

# Function-free Optimization via Comparison Oracles

Katya Scheinberg\*      Zikai Xiong†

May 2026

## Abstract

In this work, we study optimization specified only through a comparison oracle: given two points, it reports which one is preferred. We call it function-free optimization because we do not assume access to, nor the existence of, a canonical application-given objective function. Instead, our goal is to find a most-preferred feasible point, which we call an optimal solution. This model arises in preference- and ranking-based settings, where the objective values and derivatives are unavailable, meaningless, or non-identifiable. Even if a representative function exists for the preference relation, it may be nonsmooth, nonconvex, or even discontinuous. We develop an analytical and algorithmic framework based on the geometry of preference level sets, which remains well-defined from comparisons alone. We introduce a new optimality measure, the level-set optimality gap, defined as the distance from the preference level set to the optimal solutions, and also the regularity radius, which plays the role of a stationarity certificate. Under regularity of the preference relation in a  $d$ -dimensional Euclidean space, we propose a method for estimating normal directions to accuracy  $\epsilon$  using  $O(d \log(d/\epsilon))$  comparisons, nearly matching a lower bound of  $\Omega(d \log(1/\epsilon))$ . Under convexity and regularity of the preference relation and a local growth condition on the regularity radius, the resulting normal direction descent method reaches an  $\epsilon$  level-set optimality gap using at most  $\tilde{O}(dD^2/\epsilon^2)$  comparisons, over  $O(D^2/\epsilon^2)$  normal direction estimation steps. Here  $D$  is the distance from the initial point to the optimal set. This number of normal direction estimation steps matches the lower bound of  $\Omega(D^2/\epsilon^2)$  for normal direction span-based methods. Since prior knowledge in practical applications of function-free optimization is usually very limited, we also develop adaptive schemes for both estimating the normal direction and solving the optimization problem. These adaptive schemes match the fixed-parameter complexity bounds up to logarithmic factors.

**Key words:** derivative-free optimization, comparison oracle, convex optimization, oracle complexity

**MSC codes:** 90C56, 90C60, 90C25

## 1 Introduction

Comparison- and preference-based feedback is often the natural form of information in applications where objective values are unavailable or not clearly defined. In search and ranking systems, user behavior can reveal relative preferences rather than utility values [44, 43]. Related preference-based optimization models also arise in recommender systems and A/B testing, where a latent objective is queried through pairwise comparisons [10]. In human-in-the-loop personalization and clinical

---

\*H. Milton School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. [katya.scheinberg@isye.gatech.edu](mailto:katya.scheinberg@isye.gatech.edu)

†H. Milton School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. [zxiong84@gatech.edu](mailto:zxiong84@gatech.edu)

tuning, comparisons between alternatives are more reliable than numerical scoring [41, 46]. In preference-based reinforcement learning and reinforcement learning from human feedback, signals are commonly collected as preferences over trajectory segments or rankings of model outputs rather than through direct access to a reward function [8, 5, 32]. These examples suggest that, in many applications, the observable object is not a numerical objective value but a preference relation between feasible alternatives.

Comparing the objective values of two (or more) points, as component of an algorithm, has long been used in derivative-free optimization. However, most algorithms regard this procedure as a way of using function values, or zeroth-order oracles, rather than as a separate (weaker) oracle. For example, classical direct-search methods compare trial points (in terms of their objective values) with the best point found so far [13], and the Nelder–Mead method is likewise order-based [27, 19]. These methods tend to have poor or no convergence guarantees. More recently, several algorithms that use function comparisons and yet have some complexity guarantees (under certain structural assumptions on the objective function) have been proposed for zeroth-order optimization [14, 2, 9] and gradient estimation [3, 39]. Paper [14] is the first in this line of work which we are aware of that explicitly presents the operation of comparing two points as a so-called “comparison oracle,” thus explicitly avoiding the need to compute function values.

“Comparison oracle,” namely an oracle that returns the relative order of the function values at two queried points have appeared in a variety of recent papers. Comparison oracle based optimization is also studied under the name of dueling optimization [35, 36, 33]. Related notions sub-zeroth order oracles [16] and order oracles [23] are other names of comparison oracles. Beyond pairwise comparisons, ranking oracles can also be obtained from multiple pairwise comparisons [38].

Both the classical and the recent methods treat comparisons primarily as a weak interface to a latent scalar objective function. Specifically, comparison oracle is generated by value differences of an underlying function. More importantly the guarantees are stated in terms of properties of that function, such as function-value gaps, regret, or gradient norms and the complexity results, when they exists, rely on the properties of the function. This creates a gap with the underlying applications. In preference-driven problems, there may not exist an application-given scalar objective function. Even when an objective function representation exists, it may be non-identifiable from comparisons, nonunique, or only a surrogate introduced for modeling convenience. The authors of [9] observe the fact that multiple functions can result in identical comparison oracles and state their results in terms of the “best” strongly convex function that “fits” the oracle.

In summary, prior work aims to optimize a scalar function through weaker oracles. In contrast, our central goal is to formulate an optimization problem when the preference relation itself is the basic object defining optimality. The main question motivating this paper is:

**Question 1.** *How should we formulate and analyze optimization using only a preference relation, without choosing any application-given scalar objective function representative?*

This question is not merely a matter of notation. Figure 1 shows three functions that are dramatically different. One is convex and smooth except for the optimal solution, another is nonconvex, and another is discontinuous. They have different graphs but their level sets have the same shape (spheres centered at the origin). They induce the same preference relation:  $x$  is preferred to  $y$  if and only if  $\|x\|_2 \leq \|y\|_2$ . Hence, these three functions are indistinguishable from the perspective of comparisons, and running any comparison based algorithm on these three different functions will observe the same comparison feedbacks and yield exactly the same trajectories of iterates. Therefore, manually choosing one representative function introduces artificial information that is invisible to comparisons. To answer Question 1, we need assumptions, algorithms, and guarantees expressed directly in terms of the preference relation.

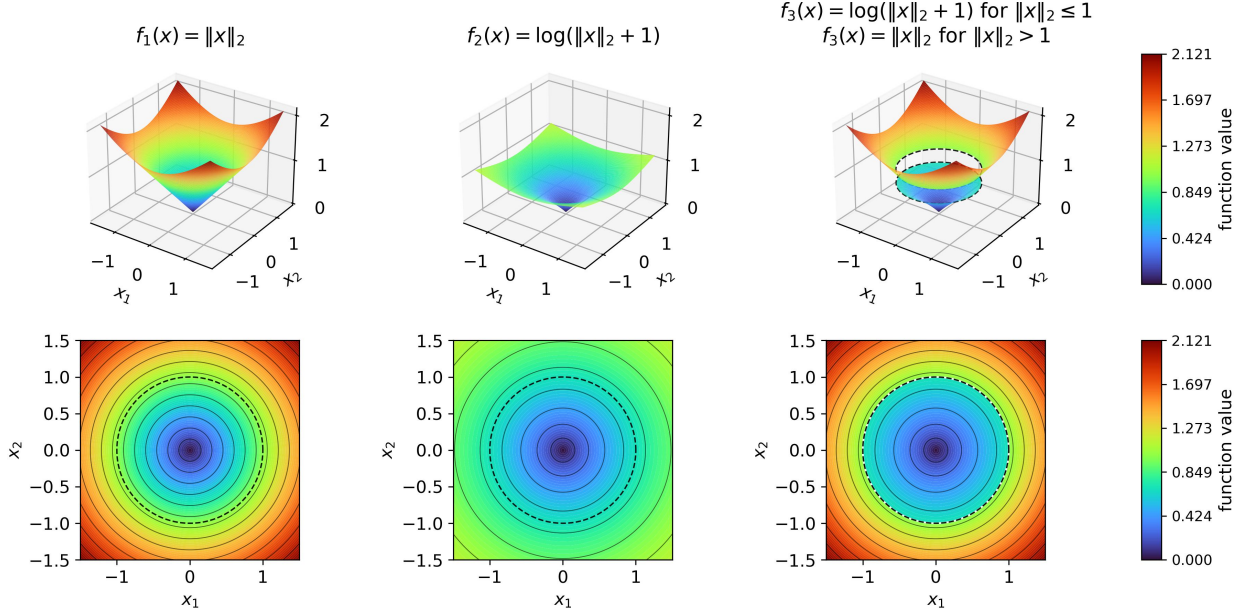


Figure 1: Graphs and level sets of three functions on  $x = (x_1, x_2) \in \mathbb{R}^2$ :  $f_1(x) := \|x\|_2$ ,  $f_2(x) := \log(\|x\|_2 + 1)$ , and  $f_3(x) := \log(\|x\|_2 + 1)$  if  $\|x\|_2 \leq 1$ , and  $f_3(x) := \|x\|_2$  otherwise.

