,q}} \left| \omega_{\alpha,z}^{(d,q)} \right| |\Phi(z) - P_{q-1}(z)| \\
&\le C_{d,q}\, \mathrm{Lip}(D^{q-1}\phi)\, h^q.
\end{aligned}$$

There are only finitely many weights and multi-indices under consideration, all determined by d, q. Dividing by h^s proves (10).

Apply this estimate to the coordinate functions $\phi_i := \partial_i f$, $i \in [d]$. Their derivatives of order $q-1$ are contractions of $D^q f$ with a coordinate unit vector. Therefore

$$\mathrm{Lip}(D^{q-1}\phi_i) \le L_q.$$

For every α with $|\alpha| = r-1$, this gives

$$\left| \frac{1}{h^{r-1}} \sum_{z \in S_{d,q}} \omega_{\alpha,z}^{(d,q)} \partial_i f(x + hz) - \partial^\alpha \partial_i f(x) \right| \le C_{d,q} L_q h^{q-r+1}.$$

Fix $r \in \{2, \dots, q\}$ and an ordered tuple $I = (i_1, \dots, i_r) \in [d]^r$. For each $\ell \in \{1, \dots, r\}$, $|\alpha_{I,\ell}| = r-1$, and equality of mixed partials yields

$$\partial^{\alpha_{I,\ell}} \partial_{i_\ell} f(x) = \partial_{i_1} \cdots \partial_{i_r} f(x) = \big(D^r f(x)\big)_{i_1,\dots,i_r}.$$

By (5) and (6),

$$\big(G_r(x)\big)_{i_1,\dots,i_r} = \frac{1}{r} \sum_{\ell=1}^{r} \left[\frac{1}{h^{r-1}} \sum_{z \in S_{d,q}} \omega_{\alpha_{I,\ell},z}^{(d,q)} \partial_{i_\ell} f(x + hz) \right].$$

Applying the scalar estimate to each bracket and averaging gives

$$\left|\left(G_r(x) - D^r f(x)\right)_{i_1,\dots,i_r}\right| \le C_{d,q} L_q h^{q-r+1}.$$

To pass to the operator norm, an r-tensor whose entries have absolute value at most E satisfies

$$|A[u_1,\dots,u_r]| \le E \prod_{j=1}^{r} \|u_j\|_1 \le E\, d^{r/2} \prod_{j=1}^{r} \|u_j\|.$$

Taking unit vectors and using $r \le q$ proves (9), after absorbing the dimension factor in $C^{\mathrm{FD}}_{d,q}$.

The query count and query radius follow directly from the definition of $S_{d,q}$. For the first derivative, no finite-difference reconstruction is needed: $G_1(x) = \nabla f(x)$ is queried exactly. □

We now combine derivative reconstruction and regularized descent. All constants in the next argument are chosen in a fixed order: the regularization parameter is $2L_q$, then a small-step threshold is fixed, then the permitted tensor error, and finally the finite-difference radius.

Lemma C.4. *Fix $q \ge 1$. Suppose $f \in C^q(\mathbb{R}^d)$ and $D^q f$ is L_q-Lipschitz. There is a deterministic procedure that, from a point x, uses $O_{d,q}(1)$ exact gradient queries and returns either a point y with $\|\nabla f(y)\| \le \epsilon$, or a point x^+ satisfying*

$$f(x) - f(x^+) \ge c_{d,q} L_q^{-1/q} \epsilon^{(q+1)/q},$$

where $c_{d,q} > 0$ depends only on d, q. For $q \ge 2$, the procedure uses the reconstruction in Algorithm 2 with radius $h = \nu_{d,q}(\epsilon/L_q)^{1/q}$, where $\nu_{d,q} > 0$ depends only on d, q, and model regularization parameter $\sigma = 2L_q$.

Proof. For $q = 1$, query $g = \nabla f(x)$. If $\|g\| \le \epsilon$, return x. Otherwise let $x^+ = x - L_1^{-1} g$. Lemma C.1 gives

$$f(x) - f(x^+) \ge \frac{1}{2L_1}\|g\|^2 > \frac{\epsilon^2}{2L_1}.$$

This proves the statement in this case.

Now suppose $q \ge 2$, and put

$$\tau_q := \left(\frac{\epsilon}{L_q}\right)^{1/q}.$$

Query $g = \nabla f(x)$ and return x if $\|g\| \le \epsilon$. Fix $c_0 \in (0, 1]$, depending only on q, such that

$$\frac{3c_0^q}{q!} \le \frac{1}{2}.$$

Next fix $\beta_{d,q} > 0$, depending only on d, q, so small that

$$\beta_{d,q} \sum_{r=2}^{q} \frac{c_0^{r-1}}{(r-1)!} \le \frac{1}{2}, \tag{11}$$

$$\beta_{d,q} \sum_{r=2}^{q} \frac{c_0^{-(q-r+1)}}{r!} \le \frac{1}{2(q+1)!}. \tag{12}$$

Finally choose $\nu_{d,q} \in (0, 1]$ so that

$$C^{\mathrm{FD}}_{d,q} \nu_{d,q}^{q-r+1} \le \beta_{d,q} \qquad (r = 2,\dots,q).$$

With $h = \nu_{d,q}\tau_q$, Lemma C.3 gives

$$\|G_r - D^r f(x)\|_{\mathrm{op}} \le \delta_r, \qquad \delta_r := \beta_{d,q} L_q \tau_q^{q-r+1}, \qquad r = 2,\dots,q. \tag{13}$$

Form the approximate model

$$\widetilde{m}_x(u) := \langle g, u\rangle + \sum_{r=2}^{q} \frac{1}{r!} G_r[u]^r + \frac{2L_q}{(q+1)!}\|u\|^{q+1},$$

and choose a global minimizer s. It exists by coercivity. Set $a := \|s\|$. Symmetry of the tensors gives the first-order optimality condition

$$g + \sum_{r=2}^{q} \frac{1}{(r-1)!} G_r[s]^{r-1} + \frac{2L_q}{q!} \|s\|^{q-1} s = 0. \tag{14}$$

Here $G_r[s]^{r-1}$ is the vector representing contraction with $r-1$ copies of s. Taylor's theorem for the gradient, (13), and (14) imply

$$\|\nabla f(x+s)\| \le \frac{3L_q}{q!} a^q + \sum_{r=2}^{q} \frac{\delta_r}{(r-1)!} a^{r-1}. \tag{15}$$

If $a \le c_0 \tau_q$, then $L_q \tau_q^q = \epsilon$ gives

$$\|\nabla f(x+s)\| \le \left(\frac{3c_0^q}{q!} + \beta_{d,q} \sum_{r=2}^{q} \frac{c_0^{r-1}}{(r-1)!} \right) \epsilon \le \epsilon.$$

Consequently,

$$\|\nabla f(x+s)\| > \epsilon \quad \Longrightarrow \quad a > c_0 \tau_q. \tag{16}$$

The procedure now queries $\nabla f(x+s)$. If its norm is at most ϵ, it returns $y = x + s$. It remains to prove descent when this test fails. Global minimality and $\widetilde{m}_x(0) = 0$ give

$$\langle g, s \rangle + \sum_{r=2}^{q} \frac{1}{r!} G_r[s]^r + \frac{2L_q}{(q+1)!} a^{q+1} \le 0.$$

On the other hand, Taylor's theorem gives

$$f(x+s) \le f(x) + \langle g, s \rangle + \sum_{r=2}^{q} \frac{1}{r!} D^r f(x)[s]^r + \frac{L_q}{(q+1)!} a^{q+1}.$$

Subtracting the model inequality and using (13) yields

$$f(x) - f(x+s) \ge \frac{L_q}{(q+1)!} a^{q+1} - \sum_{r=2}^{q} \frac{\delta_r}{r!} a^r.$$

By (16), each error term satisfies

$$\delta_r a^r = \beta_{d,q} L_q \tau_q^{q-r+1} a^r \le \beta_{d,q} c_0^{-(q-r+1)} L_q a^{q+1}.$$

Thus (12) ensures

$$f(x) - f(x+s) \ge \frac{L_q}{2(q+1)!} a^{q+1} \ge \frac{c_0^{q+1}}{2(q+1)!} L_q \tau_q^{q+1} = c_{d,q} L_q^{-1/q} \epsilon^{(q+1)/q}.$$

Returning $x^+ = x + s$ proves the asserted alternative.

There is one gradient query at x, at most $N_{d,q}$ stencil queries, and one endpoint query. Their total is at most $N_{d,q} + 2 = O_{d,q}(1)$; coincident queries may be reused. □

Proof of Proposition 3.3. Fix $q \in \{1, \dots, p\}$. Every call to the procedure of Lemma C.4 either terminates at an ϵ-stationary point or produces a successor whose objective value is smaller by at least

$$c_{d,q} L_q^{-1/q} \epsilon^{(q+1)/q}.$$

If N calls return such descending successors, then telescoping along those successors and using $f(x_0) - \inf_x f(x) \le \Delta$ gives

$$N c_{d,q} L_q^{-1/q} \epsilon^{(q+1)/q} \le f(x_0) - f(x_N) \le \Delta.$$

There are therefore at most

$$1 + C_{d,q} \Delta L_q^{1/q} \epsilon^{-(q+1)/q}$$

calls, including the final terminating call. No descent is needed for a terminal step: its endpoint is already a valid output. Multiplying by the $O_{d,q}(1)$ queries per call yields

$$C'_{d,q}\left(1+\Delta L_q^{1/q}\epsilon^{-(q+1)/q}\right)$$

gradient queries.

Algorithm 1 implements these steps and selects $q=q_\star$ minimizing $L_q^{1/q}\epsilon^{-(q+1)/q}$. Taking the largest of the finitely many constants $C'_{d,q}$, $1\le q\le p$, proves the stated bound with a constant depending only on d,p. □

C.2 Lower-bound proofs

The two ingredients can be summarized as follows.

1. *A finite-budget resisting oracle.* Fix a query budget N, and answer each gradient query by 2ϵ. Determinism fixes the query points on this transcript. If the algorithm outputs a point within the budget, its output is also fixed and is included in the set to be covered. If the algorithm instead requests a further query, only the first N answered queries need to be covered. This finite-budget formulation does not assume that an algorithm terminates against an indefinitely repeated constant reply.
2. *A smooth plateau covering.* For the resulting finite set S, construct a smooth function ϕ taking values in $[0,1]$, equal to one at every point of S, and supported on a set of measure $O_p((|S|+1)\rho)$. Its derivatives obey $\left\|\phi^{(q)}\right\|_\infty = O_p(\rho^{-q})$ for $1\le q\le p$. Consequently, $g=2\epsilon\phi$ can be the derivative of a legitimate objective with small function-value gap. Taking ρ to be a sufficiently large constant multiple of ρ_ϵ from (1) ensures all the required smoothness bounds.

The support may have a large diameter, since the algorithm may query arbitrarily distant points. Only its total measure enters the function-value-gap bound. This distinction permits arbitrary query locations and is essential to the construction.

Lemma C.5. *Fix an integer $p\ge 1$. There exists a constant $K_p\ge 1$, depending only on p, such that, for every finite set $S\subset\mathbb{R}$ and every $\rho>0$, there is a compactly supported C^∞ function $\phi:\mathbb{R}\to[0,1]$ satisfying*

$$\phi(s)=1 \quad (s\in S), \qquad \mathrm{Leb}(\operatorname{supp}\phi)\le K_p(|S|+1)\rho,$$

and

$$\left\|\phi^{(q)}\right\|_\infty \le K_p\rho^{-q}, \qquad q=1,\dots,p.$$

Here Leb *denotes Lebesgue measure and* $\operatorname{supp}\phi=\overline{\{x\in\mathbb{R}:\phi(x)\neq 0\}}$.