Our answer to Question 1 is a *function-free optimization* framework, which is also our first main contribution. We model the available information by a binary relation  $\preceq$  on a  $d$ -dimensional Euclidean space  $\mathcal{X}$ , interpreted as a preference relation:

$$x \preceq y \quad \text{means that } x \text{ is preferred to } y; \text{ equivalently, } x \text{ is no worse than } y. \quad (1)$$

Given a feasible set  $\mathcal{C} \subseteq \mathcal{X}$ , the optimization task is to identify a most-preferred feasible point under  $\preceq$ :

$$\text{find } x \in \mathcal{C}, \quad \text{s.t. } x \preceq y \text{ for all } y \in \mathcal{C}. \quad (2)$$

This is the main optimization problem studied in this paper. We call  $x^* \in \mathcal{C}$  an optimal solution if it is no worse than every feasible point. Equivalently, the set of all optimal solutions is

$$\mathcal{X}^* := \{x \in \mathcal{C} : x \preceq y \text{ for all } y \in \mathcal{C}\}. \quad (3)$$

To measure approximate optimality in this framework, we propose new measures defined purely through the preference relation. The main near-optimality measure in this paper is the *level-set optimality gap*: intuitively, the preference level set through  $x$  consists of all points that are indifferent to  $x$ , and the level-set optimality gap measures the distance from this preference level set to the optimal set  $\mathcal{X}^*$ . This quantity is defined entirely from the preference relation and the Euclidean geometry. We also introduce the *regularity radius*, a local geometric scale describing how regularly the preference level set behaves near the current point. A small regularity radius is a function-free analogue of near-stationarity, and under a local growth condition it can also imply near-optimality. The formal definitions of preference level sets, the level-set optimality gap, the regularity radius, and the assumptions on the preference relation are given in Section 2. Importantly, assumptions such as plateau-freeness, convexity, and regularity are imposed directly on the preference relation.

Once having the function-free optimization framework, the next question is algorithmic: how can one solve the problem without gradients or function values? In smooth optimization, the

gradient is critical because it indicates how the function changes and which direction improves the objective. In function-free optimization, a particular gradient is not available and it is not intrinsic to the preference relation. The replacement used in this paper is the *normal direction* to the current preference level set. Roughly speaking, the normal direction is the direction in which one leaves the current preference level fastest. Thus, moving in the opposite direction usually leads to better points. Therefore, in this paper, we are interested in estimating normal directions and using normal directions to design algorithms.

Throughout the paper, both the normal direction estimation problem and the optimization problem are guided by two principles: optimality and adaptivity. Optimality matters because we seek worst-case complexity guarantees that cannot be substantially improved in general. It is a valuable benchmark for understanding the fundamental limits and remaining potential of an algorithm. Adaptivity matters because, in applications with only ordinal feedback, the relevant local preference parameters are rarely known in advance, so the algorithm should not rely on parameters that need knowledge of them. Therefore, we seek comparison based methods that are near-optimal in the worst case while also adjusting automatically to unknown local geometry.

This leads to the remaining two questions studied in this paper:

**Question 2.** *Can normal directions be estimated from comparisons as efficiently as possible, while remaining adaptive to local preference relation geometry?*

**Question 3.** *Once normal directions can be estimated, can they be used to design optimization methods that are likewise optimal and adaptive?*

Our answers to Questions 2 and 3 form our second and third main contributions of this paper, below we give a high-level summary.

Our answer to Question 2 is a near-optimal adaptive normal-direction estimation method that uses only pairwise comparisons. The key idea is an orthogonal-complement construction. Instead of estimating the normal direction directly, the method constructs  $d - 1$  approximately orthogonal directions to the normal direction, and then the one-dimensional orthogonal complement approximates the normal direction. The method estimates the normal direction to accuracy  $\epsilon$  using  $O(d \log \frac{d}{\epsilon})$  pairwise comparisons, provided the comparison radius is chosen small enough. We also prove a minimax lower bound of order  $\Omega(d \log \frac{1}{\epsilon})$  comparisons for normal-direction estimation. Hence the fixed-radius estimator is nearly optimal. We then give an adaptive version for convex preference relations. This version uses line search to choose the comparison radius automatically and it achieves the same leading dependence on  $d$  and  $\epsilon$ , with only logarithmic factor overhead.

To answer Question 3, we study mainly in the class of *normal-direction span-based methods* for convex preference relations, which parallels the first-order methods in classical convex optimization. In this class, we propose the normal direction descent method (NDD), and establish a two-case guarantee in terms of the level-set optimality gap and the regularity radius. To ensure smaller level-set optimality gap, the method needs to use more comparisons, while getting a smaller regularity radius only requires choosing a smaller comparison radius when estimating normal directions. Under a local growth condition of the regularity radius, this yields an  $\epsilon$  level-set optimality gap guarantee of consuming at most  $\tilde{O}\left(\frac{dD^2}{\epsilon^2}\right)$  pairwise comparisons over  $O\left(\frac{D^2}{\epsilon^2}\right)$  normal-direction estimation steps. Here  $D$  denotes the distance from the initial point to the optimal solutions, and  $\tilde{O}(\cdot)$  hides only logarithmic factors of  $d$  and  $\frac{D}{\epsilon}$ . This normal direction complexity is optimal as we prove a minimax lower bound of  $\Omega\left(\frac{D^2}{\epsilon^2}\right)$  for all normal-direction span-based methods. We also develop an adaptive method, namely adaNDD. It does not require prior knowledge of  $D$  or the local growth parameters, and it matches the rates of NDD up to a logarithmic factor.

Before concluding this section we want to emphasize an important distinction between two settings induced by comparison oracles. The setting we consider here is based on a noiseless oracle. In the literature, comparison oracles are often allowed to be noisy, meaning that they may return the wrong order with some probability. Among them, when the oracle gives the correct answer with probability  $\frac{1}{2}$  plus a constant positive term, which we call "advantage", repeated independent queries can simulate a noiseless oracle with high probability [35]. A more general model assumes that the probability of giving the correct answer depends on the function-value difference between the two points [14, 36]. The more general model however uses function value in the oracle definition, thus the setting is inherently tied to an objective function and is no longer function-free. Such oracles are thus beyond the study in this paper.

## 1.1 Literature review

We now summarize the key aspects of the existing relevant literature mentioned above.

Comparison-based rules have a long history in derivative-free optimization. Early direct-search methods iterate by comparing trial solutions with the best point found so far [13], the Nelder–Mead method orders simplex vertices by objective value and uses this ordering to choose reflection, expansion, contraction, and shrink steps [27]. Modern direct-search and pattern-search methods formalize polling and acceptance rules based on sampled objective values [40, 18, 1]. However, Nelder–Mead method has only limited general convergence theory and can converge to non stationary points [19, 24], while direct-search methods that enjoy reasonable complexity guarantees rely on so-called *sufficient-decrease conditions* which required knowledge of the difference of function values and not just how they compare [42, 18]. While purely comparison-based variants of direct search methods exist they have much weaker (exponential time) worst-case guarantees.