The proof below uses convolution of an indicator function with a fixed smooth kernel. In particular, the derivative bounds do not grow with the number of overlapping neighborhoods of points in S.

Proof of Lemma C.5. Choose once and for all a nonnegative function $\eta\in C_c^\infty((-1,1))$ with $\int_{\mathbb{R}}\eta(t)\,dt=1$. For $\rho>0$, put

$$\eta_\rho(t)=\rho^{-1}\eta(t/\rho), \qquad E=\bigcup_{s\in S}[s-\rho,s+\rho], \qquad \phi=\mathbf{1}_E*\eta_\rho.$$

If $S=\varnothing$, this definition gives $\phi=0$, and all the required properties hold. Otherwise E is compact and has finite measure, so $\phi\in C_c^\infty(\mathbb{R})$. Nonnegativity and unit mass of the kernel give $0\le\phi\le 1$. For any $s\in S$, the interval $[s-\rho,s+\rho]$ lies in E, and it contains the support of the function $t\mapsto\eta_\rho(s-t)$. Thus

$$\phi(s)=\int_E\eta_\rho(s-t)\,dt=\int_{\mathbb{R}}\eta_\rho(s-t)\,dt=1.$$

The support inclusion for convolution yields

$$\operatorname{supp}\phi \subseteq E + [-\rho, \rho] = \bigcup_{s\in S} [s - 2\rho, s + 2\rho].$$

Consequently,

$$\operatorname{Leb}(\operatorname{supp}\phi) \le 4\,|S|\,\rho.$$

This bound is valid whether or not the intervals overlap. Differentiating the smooth kernel, and using the L^∞-L^1 convolution inequality, gives, for every $q \ge 1$,

$$\left\|\phi^{(q)}\right\|_\infty = \left\|\mathbf{1}_E * \eta_\rho^{(q)}\right\|_\infty \le \|\mathbf{1}_E\|_\infty \left\|\eta_\rho^{(q)}\right\|_{L^1} \le \rho^{-q} \left\|\eta^{(q)}\right\|_{L^1}.$$

Taking

$$K_p = \max\left\{4, \max_{1\le q\le p} \left\|\eta^{(q)}\right\|_{L^1}\right\}$$

proves the lemma. In particular, K_p is independent of S, its diameter, and the number of overlaps among the intervals defining E. □

To see the role of the gap constraint, apply Lemma C.5 to the finite transcript set at scale $\rho_\epsilon^* = C_p^* \rho_\epsilon$, where C_p^* depends only on p, and define

$$g(x) = 2\epsilon\phi(x), \qquad f(x) = \int_{-\infty}^{x} g(t)\,dt.$$

Then $f'(s) = 2\epsilon$ at every covered point, whereas $f(0) - \inf f$ is at most $O_p((N+2)\epsilon\rho_\epsilon)$. For a sufficiently small budget relative to $\Delta/(\epsilon\rho_\epsilon)$, this is a valid objective in the prescribed class. It either makes the algorithm request more than N queries or makes its output fail the stationarity test. The following proof verifies both transcript cases and all simultaneous smoothness estimates.

Proof of Proposition 3.4. Let K_p be the constant from Lemma C.5. Define

$$C_p^* = \max\left\{1, \max_{1\le q\le p} (2K_p)^{1/q}\right\}, \qquad \rho_\epsilon^* = C_p^*\rho_\epsilon, \qquad B_p = 2K_pC_p^*.$$

All these constants except the scale ρ_ϵ^* depend only on p. Fix an integer $N \ge 0$ for which

$$B_p(N+2)\epsilon\rho_\epsilon \le \Delta. \tag{17}$$

We show that no deterministic gradient-only algorithm can guarantee success on the whole class using at most N queries.

Let A be such an algorithm, and run it against the formal replies

$$\mathcal{O}(x) = 2\epsilon$$

at every query. Inspect this execution until the algorithm either outputs a point or requests its $(N+1)$-st oracle answer; in the latter case, do not provide that additional answer. As in the usual oracle model, the next query or final output is determined by the finite transcript, and internal computation is not charged. The two cases give a finite set S as follows:

1. If A outputs $\widehat{x}$ after $t \le N$ queries, set

   $$S = \{x_1, \dots, x_t, \widehat{x}\}.$$

   The output is included even if it was never queried.

2. If A requests a further query after receiving N answers, set

   $$S = \{x_1, \dots, x_N\}.$$

In either case, $|S| \le N+1$. No claim is made that the infinite constant transcript is itself generated by a member of the function class. Only this finite prefix will be realized. If an algorithm were allowed to fail to complete its internal computation after a shorter finite prefix, the same construction for that prefix would already contradict its assumed success on every valid objective.

Apply Lemma C.5 with scale ρ_ϵ^*, and define

$$g(x) = 2\epsilon\phi(x), \qquad f(x) = \int_{-\infty}^{x} g(t)\,dt.$$

The compact support of ϕ makes this integral finite for every $x \in \mathbb{R}$. Moreover, $f \in C^\infty(\mathbb{R})$, $f' = g$, and

$$f'(s) = 2\epsilon \qquad (s \in S). \tag{18}$$

We first verify that f belongs to the prescribed class.

For every $q \in \{1, \dots, p\}$, the definition of ρ_ϵ gives $\rho_\epsilon \ge (\epsilon/L_q)^{1/q}$, and hence

$$\begin{aligned}\left\|g^{(q)}\right\|_\infty &\le 2\epsilon K_p(\rho_\epsilon^*)^{-q} \\ &= 2K_p(C_p^*)^{-q}\epsilon\rho_\epsilon^{-q} \le \epsilon\rho_\epsilon^{-q} \le L_q.\end{aligned}$$

Since $f^{(q+1)} = g^{(q)}$, the fundamental theorem of calculus now implies

$$\left|f^{(q)}(x) - f^{(q)}(y)\right| \le \left\|g^{(q)}\right\|_\infty |x-y| \le L_q\,|x-y| \qquad (x, y \in \mathbb{R}).$$

Thus all the derivative-Lipschitz constraints in Definition 2.1 hold simultaneously.

To verify the gap condition, observe that $g \ge 0$ and g has compact support. Therefore $f \ge 0$, and f vanishes sufficiently far to the left, so $\inf_{x\in\mathbb{R}} f(x) = 0$. The support estimate and $\|g\|_\infty \le 2\epsilon$ yield

$$\begin{aligned}f(0) - \inf_{x\in\mathbb{R}} f(x) &\le \int_{\mathbb{R}} g(t)\,dt \\ &\le 2\epsilon K_p(|S|+1)\rho_\epsilon^* \\ &\le 2K_pC_p^*(N+2)\epsilon\rho_\epsilon = B_p(N+2)\epsilon\rho_\epsilon \le \Delta.\end{aligned}$$

Here the scale ratio absorbed in the constant is $\rho_\epsilon^*/\rho_\epsilon = C_p^*$, which is independent of $\epsilon, \Delta, L_1, \dots, L_p$. We have proved that $f \in \mathcal{F}_{1:p}(\Delta, L_1, \dots, L_p)$.

By (18) and determinism, the real gradient oracle for this f reproduces every answer in the recorded prefix. In the first case, A therefore outputs the same $\widehat{x}$, but

$$|f'(\widehat{x})| = 2\epsilon > \epsilon,$$

so it fails. In the second case, A requests its $(N+1)$-st query also on this valid objective, and so exceeds the budget. Thus every budget satisfying (17) is insufficient for guaranteed success.

Recall $R_\epsilon = \Delta/(\epsilon\rho_\epsilon)$ from (1). If $R_\epsilon \ge 2B_p$, choose $N = \lfloor R_\epsilon/B_p - 2 \rfloor$. Then the worst-case query complexity is strictly greater than N, and in particular it is at least $R_\epsilon/B_p - 2$. If $R_\epsilon < 2B_p$, that last expression is negative, and the same lower bound follows from nonnegativity of query complexity. Hence the proposition holds, for example, with $c_p = B_p^{-1}$ and $C_p = 2$. For $R_\epsilon \ge \kappa_p := 4B_p$, it further gives the positive lower bound $R_\epsilon/(2B_p)$. Finally,

$$\frac{1}{\epsilon\rho_\epsilon} = \min_{1\le q\le p} L_q^{1/q}\epsilon^{-(q+1)/q},$$

which proves the stated parameter-dependent form. □

Proof of Proposition 3.5. Fix $d \ge 1$. For any scalar objective $\varphi \in \mathcal{F}_{1:p}(\Delta, L_1, \dots, L_p)$ on $\mathbb{R}$, define

$$f(x_1, \dots, x_d) = \varphi(x_1).$$

Then

$$\nabla f(x) = (\varphi'(x_1), 0, \dots, 0), \qquad \|\nabla f(x)\| = |\varphi'(x_1)|,$$

and

$$f(0) - \inf_{x \in \mathbb{R}^d} f(x) = \varphi(0) - \inf_{t \in \mathbb{R}} \varphi(t) \le \Delta.$$

For $1 \le q \le p$, the derivative tensor is $D^q f(x) = \varphi^{(q)}(x_1) e_1^{\otimes q}$. The rank-one tensor $e_1^{\otimes q}$ has operator norm one, so, for any $x, y \in \mathbb{R}^d$,

$$\begin{aligned} \|D^q f(x) - D^q f(y)\|_{\mathrm{op}} &= \left|\varphi^{(q)}(x_1) - \varphi^{(q)}(y_1)\right| \\ &\le L_q |x_1 - y_1| \le L_q \|x - y\|. \end{aligned}$$

Thus the embedded function belongs to the same simultaneous smoothness class, with the same parameters.

Given a deterministic gradient-only algorithm in $\mathbb{R}^d$, simulate it on such embedded objectives. Its query at $x \in \mathbb{R}^d$ is answered by making one scalar derivative query at x_1, then returning $(\varphi'(x_1), 0, \dots, 0)$. If the algorithm outputs $\widehat{x} \in \mathbb{R}^d$, return its first coordinate as the scalar output. The equality of gradient norms above shows that a successful output in $\mathbb{R}^d$ is a successful scalar output, and the simulation uses exactly one scalar query per gradient query. A better worst-case budget would therefore contradict Proposition 3.4. The lower-bound constants are unchanged, and in particular are independent of d. □

D Proofs of the computational hardness result

This appendix contains the technical statements underlying Theorem 4.2. We first establish membership on all encoded inputs, then construct valid higher-smoothness hard inputs. The latter part separates the source representation, analytic smoothing, exact evaluation, and search reductions, so both the analytic and finite-encoding requirements can be checked in order.

D.1 Membership in the finite local-search model

Membership is inherited from the finite local-search construction for Stationary. A descent inequality by itself would not supply the finite search space, potential, and accepted solution relation.

Proposition D.1 (Membership)**.** *For every fixed $p \ge 1$ and every fixed $d \ge 1$,* $\mathsf{B\text{-}Stationary}_{d,p} \in \mathsf{PLS}$.