Some recent zeroth-order methods use comparison-based update rules and manage to develop complexity guarantees under structural assumptions on the objective function. Bergou et al. [2] propose the stochastic three-point method, which samples a direction  $s$  and moves to the best point among  $x$ ,  $x + \alpha s$ , and  $x - \alpha s$ . They give complexity guarantees for smooth objectives under various convexity assumptions. Golovin et al. [9] propose the GradientLess Descent method for smooth strongly convex objective. Each iteration of the method searches around the current point at different radii and iteratively updates to the best point among the search points. Golovin et al. [9] realizes that since the method is purely comparison-based, it also applied to monotone transformations of the objective and thus their complexity results are stated in terms of the "best" strongly convex function that can be obtained under such transformation.

A more recent line of work treats pairwise comparisons as an explicit oracle for optimization. Jamieson et al. [14] give lower bounds for derivative-free optimization with comparison oracles and propose a comparison-based randomized coordinate-descent method for smooth strongly convex objectives, where comparisons are used to perform line search along coordinate directions. Lobanov et al. [23] develop a comparison-based accelerated coordinate-descent method for smooth strongly convex objectives. Noisy comparison oracles have also been studied. If the oracle returns the correct comparison with a fixed advantage over random guessing, then repeated independent queries can simulate a noiseless comparison oracle with high probability, while the number of queries scales quadratically with the inverse of the advantage over random guessing [14, 35]. Saha et al. [35] study smooth convex and strongly convex optimization under such fixed-advantage noisy comparisons, and call this setting *dueling* convex optimization. Their method uses comparisons of randomly perturbed points to estimate normalized descent directions and then applies normalized gradient descent. This was later extended to noisy oracles whose advantage depends on the function-value difference [36]. These work treat comparisons as a weak interface to the objective function, so the

convergence guarantees are established in terms of either function-value gap or gradient norms, and the algorithm needs prior knowledge of parameters of the underlying objective (such as smoothness parameter  $L$ ).

Another related line of work focuses on estimating normalized gradients for smooth functions. Cai et al. [3] assume sparse gradients and use one-bit compressed sensing to estimate a normalized gradient from pairwise comparisons. A related estimator is later used by Chen et al. [4] for nonconvex smooth optimization in preference alignment. Karabag et al. [16] maintain a cone of plausible directions and shrink this cone using half-space cuts extracted from comparison feedback. This method achieves logarithmic dependence on the estimation error  $\epsilon$ , but its cost grows quadratically with the dimension. They also use a variant of the ellipsoid method to solve smooth convex optimization problems. Zhang and Li [45] propose a two-phase method: first finding a dominant gradient component with the largest magnitude, and then estimating the ratio between this component and each of the other components. This improves the dimension dependence, but requires the comparison radius to be in a restrictive range. Very recently, while preparing the first version of this manuscript, we found a concurrent work [39], which further improves the method of [45] by first rotating the basis toward a reference direction that is close to the normalized gradient. This loosens the requirement on the comparison radius, although the guarantee is no longer deterministic. Tao et al. [39] also give a lower bound and show that their method is nearly optimal in terms of the comparison complexity. When the preference relation is realized by a smooth objective, the normalized gradient is identical to the normal direction. Our normal direction estimation method is deterministic and also nearly optimal, and furthermore for convex preference relations, it is adaptive to the local geometry. A more detailed comparison of these methods and our method will be given in Section 3.

Comparison feedback has also been extended to other settings. Saha et al. [34] study batched and multiway comparison oracle feedbacks and show that they can improve convergence rates. Tang et al. [38] use ranking oracles, which return an ordering over several queried points and can be simulated by multiple pairwise comparisons, to estimate descent directions and develop a stochastic gradient descent type method. Very recently, Ren et al. [33] study optimization on Riemannian manifolds using comparison oracles. El Bakkali et al. [6] study nonsmooth nonconvex optimization with noisy comparison oracles where the advantage of noisy oracles depends on the function-value difference via a link function whose expression is explicitly known.

## 1.2 Organization and notation

**Notation.** In this paper, we let  $(\mathcal{X}, \langle \cdot, \cdot \rangle)$  be a  $d$ -dimensional real Euclidean vector space, and let  $\|x\| := \sqrt{\langle x, x \rangle}$  be the induced norm. For a set  $A \subseteq \mathcal{X}$ , we write  $\text{int}(A)$ ,  $\bar{A}$ , and  $\partial A$  for its interior, closure, and boundary, respectively. Let  $\text{dist}(x, A) := \inf_{y \in A} \|x - y\|$  denote the Euclidean distance from  $x$  to  $A$ , and let  $\text{dist}(A_1, A_2) := \inf_{x \in A_1, y \in A_2} \|x - y\|$  denote the distance between two sets  $A_1$  and  $A_2$  in  $\mathcal{X}$ . Let  $B(x, r)$  denote the closed Euclidean ball centered at  $x$  with radius  $r$ . Let  $\mathbb{S}^{d-1}$  denote the unit sphere in  $\mathcal{X}$ . For set  $\mathcal{C}$ , we use  $\text{diam}(\mathcal{C}) := \sup_{x, y \in \mathcal{C}} \|x - y\|$  to denote its diameter. When  $\mathcal{C}$  is closed and convex,  $\Pi_{\mathcal{C}}(x)$  denotes the Euclidean projection of  $x$  onto  $\mathcal{C}$ . For nonzero vectors  $u$  and  $v$ , we define  $\angle(u, v) := \arccos\left(\frac{\langle u, v \rangle}{\|u\| \|v\|}\right) \in [0, \pi]$ . The notation  $a = O(b)$  means  $a \leq Cb$  for some absolute constant  $C > 0$ , and  $a = \Omega(b)$  means  $a \geq cb$  for some absolute constant  $c > 0$ . Similarly,  $a \lesssim b$  means  $a \leq Cb$  for an absolute constant  $C > 0$ . The notation  $a = o(b)$  means  $a/b \rightarrow 0$  in the stated limiting regime. We use  $\tilde{O}(\cdot)$  to suppress absolute constants and logarithmic factors.

**Outline of the paper.** The rest of the paper is organized as follows. Sections 2, 3, and 4 correspond to the three main contributions of the paper, answering Questions 1, 2, and 3, respectively. In Section 2, we propose the function-free optimization framework. And we also give definitions of the ingredients of the framework, including common assumptions, and optimality and stationarity criteria. We also explain the relation to comparison-based optimization when a scalar objective function exists. In Section 3, we study normal direction estimation using pairwise comparisons. We present the fixed comparison radius normal direction estimator, the adaptive version for convex preference relations, and the lower bound showing near-optimality of these two methods. We also discuss how normal directions directly enable cutting-plane methods. In Section 4, we propose and analyze the normal direction descent method NDD, and its adaptive version adaNDD. We also prove lower bounds for the general class of normal-direction span-based methods and show the near-optimality of NDD and adaNDD. Finally, in Section 5, we summarize the contributions and discuss future directions.

## 2 Function-free optimization via preferences and comparisons

In this section, we introduce the function-free optimization framework.

### 2.1 Preference relation and comparison feedback

Recall that the preference relation  $\preceq$  is defined in (1), on a  $d$ -dimensional Euclidean space  $(\mathcal{X}, \langle \cdot, \cdot \rangle)$  (e.g.,  $\mathcal{X} = \mathbb{R}^d$ ), equipped with its induced norm  $\|\cdot\|$ . Throughout this paper, we assume the preference relation  $\preceq$  is complete and transitive, defined as follows.

**Definition 2.1** (Complete and transitive preferences). The preference relation  $\preceq$  is a complete and transitive preference relation if

- (i) (*Completeness*) for all  $x, y \in \mathcal{X}$ , either  $x \preceq y$  or  $y \preceq x$  (or both);
- (ii) (*Transitivity*) for all  $x, y, z \in \mathcal{X}$ , if  $x \preceq y$  and  $y \preceq z$ , then  $x \preceq z$ .

Given  $\preceq$ , we define the strict preference relation  $\prec$  and the indifference relation  $\sim$  by

$$x \prec y \iff (x \preceq y) \text{ and not } (y \preceq x), \quad x \sim y \iff (x \preceq y) \text{ and } (y \preceq x).$$

Intuitively,  $x \prec y$  means that  $x$  is strictly preferred to  $y$ , while  $x \sim y$  means that  $x$  and  $y$  are tied (indifferent).