Proof of Proposition D.1. Map an input of $\mathsf{B\text{-}Stationary}_{d,p}$ to its associated base Stationary instance using (3), and retain the evaluator descriptions unchanged. Taking a maximum of a fixed number of rational parameters is a polynomial-time operation. By definition, every accepted finite solution of the base instance is an accepted solution of the totalized input, so the solution map is the identity. The membership theorem for Stationary in (Hollender & Zampetakis, 2025, Section 3.1 and Appendix B) therefore proves the proposition on all encoded inputs, including inputs that do not satisfy the higher-smoothness promise.

For completeness, the analytic validity of this map on promised inputs follows directly from the declared bounds. The derivative bound M_1 implies

$$|f(x) - f(y)| \le M_1 \|x - y\|.$$

For $p = 1$, ∇f is Λ_p-Lipschitz by assumption. For $p \ge 2$, integration of the Hessian along the segment from x to y gives

$$\|\nabla f(x) - \nabla f(y)\| \le \int_0^1 \left\|D^2 f(y + t(x - y))\right\|_{\mathrm{op}} \|x - y\| \, dt \le M_2 \|x - y\|.$$

Also $|f(x)| \le M_0$. Thus the associated base instance satisfies its analytic promises, and the gradient evaluator is consistent with the value evaluator. No base violation certificate is valid on this promised subclass.

The finite local-search construction in the cited membership theorem is necessary here. Lemma C.1 explains why a nonstationary point has a descent step, but an iteration on the continuous, unbounded domain does not by itself define a PLS problem. The base construction supplies a finite grid of polynomial-bit points, a computable finite potential, and a neighborhood rule. It distinguishes points that can satisfy the descent value–distance bound from invalid grid points, thereby avoiding artificial local optima at a truncation boundary. Exact rational evaluation supplies the finite comparisons used in this construction. The base verification and violation conventions are inherited in Definition 4.1; no decision procedure for higher differentiability is used in the reduction. □

D.2 The source family and its decoding threshold

For hardness, let I be an instance of the canonical PLS-complete problem Iter. We record the precise interface to the source construction before choosing a smoothing radius.

Lemma D.2. *The construction underlying (Hollender & Zampetakis, 2025, Theorem 3.1) gives a fixed rational constant $\epsilon_0 > 0$ and a polynomial-time reduction from Iter to a two-dimensional objective $F_I : \mathbb{R}^2 \to \mathbb{R}$ with the following properties. For an instance I of length n:*

1. *$F_I \in \mathcal{B}^{1,1}(\mathbb{R}^2)$. Explicit positive rational bounds B_I, K_I, L_I, computable in polynomial time and of polynomial bit length, satisfy*
$$|F_I(x)| \le B_I, \qquad \|\nabla F_I(x)\| \le K_I, \qquad \|\nabla F_I(x) - \nabla F_I(y)\| \le L_I \|x - y\|.$$
In particular, $B_I, K_I, L_I \le 2^{\mathrm{poly}(n)}$.
2. *There is a polynomial-time decoder* Dec *such that any rational point x with $\|\nabla F_I(x)\| \le \epsilon_0$ yields a solution $\mathrm{Dec}(I, x)$ of I.*
3. *F_I is periodic, with an integer period of polynomial bit length. On each unit cell $[a, a+1] \times [b, b+1]$, $a, b \in \mathbb{Z}$, it is a polynomial of degree at most three in each coordinate. The rational coefficients, expressed in local cell coordinates, are obtainable from a constant number of evaluations of the circuit describing I, in polynomial time. Value and gradient evaluation at rational points is therefore exact and of polynomial bit complexity.*

These structural assertions follow from (Hollender & Zampetakis, 2025, Appendices C.2–C.3 and C.6–C.7), together with the decoding argument in that construction.

Verification of the source-family interface. We use the explicit construction in (Hollender & Zampetakis, 2025, Appendix C), not an abstract oracle representation of its hard function. Appendix C.1 makes the objective periodic with an integer period P_I of polynomial bit length. Appendix C.2 specifies values and first derivatives on the integer grid, and Appendix C.3 uses bicubic interpolation on each unit cell. Its displayed coefficient formula is a fixed rational linear transformation of the four corner values and gradients, with the mixed corner derivatives set to zero. This proves the degree and rational-coefficient assertions. Appendix C.7 shows that the local data are obtained from a constant number of queries to the source circuit; floor operations and reduction modulo P_I have polynomial bit cost even for large rational query coordinates.

Lemma 11 in Appendix C.6 of that work gives C^1 regularity, $|F_I| \le 2^{14} P_I$, and a common Lipschitz bound $2^{18} P_I$ for F_I and its gradient. Since a differentiable K-Lipschitz scalar function has gradient norm at most K, we may take
$$B_I = 2^{14} P_I, \qquad K_I = L_I = 2^{18} P_I.$$
These bounds have polynomial bit length. Finally, the stationary-point classification and decoding argument in Appendices C.4–C.7 yields a source solution at every rational point with gradient norm at most 0.01. All decoding operations use the same local rules and polynomial-bit arithmetic. This establishes each assertion of the interface, with $\epsilon_0 = 1/100$. □

Remark D.3 (The fixed decoding threshold). The source construction permits $\epsilon_0 = 1/100$. Its exact numerical value is unimportant. What matters is that it is a fixed positive rational number, independent of the instance length.

D.3 A polynomial kernel and preservation of the gradient

We use a compactly supported kernel that is polynomial on its support. Its finite degree is important for exact evaluation later; its boundary vanishing supplies the desired global differentiability.

Fix $p \geq 1$, and let

$$\theta_p(t) := c_p^{\mathrm{ker}}(1-t^2)^{p+2}\mathbf{1}_{[-1,1]}(t), \qquad c_p^{\mathrm{ker}} := \left(\int_{-1}^{1}(1-t^2)^{p+2}\,dt\right)^{-1}, \tag{19}$$
$$\psi_p(u_1,u_2) := \theta_p(u_1)\theta_p(u_2), \qquad \psi_{p,\eta}(u) := \eta^{-2}\psi_p(u/\eta) \quad (\eta>0).$$

The normalizing constant c_p^{ker} is a positive rational number: its reciprocal is the integral of a rational-coefficient polynomial over an interval with rational endpoints. The kernel is nonnegative, has total integral one, and satisfies

$$\mathrm{supp}(\psi_{p,\eta}) \subseteq [-\eta,\eta]^2.$$

Lemma D.4. *The kernel ψ_p is compactly supported and belongs to $C^{p+1}(\mathbb{R}^2)$. For each fixed p, one can choose explicit positive rational constants $A_{p,j}$, $0 \leq j \leq p+1$, such that*

$$\int_{\mathbb{R}^2} \left\|D^j\psi_{p,\eta}(u)\right\|_{\mathrm{op}}\,du \leq A_{p,j}\eta^{-j}.$$

For $j=0$, the integrand means $|\psi_{p,\eta}(u)|$, and $A_{p,0}=1$ is admissible.

Proof. The polynomial $(1-t^2)^{p+2}$ vanishes to order $p+2$ at each endpoint of $[-1,1]$. Consequently, its derivatives of orders zero through $p+1$ agree there with the corresponding derivatives of the zero function. Extension by zero is therefore C^{p+1}. Taking the product in (19) proves the same regularity for ψ_p, including at corners of its support square.

Every component of $D^j\psi_p$, $j \leq p+1$, is a rational-coefficient polynomial inside the square and vanishes outside it. Its integral norm is finite. An explicit rational upper bound can be obtained without integrating an operator norm exactly: bound that norm by the sum of the absolute values of all tensor components, expand the component polynomials, and bound each monomial in absolute value by one on $[-1,1]^2$. Four times the resulting sum of absolute coefficients is a rational upper bound for the integral. There are finitely many components and coefficients, with their number depending only on p. This provides the constants $A_{p,j}$; for $j=0$, nonnegativity and normalization give the sharper value one.

Differentiation of the scaled kernel gives

$$D^j\psi_{p,\eta}(u) = \eta^{-2-j}D^j\psi_p(u/\eta).$$

After the change of variables $u=\eta v$,

$$\int_{\mathbb{R}^2}\left\|D^j\psi_{p,\eta}(u)\right\|_{\mathrm{op}}\,du = \eta^{-j}\int_{\mathbb{R}^2}\left\|D^j\psi_p(v)\right\|_{\mathrm{op}}\,dv,$$

which proves the claimed scaling. □

Fix $p \geq 1$ and choose

$$\eta_I := \min\left\{\frac{1}{8}, \frac{\epsilon_0}{8L_I}\right\}. \tag{20}$$

This is a positive rational number of polynomial bit length, and

$$\eta_I \leq \frac{1}{8}, \qquad \sqrt{2}L_I\eta_I \leq \frac{\epsilon_0}{4}.$$

Using the probability kernel $\psi_{p,\eta}$ defined in (19), set

$$G_I(x) := \int_{\mathbb{R}^2} F_I(x-u)\psi_{p,\eta_I}(u)\,du. \tag{21}$$

The integral is well defined because F_I is bounded and the kernel is compactly supported.

Lemma D.5. *The function G_I belongs to $\mathcal{B}^{p,1}(\mathbb{R}^2)$. In polynomial time, one can compute a positive rational number $Q_I \geq 1$ such that*

$$\sup_x |G_I(x)| \leq Q_I, \qquad \sup_x \left\|D^j G_I(x)\right\|_{\mathrm{op}} \leq Q_I \quad (1 \leq j \leq p),$$

and

$$\left\|D^p G_I(x) - D^p G_I(y)\right\|_{\mathrm{op}} \leq Q_I \left\|x - y\right\|.$$

Moreover, $Q_I \leq 2^{\mathrm{poly}_p(n)}$, where the polynomial may depend on the fixed order p.

Proof of Lemma D.5. Write $\eta = \eta_I$ and $G = G_I$. Since F_I is bounded and the kernel and its derivatives through order $p+1$ are continuous with compact support, differentiation under the convolution integral yields

$$D^j G(x) = \int_{\mathbb{R}^2} F_I(v) D^j \psi_{p,\eta}(x - v)\, dv, \qquad 0 \leq j \leq p+1. \tag{22}$$

For example, on a fixed bounded neighborhood of x, all translated kernel supports lie in one compact set, so the boundedness of F_I supplies a common integrable dominating bound. The resulting derivatives are continuous. Taking the operator norm in (22) and applying Lemma D.4 gives

$$\sup_x \left\|D^j G(x)\right\|_{\mathrm{op}} \leq B_I A_{p,j} \eta^{-j}, \qquad 1 \leq j \leq p+1. \tag{23}$$

For the function itself, the probability-kernel identity gives $|G(x)| \leq B_I$.

To verify the required Lipschitz property in the operator norm, integrate the $(p+1)$-st derivative along a segment. For unit vectors $v_1, \ldots, v_p$,

$$\begin{aligned}&\left|\left(D^p G(x) - D^p G(y)\right)[v_1, \ldots, v_p]\right| \\ &\quad \leq \int_0^1 \left\|D^{p+1} G(y + t(x-y))\right\|_{\mathrm{op}} \left\|x - y\right\| dt \leq B_I A_{p,p+1} \eta^{-(p+1)} \left\|x - y\right\|.\end{aligned}$$

Taking the supremum over the unit vectors proves that $D^p G$ is globally Lipschitz. This also makes explicit that the derivative estimates use the operator norm of the entire tensor, rather than only coordinatewise derivative bounds.

Let $A_p := \max\{1, A_{p,1}, \ldots, A_{p,p+1}\}$, and choose

$$Q_I := 1 + B_I A_p \eta_I^{-(p+1)}. \tag{24}$$

Since $\eta_I \leq 1$, this rational number bounds the function, every derivative through order p, and the Lipschitz constant just obtained. It can be computed in polynomial time for fixed p.