Given a feasible set  $\mathcal{C} \subseteq \mathcal{X}$ , the goal is to find a point  $x^* \in \mathcal{C}$  such that  $x^* \preceq y$  for all  $y \in \mathcal{C}$ . We have defined this problem in (2) and its solution set, denoted by  $\mathcal{X}^*$ , has been defined in (3). Throughout this paper we assume having access to efficient implementation of Euclidean projections onto  $\mathcal{C}$ .

Since we do not have numerical objective values, some typical oracles used by classical optimization algorithms (e.g., function values or gradients) are unavailable. Instead, we access the preference relation  $\preceq$  through pairwise comparisons. A comparison between two points  $x, y \in \mathcal{X}$  reveals which of the three cases holds:  $x \prec y$ ,  $x \sim y$ , or  $y \prec x$ .

**Definition 2.2** (Comparison oracle). A comparison oracle is a mapping  $\mathcal{C}_{\text{cmp}} : \mathcal{X} \times \mathcal{X} \rightarrow \{-1, 0, +1\}$  such that for any  $x, y \in \mathcal{X}$ :  $\mathcal{C}_{\text{cmp}}(x, y) = -1$  if  $y \prec x$ ,  $\mathcal{C}_{\text{cmp}}(x, y) = 0$  if  $y \sim x$ , and  $\mathcal{C}_{\text{cmp}}(x, y) = +1$  if  $x \prec y$ .

Throughout the paper, one use of the comparison oracle is simply called a comparison and the comparison complexity of an algorithm is the total number of comparisons it uses.

Before presenting algorithms for (2), we introduce several basic geometric concepts defined directly from the preference relation  $\preceq$ . These concepts will be used later to design algorithms and to analyze their performance.

## 2.2 Preference sublevel sets and level sets

Using the preference relation  $\preceq$ , we define the (*preference*) *sublevel set* and (*preference*) *level set* at any point  $x \in \mathcal{X}$  as follows.

**Definition 2.3** (Preference sublevel and level sets). For any  $x \in \mathcal{X}$ , the preference sublevel set at  $x$  and the preference level set at  $x$  are defined as

$$\mathcal{S}_x := \{y \in \mathcal{X} : y \preceq x\}, \quad \mathcal{L}_x := \{y \in \mathcal{X} : y \sim x\}. \quad (4)$$

We call them “preference sublevel sets” and “preference level sets” to distinguish them from the usual function value based sublevel sets and level sets. For simplicity, we will omit the word “preference” when there is no risk of confusion. The sets  $\mathcal{S}_x$  and  $\mathcal{L}_x$  are intrinsic to the preference relation  $\preceq$ : they are defined purely from comparisons and do not require any numerical objective function.

**Lemma 2.4.** *Assume the preference relation  $\preceq$  is complete and transitive. Then for any  $x, y \in \mathcal{X}$ ,  $y \preceq x$  if and only if  $\mathcal{S}_y \subseteq \mathcal{S}_x$ . In particular,  $y \sim x$  if and only if  $\mathcal{S}_y = \mathcal{S}_x$  (equivalently,  $\mathcal{L}_y = \mathcal{L}_x$ ). Moreover, the set of optimal solutions satisfies  $\mathcal{X}^* = \bigcap_{x \in \mathcal{C}} (\mathcal{S}_x \cap \mathcal{C})$ .*

*Proof.* If  $y \preceq x$  and  $z \in \mathcal{S}_y$ , then  $z \preceq y$ ; by transitivity,  $z \preceq x$ , so  $z \in \mathcal{S}_x$  and  $\mathcal{S}_y \subseteq \mathcal{S}_x$ . Conversely, if  $\mathcal{S}_y \subseteq \mathcal{S}_x$ , then  $y \in \mathcal{S}_y$  (since  $y \preceq y$ ), hence  $y \in \mathcal{S}_x$  and  $y \preceq x$ . The equivalence for  $y \sim x$  follows by applying the first statement in both directions.

Finally,  $z \in \bigcap_{x \in \mathcal{C}} (\mathcal{S}_x \cap \mathcal{C})$  if and only if  $z \in \mathcal{C}$  and  $z \preceq x$  for all  $x \in \mathcal{C}$ , which is exactly the definition of  $z \in \mathcal{X}^*$  in (3).  $\square$

**Preference properties via sublevel-set geometry.** Below we introduce several geometric regularity properties of the preference relation  $\preceq$  through the sublevel sets  $\mathcal{S}_x$  and level sets  $\mathcal{L}_x$ .

**Definition 2.5** (Plateau-free and convex preferences). We say the preference relation  $\preceq$  is:

- *plateau-free* if for every  $x \in \mathcal{X} \setminus \mathcal{X}^*$ , ties occur only on the boundary of the sublevel set at  $x$ , i.e.,  $\partial \mathcal{S}_x = \mathcal{L}_x$ ;
- *convex* if for every  $x \in \mathcal{X}$ , the sublevel set  $\mathcal{S}_x$  is convex.

These properties are stated directly in terms of sublevel sets and level sets, which are essentially equivalent to the preference relation  $\preceq$  (Lemma 2.4) and do not require any numerical objective function. They also match common modeling assumptions in preference data: plateau-free preferences rule out regions of exact indifference with nontrivial volume (ties occur on a boundary rather than on a flat plateau), and convexity captures the idea that convex combinations of points that are no worse than  $x$  are still no worse than  $x$ . Furthermore, with plateau-freeness, the level set  $\mathcal{L}_x$  is exactly the boundary of the sublevel set  $\mathcal{S}_x$ . As  $\mathcal{L}_x$  is contained within  $\mathcal{S}_x$  according to its definition, the sublevel set  $\mathcal{S}_x$  is closed. In the rest of the paper, we work primarily in the plateau-free setting. Convexity will only be invoked when deriving global guarantees for optimization algorithms.

**Regularity and normal direction.** Furthermore, we define the following regularity condition on the preference relation  $\preceq$ :

**Definition 2.6** (Regularity of preference relation). For  $x \in \partial\mathcal{S}_x$  and  $r \geq 0$ , the preference relation  $\preceq$  is  $r$ -regular at  $x$ , if there exists  $n \in \mathbb{S}^{d-1}$ , such that  $B(x - rn, r) \subseteq \mathcal{S}_x$  and  $B(x + rn, r) \subseteq \overline{\mathcal{X} \setminus \mathcal{S}_x}$ .

Here  $B(z, r)$  in 2.6 denotes the closed Euclidean ball centered at  $z$  with radius  $r$ , i.e.,  $B(z, r) := \{y \in \mathcal{X} : \|y - z\| \leq r\}$ . The condition in 2.6 says that, at scale  $r$ ,  $\mathcal{S}_x$  contains an interior ball of radius  $r$  tangent at  $x$  and, simultaneously, its complement contains an exterior ball of radius  $r$  tangent at  $x$  along direction  $n$ . The unit direction of  $n$ , denoted by  $n_x := \frac{n}{\|n\|}$  (which is equal to  $n$ ), is the unique outward unit normal direction to  $\partial\mathcal{S}_x$  at  $x$ . We call  $n_x$  the *normal vector* at  $x$  for short.

To quantify how “curved” the boundary  $\partial\mathcal{S}_x$  is near  $x$ , we introduce the *regularity radius*  $r_x$  of  $x$  and the curvature  $\kappa_x$  of  $x$ .

**Definition 2.7** (Regularity radius and curvature). Assume the preference relation  $\preceq$  is  $r$ -regular at  $x$  for  $r \geq 0$ , and let  $n_x$  be the outward unit normal direction to  $\partial\mathcal{S}_x$  at  $x$ . The regularity radius  $r_x$  of  $x$  is defined as follows:

$$r_x := \sup \left\{ r \geq 0 : B(x - rn_x, r) \subseteq \mathcal{S}_x \text{ and } B(x + rn_x, r) \subseteq \overline{\mathcal{X} \setminus \mathcal{S}_x} \right\} \in (0, \infty] , \quad (5)$$

and the curvature  $\kappa_x$  of  $x$  is defined as  $\kappa_x := \frac{1}{r_x}$ , with the convention  $1/\infty = 0$  and  $1/0 = \infty$ .