By (20),

$$\eta_I^{-1} = \max\{8, 8L_I/\epsilon_0\} \leq 2^{\mathrm{poly}(n)}.$$

The same type of bound holds for B_I. The constants A_p and the exponent $p+1$ depend only on fixed p, so (24) has polynomial bit length and satisfies $Q_I \leq 2^{\mathrm{poly}_p(n)}$. Taking every declared bound equal to Q_I is therefore a valid finite higher-smoothness annotation. □

Lemma D.6. *For every $x \in \mathbb{R}^2$,*

$$\left\|\nabla G_I(x) - \nabla F_I(x)\right\| \leq \frac{\epsilon_0}{4}.$$

Proof of Lemma D.6. The source function F_I is continuously differentiable with bounded gradient. Differentiating (21) with respect to its first factor gives

$$\nabla G_I(x) = \int_{\mathbb{R}^2} \nabla F_I(x - u) \psi_{p,\eta_I}(u)\, du.$$

Since the kernel integrates to one, subtraction of $\nabla F_I(x)$ and the triangle inequality yield

$$\begin{aligned}\|\nabla G_I(x) - \nabla F_I(x)\| &\le \int_{\mathbb{R}^2} \|\nabla F_I(x-u) - \nabla F_I(x)\|\, \psi_{p,\eta_I}(u)\, du \\ &\le L_I \int_{\mathbb{R}^2} \|u\|\, \psi_{p,\eta_I}(u)\, du \\ &\le \sqrt{2} L_I \eta_I \le \frac{\epsilon_0}{4}.\end{aligned}$$

The penultimate inequality uses $\|u\| \le \sqrt{2}\eta_I$ throughout the support. This bound is uniform over $\mathbb{R}^2$, including neighborhoods of cell boundaries and periodic seams. □

Lemma D.7 (No spurious approximate stationary points)**.** *If $x \in \mathbb{R}^2$ satisfies $\|\nabla G_I(x)\| \le \epsilon_0/4$, then*

$$\|\nabla F_I(x)\| \le \frac{\epsilon_0}{2} < \epsilon_0.$$

In particular, a rational such point decodes a solution of I.

Proof of Lemma D.7. By Lemma D.6,

$$\|\nabla F_I(x)\| \le \|\nabla F_I(x) - \nabla G_I(x)\| + \|\nabla G_I(x)\| \le \frac{\epsilon_0}{4} + \frac{\epsilon_0}{4} = \frac{\epsilon_0}{2} < \epsilon_0.$$

For rational x, the decoder from Lemma D.2 therefore returns a valid solution of I. □

D.4 Exact evaluation and polynomial encoding lengths

Analytic smoothing alone is insufficient for a white-box reduction. The next lemma implements both evaluator descriptions and bounds their bit complexity, including queries crossing cell boundaries or periodic seams.

Lemma D.8. *For fixed p, polynomial-time evaluator descriptions for G_I and ∇G_I can be constructed from I. At a rational point x, both evaluations are exact rational numbers with bit lengths polynomial in the length of I and the bit length of x. Consequently, evaluations to r bits of accuracy can be computed in time polynomial in these lengths and r.*

Proof of Lemma D.8. Let the query point $x = (x_1, x_2)$ be rational, with total bit length ℓ_x. The convolution uses values of F_I only in

$$x + [-\eta_I, \eta_I]^2.$$

Each side of this square has length $2\eta_I \le 1/4$. Since the source grid has unit spacing, the square intersects at most two grid intervals in each coordinate with positive length. Thus it intersects at most four cells with positive area. Intersections consisting only of an edge or a vertex have measure zero and may be discarded. These statements remain true across periodic seams, because the periodic extension is still piecewise polynomial on the unit integer grid.

The relevant cells can be enumerated using rational comparisons and integer floor operations. For a cell $C_{a,b} = [a, a+1] \times [b, b+1]$, write its source polynomial as $P_{a,b}(v)$. Here $a, b \in \mathbb{Z}$ are cell coordinates. By Lemma D.2, its coefficients in local coordinates can be recovered from a constant number of circuit evaluations. Reducing the cell coordinates modulo the source period and evaluating these local rules takes time polynomial in the length of I and the bit lengths of the coordinates.

In the integration variable u, the corresponding rectangle is

$$\begin{aligned}R_{a,b}(x) &= [-\eta_I, \eta_I]^2 \cap (x - C_{a,b}) \\ &= \big([-\eta_I, \eta_I] \cap [x_1 - a - 1, x_1 - a]\big) \\ &\qquad \times \big([-\eta_I, \eta_I] \cap [x_2 - b - 1, x_2 - b]\big).\end{aligned}$$

Its endpoints are rational and can be computed exactly. The convolution is the sum, over at most four nonempty rectangles, of

$$\int_{R_{a,b}(x)} P_{a,b}(x-u)\,\eta_I^{-2}\theta_p(u_1/\eta_I)\theta_p(u_2/\eta_I)\,du. \tag{25}$$

Inside each rectangle, the integrand is a rational-coefficient polynomial. Its degree in each coordinate is at most $3+(2p+4)=2p+7$. For a monomial $c\,u_1^k u_2^\ell$ on a rectangle $[\alpha_1,\beta_1]\times[\alpha_2,\beta_2]$, its integral is exactly

$$c\,\frac{\beta_1^{k+1}-\alpha_1^{k+1}}{k+1}\,\frac{\beta_2^{\ell+1}-\alpha_2^{\ell+1}}{\ell+1}.$$

All terms in this expression are rational. For fixed p, the number of coefficients, the exponents, and the number of rectangles are constants. The coefficients of the source polynomial, the query coordinates, and η_I have polynomial bit length. A constant number, depending on p, of rational additions, multiplications, and fixed-degree powers therefore gives an exact rational value for (25), with polynomial bit length and running time. Summing the rectangles proves the value-evaluation claim.

For the gradient, use

$$\nabla G_I(x)=\int_{\mathbb{R}^2}\nabla F_I(x-u)\psi_{p,\eta_I}(u)\,du.$$

On each rectangle, replace $P_{a,b}$ by its coordinate derivatives. Their degrees are no larger, so exactly the same integration and bit-complexity argument applies. No differentiation of moving cell endpoints is needed.

Finally, the evaluator description consists of a fixed algorithm for cell enumeration and polynomial integration, together with the circuit for I, the rational radius η_I, and the fixed kernel coefficients. It has polynomial size and is constructible in polynomial time. The established running-time bound supplies a polynomial clock, giving exact rational evaluators of the form required in Definition 4.1. As an optional additional operation, rounding an exact rational output to absolute error at most 2^{-r} takes time polynomial in its bit length and r; this rounding is not needed for the search reduction. □

D.5 The reductions and their normalization

The preceding lemmas show that the smoothed function has the required regularity, preserves source decoding, and admits exact polynomial-time evaluation. We now combine these properties into the search reductions.

Proposition D.9 (Hardness in dimension two)**.** *For every fixed finite $p\ge 1$, $\mathsf{B\text{-}Stationary}_{2,p}$ is PLS-hard, even on inputs satisfying the higher-smoothness promise.*

Proof of Proposition D.9. Given an Iter instance I, construct G_I using (20) and (21), and set

$$\epsilon_I:=\frac{\epsilon_0}{4},\qquad M_0=M_1=\cdots=M_p=\Lambda_p:=Q_I,$$

with Q_I from (24). Lemma D.5 proves that all these declared bounds hold. Lemma D.8 supplies polynomial-time evaluator descriptions of polynomial size, and all numerical parameters have polynomial bit length. Thus this is a polynomial-time constructible input satisfying the higher-smoothness promise.

Because the input is valid, no accepted base violation certificate exists. Any solution must therefore be a finitely encoded rational point x satisfying $\|\nabla G_I(x)\|\le\epsilon_I$. Lemma D.7 gives $\|\nabla F_I(x)\|\le\epsilon_0/2$, so $\mathrm{Dec}(I,x)$ is a valid solution of the original Iter instance. The decoder runs in time polynomial in the input and output bit lengths, and the underlying finite search model bounds accepted output lengths polynomially in the input length. This proves a polynomial-time search reduction and hence PLS-hardness. □

Proposition D.10 (Hardness in all fixed dimensions at least two)**.** *For every fixed finite $p\ge 1$ and every fixed $d\ge 2$, $\mathsf{B\text{-}Stationary}_{d,p}$ is PLS-hard, even on inputs satisfying the higher-smoothness promise.*

Proof of Proposition D.10. The case $d = 2$ is Proposition D.9. For fixed $d > 2$, let $\pi : \mathbb{R}^d \to \mathbb{R}^2$ denote projection onto the first two coordinates, and define

$$H_I(x) := G_I(\pi x) = G_I(x_1, x_2).$$

For every $1 \le j \le p$,

$$D^j H_I(x)[v_1, \dots, v_j] = D^j G_I(\pi x)[\pi v_1, \dots, \pi v_j].$$

Since π has operator norm one, each derivative bound transfers from G_I to H_I. The same calculation at two points, together with $\|\pi x - \pi y\| \le \|x - y\|$, transfers the Lipschitz bound on the p-th derivative. The function bound also remains unchanged. Thus $H_I \in \mathcal{B}^{p,1}(\mathbb{R}^d)$ with the same declarations.

Its gradient is

$$\nabla H_I(x) = \big(\partial_1 G_I(x_1, x_2), \partial_2 G_I(x_1, x_2), 0, \dots, 0\big),$$

so

$$\|\nabla H_I(x)\| = \|\nabla G_I(x_1, x_2)\|.$$

The value and gradient evaluators for H_I call those for G_I on the first two coordinates and append zeros. This has polynomial cost for fixed d. Any solution to the resulting valid promised instance supplies an approximate stationary point of G_I in its first two coordinates and therefore decodes a solution of I. This proves hardness for every fixed $d \ge 2$. □

Proof of Theorem 4.2. Proposition D.1 gives membership of the totalized problem in PLS. Propositions D.9 and D.10 give PLS-hardness by reductions whose outputs all satisfy the higher-smoothness promise. The two assertions prove the theorem. In particular, an algorithm required to succeed only on promised inputs still solves the hard instances produced by these reductions. □

Remark D.11 (Normalization and accuracy). The hard family can also be normalized so that every displayed value, derivative, and Lipschitz bound is at most one. Specifically, use $\widetilde{G}_I = G_I/Q_I$ and accuracy

$$\widetilde{\epsilon}_I = \frac{\epsilon_0}{4Q_I}.$$

The normalization remains a polynomial-time reduction because Q_I has polynomial bit length. The accuracy must be rescaled together with the objective; this argument does not assert hardness at a fixed positive accuracy while all analytic bounds are normalized.

To verify the normalization assertion in Remark D.11, divide the objective and both evaluator outputs by Q_I. Every derivative and every derivative Lipschitz constant scales by the same factor, so the normalized declarations can all be one. If

$$\left\|\nabla \widetilde{G}_I(x)\right\| \le \widetilde{\epsilon}_I = \frac{\epsilon_0}{4Q_I},$$

then $\|\nabla G_I(x)\| \le \epsilon_0/4$, which is the threshold required by Lemma D.7. Both division by Q_I and construction of $\widetilde{\epsilon}_I$ have polynomial bit complexity. The same argument applies to H_I after the dimension embedding. This proves normalized hardness with the stated, instance-dependent accuracy.