Thus  $r_x$  is the largest radius at which  $\partial\mathcal{S}_x$  can be locally supported by two tangent balls. Figure 2 shows an illustration of the normal direction  $n_x$  and the regularity radius  $r_x$ . Large  $r_x$  corresponds to a locally “flat” boundary, while small  $r_x$  corresponds to a highly curved boundary.

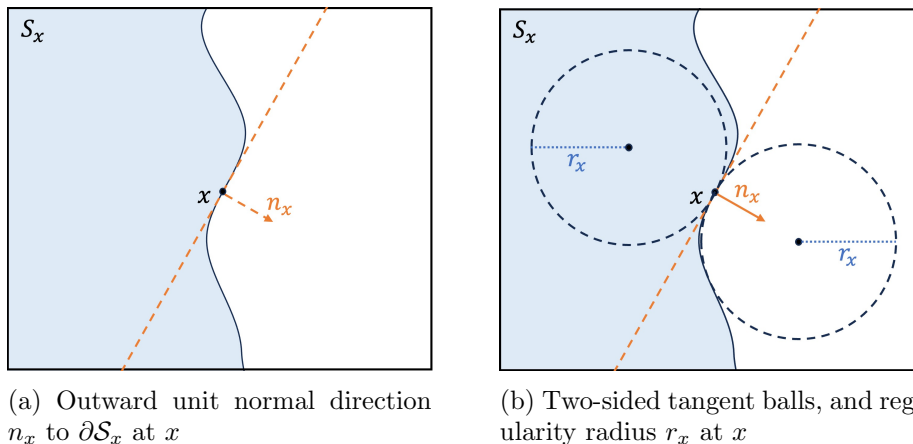


Figure 2: Illustration of the outward unit normal direction  $n_x$  and the regularity radius  $r_x$ .

It should be noted that, the regularity radius and curvature are local properties for  $x$ . Intuitively, the smaller the sublevel set  $\mathcal{S}_x$  is, the smaller the largest incirbing ball may be, and thus the smaller the regularity radius  $r_x$  and the larger the curvature  $\kappa_x$  for a boundary point  $x$  can achieve. Furthermore, when the preference is convex, the sublevel set  $\mathcal{S}_x$  is convex, and thus no matter how large  $r$  is,  $B(x + rn_x, r) \subseteq \overline{\mathcal{X} \setminus \mathcal{S}_x}$ , so the two-sided tangent-ball condition is equivalent to the “one-sided” tangent-ball condition  $r_x := \sup_r \{r \geq 0 : B(x - rn_x, r) \subseteq \mathcal{S}_x\}$ . In geometric terminology, radii defined through tangent balls are closely related to the notion of reach for sets with positive reach [7]. In that language,  $r_x$  plays the role of a local reach parameter.

## 2.3 Relation with comparison-based optimization

The framework above takes the preference relation  $\preceq$  (and the comparison oracle  $\mathcal{C}_{\text{cmp}}$ ) as the primitive description of the problem. Much of the comparison-based optimization literature, however, starts from an objective function  $f$  and then assumes that the algorithm can only access  $f$  through comparisons [45, 39, 16, 2]. This subsection clarifies the relationship: whenever an objective  $f$  realizes the preference relation, comparison-based optimization on  $f$  is exactly the same as function-free optimization under  $\preceq$ . Moreover, the function-free viewpoint shows that some objective functions are actually invariant: strictly increasing reparameterizations of  $f$  generate the same comparisons and hence the same optimization problem.

Let us say a function  $f : \mathcal{X} \rightarrow \mathbb{R}$  realizes the preference relation  $\preceq$  if for all  $x, y \in \mathcal{X}$ ,  $x \preceq y$  if and only if  $f(x) \leq f(y)$ . If such a representative function  $f$  exists, then the function-free optimization problem (2) is identical to the usual constrained minimization problem

$$\min_{x \in \mathcal{C}} f(x), \quad \text{and hence} \quad \mathcal{X}^* = \arg \min_{x \in \mathcal{C}} f(x).$$

Whenever  $f$  realizes  $\preceq$ , comparisons can be encoded numerically by

$$\mathcal{C}_f(x, y) := \text{sign}(f(y) - f(x)) \in \{-1, 0, +1\}. \quad (6)$$

In this case,  $\mathcal{C}_f$  is consistent with Definition 2.2: for all  $x, y \in \mathcal{X}$ ,  $\mathcal{C}_f(x, y) = -1$  iff  $y \prec x$ ,  $\mathcal{C}_f(x, y) = 0$  iff  $y \sim x$ , and  $\mathcal{C}_f(x, y) = +1$  iff  $x \prec y$ . Thus, one may view the oracle  $\mathcal{C}_{\text{cmp}}$  as being instantiated by  $\mathcal{C}_f$  when a representative function exists.

However, the same preference relation can have many different representative functions. We say two functions  $f, g : \mathcal{X} \rightarrow \mathbb{R}$  are preference-equivalent on  $\mathcal{X}$  if they realize the same preference relation:  $f(x) \leq f(y)$  if and only if  $g(x) \leq g(y)$  for all  $x, y \in \mathcal{X}$ . Equivalently,  $\mathcal{C}_f(x, y) = \mathcal{C}_g(x, y)$  for all  $x, y \in \mathcal{X}$ . The next lemma shows that preference-equivalence is exactly composition with a strictly increasing scalar map.

**Lemma 2.8.** *If  $f$  and  $g$  are preference-equivalent on  $\mathcal{X}$ , then there exists a unique strictly increasing map  $\phi : \text{range}(f) \rightarrow \text{range}(g)$  such that  $g = \phi \circ f$ . Conversely, for any strictly increasing  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ , the functions  $f$  and  $\phi \circ f$  are preference-equivalent on  $\mathcal{X}$ .*

*Proof.* Define  $\phi(t) := g(x)$  for any  $x \in \mathcal{X}$  with  $f(x) = t$ . This is well-defined because if  $f(x) = f(y)$ , then both  $f(x) \leq f(y)$  and  $f(y) \leq f(x)$  hold; by preference-equivalence,  $g(x) \leq g(y)$  and  $g(y) \leq g(x)$ , hence  $g(x) = g(y)$ .

If  $t_1 < t_2$  and  $f(x_i) = t_i$ , then  $f(x_1) \leq f(x_2)$  but not  $f(x_2) \leq f(x_1)$ . By preference-equivalence,  $g(x_1) \leq g(x_2)$  but not  $g(x_2) \leq g(x_1)$ , hence  $g(x_1) < g(x_2)$ . Therefore  $\phi(t_1) < \phi(t_2)$  and  $\phi$  is strictly increasing. Uniqueness is immediate from  $g = \phi \circ f$ . The converse holds because strictly increasing maps preserve order.  $\square$

The three functions in Figure 1 are preference-equivalent. Although looking very different, they induce the same preference relation and hence the same optimization problem from the perspective of comparisons.

**Proposition 2.9** (Invariance under preference-equivalent representative functions). *If two objective functions  $f$  and  $g$  induce the same preference relation, then any comparison-based algorithm will behave in exactly the same way on  $f$  and on  $g$ : it will receive the same oracle replies, generate the same sequence of queried points and iterates (up to its own randomness).*

Therefore, for these algorithms that depends only on comparison information, no matter deterministic or randomized, any guarantee can only depend on the induced preference relation (and the sets  $\mathcal{S}_x, \mathcal{L}_x$ ), not on the numerical scaling of a particular representative. For example, the smoothness and convexity of the objective functions rely on the particular numerical scaling, but the smoothness and convexity of the preference relation are invariant under strictly increasing reparameterizations.

It should be noted that the comparison oracles considered in this paper (Definition 2.2 and (6)) always return the correct comparison result. Exact comparison is common in comparison-based derivative-free optimization [13, 27, 9, 2]. In addition, sometime people assume the comparison oracle may make errors with some probability [14, 35, 36]. In those cases, repeating comparing the same pair and taking a majority vote can generate the correct comparison result with high probability [14]. If the probability of returning the right answer is at least  $\frac{1}{2} + \nu$  for some  $\nu > 0$ , then [35] shows that there exists an adaptive sampling strategy so that with fixed given  $\delta \in (0, 1)$ , it does not need prior knowledge of  $\nu$  and can recover the correct comparison result with failure probability at most  $\delta$  using at most  $O(\nu^{-2} \log(1/\delta))$  repeated comparisons.

**Relation between regularity radius and smoothness.** Although there might exist multiple functions realizing a preference relation, if one of them is smooth, then a relation between the regularity radius and the gradient norm can be established, as shown in the next lemma.

**Lemma 2.10.** *Suppose the preference relation  $\preceq$  is realized by a differentiable function  $f : \mathcal{X} \rightarrow \mathbb{R}$ , and assume  $f$  is  $L$ -smooth, i.e.,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathcal{X}. \quad (7)$$

Fix any  $x \in \mathcal{X}$  with  $\nabla f(x) \neq 0$ , then the normal direction  $n_x$  at  $x$  is given by  $\hat{n} := \frac{\nabla f(x)}{\|\nabla f(x)\|}$ . The regularity radius  $r_x$  and curvature  $\kappa_x$  satisfy

$$r_x \geq \frac{\|\nabla f(x)\|}{L}, \quad \text{and hence} \quad \kappa_x \leq \frac{L}{\|\nabla f(x)\|}. \quad (8)$$

*Proof.* If  $L = 0$ , then  $\nabla f$  is constant and the induced preference sublevel sets are halfspaces. In this case  $r_x = \infty$  and  $\kappa_x = 0$ , so (8) holds trivially. Assume below that  $L > 0$  and set  $r := \frac{\|\nabla f(x)\|}{L}$ . We will then show there exists two-sided tangent balls,  $B(x - r\hat{n}, r)$  and  $B(x + r\hat{n}, r)$ , of radius  $r$  at  $x$  along direction  $\hat{n}$ , which implies  $\hat{n}$  is the normal direction,  $r_x \geq r$  and hence (8).

Take any  $y \in B(x - r\hat{n}, r)$ . Expanding  $\|y - (x - r\hat{n})\|^2 \leq r^2$  gives  $\|y - x\|^2 + 2r\langle \hat{n}, y - x \rangle \leq 0$ , so  $\langle \hat{n}, y - x \rangle \leq -\frac{\|y - x\|^2}{2r}$ . Multiplying by  $\|\nabla f(x)\|$  yields

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

Note that smoothness (7) implies that  $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ , so  $f(y) \leq f(x)$ . Therefore,  $B(x - r\hat{n}, r) \subseteq \mathcal{S}_x$ .

Similarly, for any  $y \in B(x + r\hat{n}, r)$ , expanding  $\|y - (x + r\hat{n})\|^2 \leq r^2$  gives  $\|y - x\|^2 - 2r\langle \hat{n}, y - x \rangle \leq 0$ , so  $\langle \hat{n}, y - x \rangle \geq \frac{\|y - x\|^2}{2r}$ . Therefore

$$\langle \nabla f(x), y - x \rangle = \|\nabla f(x)\| \langle \hat{n}, y - x \rangle \geq \frac{\|\nabla f(x)\|}{2r} \|y - x\|^2 = \frac{L}{2} \|y - x\|^2.$$

Similarly, the smoothness condition (7) implies that  $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle - \frac{L}{2} \|y - x\|^2$ , so  $f(y) \geq f(x)$ . Therefore,  $B(x + r\hat{n}, r) \subseteq \mathcal{X} \setminus \mathcal{S}_x$ .

By Definitions 2.6 and 2.7,  $\hat{n}$  is the normal direction  $n_x$  at  $x$ , and  $r_x \geq r$  and  $\kappa_x \leq 1/r$ .  $\square$

## 2.4 Optimality and stationarity criteria

**Level-set optimality gap.** In function-based optimization, the quality of a solution is often measured by function suboptimality (e.g.,  $f(x) - f^*$ ). In function-free optimization, we do not have access to numerical objective values, so we instead use comparison-invariant geometric measures. Two examples are the distance to optimal solutions,  $\text{dist}(x, \mathcal{X}^*)$ , and the level-set optimality gap,  $\Delta_{\text{LS}}(x)$ , defined as

$$\Delta_{\text{LS}}(x) := \text{dist}(\mathcal{X}^*, \mathcal{L}_x).$$

Since  $x \in \mathcal{L}_x$ , we always have  $\Delta_{\text{LS}}(x) \leq \text{dist}(x, \mathcal{X}^*)$ , so  $\Delta_{\text{LS}}(x)$  is never larger than the point-to-optimal-set distance and can therefore be easier to satisfy. Moreover, because points in the same level set  $\mathcal{L}_x$  are regarded as indifferent under the preference relation, if  $\Delta_{\text{LS}}(x)$  is small, then there exists a point that is indifferent to  $x$  and close to the optimal set. In this sense,  $x$  is as good as a point near the optimal set from the perspective of the preference relation. Therefore,  $\Delta_{\text{LS}}(x)$  is more intrinsic to the preference relation  $\preceq$  than  $\text{dist}(x, \mathcal{X}^*)$ . For this reason, most of our optimization results are stated in terms of  $\Delta_{\text{LS}}(\cdot)$ .

**Regularity radius.** Although not always reliable, the regularity radius  $r_x$  can also serve as a measure of near-stationarity in some cases. The regularity radius  $r_x$  is also intrinsic to the preference relation. At a high level, small  $r_x$  indicates that the local geometry of the sublevel set near  $x$  is sharp, so small perturbations may lead to noticeable changes in the local ordering. This is consistent with the intuition that, in a near-optimal or near-stationary region, a small perturbation of the underlying preference may flip the local ordering of two points.

If a smooth function realizes the preference relation, then small regularity radius implies small gradient norm, which is widely regarded as a measure of near-stationarity. Indeed, Lemma 2.10 shows that if  $\preceq$  is realized by an  $L$ -smooth differentiable function  $f$ , then  $r_x \geq \frac{\|\nabla f(x)\|}{L}$ , and hence  $\|\nabla f(x)\| \leq Lr_x$ . Moreover, small regularity radius is in general a stronger certificate than small gradient norm, since the converse need not hold. For  $f(t) = t^3$  on  $\mathbb{R}$ , the point  $t = 0$  is stationary, but the induced preference sublevel set  $\mathcal{S}_0$  is  $(-\infty, 0]$ , so  $r_0 = \infty$ . Furthermore, it will be shown in Section 4.2 that a local growth condition widely holds for regularity radius. For example, it holds when the preference relation is realized by several commonly used classes of functions, including convex quadratic functions, strongly convex functions, and smooth functions satisfying the PL inequality and quadratic growth. If such a local growth condition holds, then small regularity radius directly implies small level-set optimality gap  $\Delta_{\text{LS}}(\cdot)$ .

Throughout the paper we use  $\Delta_{\text{LS}}(\cdot)$  as the primary measure of near-optimality, while small  $r_x$  appears as a secondary certificate. This viewpoint is also reflected in our algorithmic guarantees: the method will either drive the level-set optimality gap to be small, or return a point with small regularity radius. Achieving the former usually requires higher comparison complexity, while achieving the latter typically only requires setting certain parameters to be sufficiently small.

## 3 Normal direction estimation from comparisons

This section studies how to estimate the normal direction  $n_x$  at a point  $x$  using only pairwise comparisons. Throughout this section we assume the preference relation is plateau-free (Definition 2.5) and  $r_x$ -regular at  $x$  (Definition 2.6) with regularity radius  $r_x > 0$ , so the normal direction  $n_x$  exists and is unique. We first propose a normal direction estimation method based on comparing  $x$  and  $x + hu$  with user-chosen  $h > 0$  and  $u \in \mathbb{S}^{d-1}$ , for  $x$  at which the preference relation is  $r_x$ -regular with regularity radius  $r_x > 0$ . In 3.3, assuming in addition that the sublevel set  $\mathcal{S}_x$  is convex,

we propose a certified comparison scheme using line search to update  $h$ , yielding a parameter-free normal direction estimator. Minimax lower bounds are given in Section 3.4.

### 3.1 Halfspace certificates of normal direction via comparisons

Before presenting the algorithms, we first show how to extract halfspace certificates of normal direction  $n_x$  from comparisons, which is the key building block for our normal direction estimation methods.

**Lemma 3.1.** *Fix  $x \in \mathcal{X} \setminus \mathcal{X}^*$  and assume the preference relation is plateau-free and  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ . Fix any  $h > 0$  and set  $\tau := \frac{h}{2r_x}$ . Then for every  $u \in \mathbb{S}^{d-1}$ ,*

- (i) *if  $x \preceq x + hu$ , then  $\langle n_x, u \rangle \geq -\tau$ ;*
- (ii) *if  $x + hu \preceq x$ , then  $\langle n_x, u \rangle \leq \tau$ .*

*Proof.* We prove both claims by contradiction.

(i) Assume  $\langle n_x, u \rangle < -\tau = -\frac{h}{2r_x}$ . Then

$$\|(x + hu) - (x - r_x n_x)\|^2 = \|hu + r_x n_x\|^2 = h^2 + r_x^2 + 2hr_x \langle u, n_x \rangle < r_x^2,$$

so  $x + hu \in \text{int} B(x - r_x n_x, r_x) \subseteq \text{int}(\mathcal{S}_x)$ . Since  $x \notin \mathcal{X}^*$ , plateau-freeness implies that interior points of  $\mathcal{S}_x$  cannot tie with  $x$ ; hence  $x + hu \prec x$ . In particular,  $x \preceq x + hu$  is false, proving (i).

(ii) Assume  $\langle n_x, u \rangle > \tau = \frac{h}{2r_x}$ . Then

$$\|(x + hu) - (x + r_x n_x)\|^2 = \|hu - r_x n_x\|^2 = h^2 + r_x^2 - 2hr_x \langle u, n_x \rangle < r_x^2,$$

so  $x + hu \in \text{int} B(x + r_x n_x, r_x) \subseteq \text{int}(\mathcal{S}_x^c)$ . Therefore  $x + hu \not\preceq x$ , proving (ii).  $\square$

Lemma 3.1 says that, the comparison between  $x$  and  $x + hu$  provides a halfspace certificate of the normal direction  $n_x$  with margin  $\tau = \frac{h}{2r_x}$ . The left two plots of Figure 3 shows an illustration. Intuitively, if  $x + hu \in \mathcal{S}_x$ , then the angle between  $u$  and  $n_x$  cannot be much smaller than  $90^\circ$ , and if  $x + hu \notin \mathcal{S}_x$ , then the angle between  $u$  and  $n_x$  cannot be much larger than  $90^\circ$ . The flatter the sublevel set is (the smaller  $h$  is and the larger  $r_x$  is), the more accurate the certificate is.

Furthermore, if assuming in addition that  $\mathcal{S}_x$  is convex, then the halfspace certificate can be strengthened with no margin.

**Lemma 3.2.** *Let the preference relation be  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ , and assume in addition that  $\mathcal{S}_x$  is convex. Then  $\mathcal{S}_x$  is contained in the supporting halfspace:  $\mathcal{S}_x \subseteq \{z \in \mathcal{X} : \langle n_x, z - x \rangle \leq 0\}$ . Furthermore, for any  $h > 0$  and  $u \in \mathbb{S}^{d-1}$ , if  $x + hu \preceq x$ , then  $\langle n_x, u \rangle \leq 0$ .*

*Proof.* Since  $\mathcal{S}_x$  is convex, its tangent halfspace at  $x$  is a supporting halfspace. Now fix  $h > 0$  and  $u \in \mathbb{S}^{d-1}$ . If  $x + hu \preceq x$ , then  $x + hu \in \mathcal{S}_x$ , and  $\mathcal{S}_x \subseteq \{z \in \mathcal{X} : \langle n_x, z - x \rangle \leq 0\}$  implies

$$\langle n_x, hu \rangle = \langle n_x, (x + hu) - x \rangle \leq 0,$$

which is the desired conclusion.  $\square$

In the convex setting, Lemma 3.2 provides an exact one-sided certificate that we will exploit in 3.3 by adaptively shrinking  $h$  until comparisons become fully reliable. The right plot of Figure 3 shows an illustration.

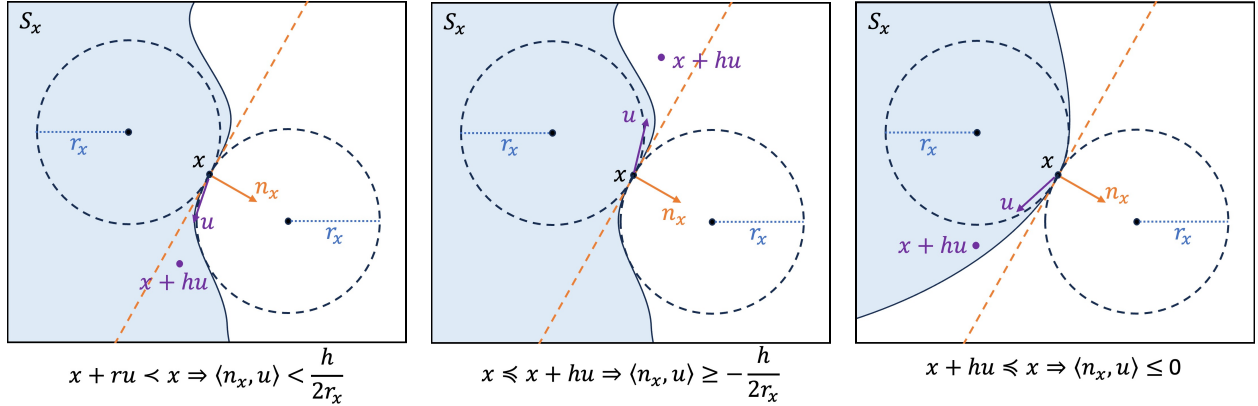


Figure 3: Illustration examples of Lemma 3.1 for nonconvex sublevel sets  $\mathcal{S}_x$  (left two plots) and Lemma 3.2 for convex sublevel sets  $\mathcal{S}_x$  (right one plot).

### 3.2 Normal direction estimation with fixed comparison radius

Our normal direction estimation method has a two-layer structure. The inner layer (Algorithm 1) performs a bisection search on the unit circle in a given two-dimensional plane, while the outer layer (Algorithm 2) uses this planar bisection search repeatedly to build an orthonormal family of  $d - 1$  approximately tangential directions. Rather than searching for  $n_x$  directly, we construct directions  $y_1, \dots, y_{d-1}$  satisfying  $|\langle n_x, y_i \rangle|$  small. The remaining one-dimensional orthogonal complement then approximates the normal direction line  $\{\pm n_x\}$ , and one final comparison fixes the outward orientation. In this subsection, we work with the scheme of fixed comparison radius  $h$ , while the adaptive comparison radius case will be presented in Section 3.3.

Algorithm 1 solves the planar subproblem in  $\text{span}\{a, b\}$  and returns an approximately tangential direction. Algorithm 2 calls this planar routine  $d - 1$  times in carefully chosen planes so that the resulting directions remain orthonormal. Theorem 3.5 then gives the overall estimation error and comparison complexity in terms of the comparison radius  $h$  and the planar depth  $T$ .

Algorithm 1 is essentially a bisection search on the unit circle in  $\text{span}\{a, b\}$ . Lines 6 and 9 initialize  $u_+$  and  $u_-$  to be two orthogonal directions that are separated by the normal direction  $n_x$  (i.e.,  $x + hu_- \preceq x \preceq x + hu_+$ ), and the loop performs a bisection search by comparing  $x$  with  $x + hu_M$ , where  $u_M$  is the normalized midpoint of  $u_+$  and  $u_-$ . The output is the last midpoint  $u_M$ . Lemma 3.3 below shows that  $u_M$  is an approximately tangential direction of  $n_x$ .

**Lemma 3.3.** *Let the preference relation be plateau-free and  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ , and  $d \geq 2$ . Run Algorithm 1 with fixed comparison radius  $h > 0$ , and let the algorithm return  $y$ . Then the output satisfies*

$$|\langle n_x, y \rangle| \leq \frac{h}{2r_x} + \frac{\pi}{2^{T+2}}. \quad (9)$$

*Proof.* Set  $\tau := h/(2r_x)$ . Recall that Lemma 3.1 shows for any  $u \in \mathbb{S}^{d-1}$ , if  $x + hu \preceq x$  then  $\langle n_x, u \rangle \leq \tau$ , and if  $x \preceq x + hu$  then  $\langle n_x, u \rangle \geq -\tau$ .

Algorithm 1 works in the plane  $\text{span}\{a, b\}$  (the random rotation only changes the in-plane basis). If the algorithm returns  $\bar{a}$ , then either (i)  $x \pm h\bar{a} \preceq x$ , or (ii)  $x \preceq x \pm h\bar{a}$ . In case (i), applying Lemma 3.1 to  $u = \bar{a}$  and  $u = -\bar{a}$  yields  $\langle n_x, \bar{a} \rangle \leq \tau$  and  $-\langle n_x, \bar{a} \rangle \leq \tau$ , hence  $|\langle n_x, \bar{a} \rangle| \leq \tau$ . Case (ii) is analogous. Thus the desired bound holds.

---

**Algorithm 1** Planar bisection for an approximately tangential direction
 

---

- 1: **Input:** base point  $x$ , comparison radius  $h > 0$ , orthonormal  $a, b \in \mathbb{S}^{d-1}$ , steps  $T \in \mathbb{N}_{++}$ .
  - 2: Sample  $\theta \sim \text{Unif}[0, 2\pi)$  and set  $\bar{a} \leftarrow \cos \theta a + \sin \theta b$ ,  $\bar{b} \leftarrow -\sin \theta a + \cos \theta b$ .
  - 3: If using fixed comparison radius, compare  $x$  with  $x + h\bar{a}$  and  $x - h\bar{a}$ .
  - 4: If using adaptive comparison radius, use Algorithm 3 to update  $h$ , and compare  $x$  with  $x + h\bar{a}$  and  $x - h\bar{a}$ .
  - 5: **If**  $(x + h\bar{a} \preceq x$  **and**  $x - h\bar{a} \preceq x)$  **or**  $(x \preceq x + h\bar{a}$  **and**  $x \preceq x - h\bar{a})$  **return**  $\bar{a}$ .
  - 6: **If**  $x + h\bar{a} \preceq x$  **set**  $(u_+, u_-) \leftarrow (-\bar{a}, \bar{a})$ ; **else set**  $(u_+, u_-) \leftarrow (\bar{a}, -\bar{a})$ .
  - 7: If using fixed comparison radius, compare  $x$  with  $x + h\bar{b}$ .
  - 8: If using adaptive comparison radius, use Algorithm 3 to update  $h$  and compare  $x$  with  $x + h\bar{b}$ .
  - 9: **If**  $x + h\bar{b} \preceq x$  **update**  $u_- \leftarrow \bar{b}$ ; **else update**  $u_+ \leftarrow \bar{b}$ .
  - 10: **for**  $t = 1$  **to**  $T$  **do**
  - 11:      $u_M \leftarrow \frac{u_+ + u_-}{\|u_+ + u_-\|}$ .
  - 12:     If using fixed comparison radius, compare  $x$  with  $x + hu_M$ .
  - 13:     If using adaptive comparison radius, use Algorithm 3 to update  $h$ , and compare  $x$  with  $x + hu_M$ .
  - 14:     **If**  $x + hu_M \preceq x$  **update**  $u_- \leftarrow u_M$ ; **else update**  $u_+ \leftarrow u_M$ .
  - 15: **end for**
  - 16: **return**  $u_M$ .
- 

Otherwise, the algorithm constructs a quadrant bracket  $(u_-, u_+)$  satisfying  $x + hu_- \preceq x$ ,  $x \preceq x + hu_+$ , and  $\angle(u_-, u_+) \leq \frac{\pi}{2}$ . Using Lemma 3.1,  $\langle n_x, u_- \rangle \leq \tau$ ,  $\langle n_x, u_+ \rangle \geq -\tau$ . Let  $I$  denote the shorter circular arc on  $\mathbb{S}^{d-1} \cap \text{span}\{a, b\}$  connecting  $u_-$  to  $u_+$ . Since map  $u \mapsto \langle n_x, u \rangle$  is continuous on  $I$ , we have  $u^* \in I$  and  $|\langle n_x, u^* \rangle| \leq \tau$ . At each bisection step, the algorithm sets  $u_M = \frac{u_- + u_+}{\|u_- + u_+\|}$ , which is the geodesic midpoint of  $I$ , so the bracket arc length halves. Moreover, updating  $u_-$  when  $x + hu_M \preceq x$  and updating  $u_+$  otherwise preserves the two implications  $x + hu_- \preceq x$  and  $x \preceq x + hu_+$ , and hence preserves the existence of some  $u^*$  in the current bracket arc with  $|\langle n_x, u^* \rangle| \leq \tau$ .

After  $T$  bisections, the bracket arc length is at most  $\frac{\pi}{2^{T+1}}$ , so the returned midpoint  $y$  satisfies  $\angle(y, u^*) \leq \frac{\pi}{2^{T+2}}$  for some  $u^*$  in the final arc with  $|\langle n_x, u^* \rangle| \leq \tau$ . Finally, since  $\|y\| = \|u^*\| = 1$  and  $\|y - u^*\| \leq \angle(y, u^*)$ , we obtain

$$|\langle n_x, y \rangle| \leq |\langle n_x, u^* \rangle| + |\langle n_x, y - u^* \rangle| \leq \tau + \|y - u^*\| \leq \tau + \angle(y, u^*) \leq \tau + \frac{\pi}{2^{T+2}},$$

which is (9). □

We now lift the planar bisection method to  $d$  dimensions by repeatedly computing orthogonal directions  $y_i$  of  $n_x$  in successive two-dimensional planes. Once such a direction is computed, it is then rotated in a 2-dimensional subspace to ensure orthonormality, so that the final normal direction estimate can be read off without solving any linear system.

**Lemma 3.4.** *In Algorithm 2, for each  $i \in \{1, \dots, d-1\}$  the pair  $(v_i, q_{i+1})$  passed to Algorithm 1 is orthonormal, and  $y_1, \dots, y_{i-1}, v_i$  are orthonormal.*

*Proof.* We prove by induction that for each  $i \in \{1, \dots, d-1\}$  the set  $\{y_1, \dots, y_{i-1}, v_i\}$  is orthonormal and satisfies

$$\text{span}\{y_1, \dots, y_{i-1}, v_i\} \subseteq \text{span}\{q_1, \dots, q_i\}.$$

For  $i = 1$ , this holds since  $v_1 = q_1$ . Let  $y_i$  be the unit vector returned by that planar call and define  $c_i := \langle y_i, v_i \rangle$  and  $s_i := \langle y_i, q_{i+1} \rangle$  so that  $y_i = c_i v_i + s_i q_{i+1}$  with  $c_i^2 + s_i^2 = 1$ . The update sets

---

**Algorithm 2** Normal direction estimation via orthogonal-subspace construction
 

---

- 1: Input: point  $x$ , dimension  $d$ , comparison radius  $h > 0$ , planar bisection steps  $T \in \mathbb{N}_{++}$ .
  - 2: Sample a Haar-uniform orthonormal frame  $(q_1, \dots, q_d)$  in  $\mathcal{X}$ .
  - 3: Set  $v_1 \leftarrow q_1$ .
  - 4: **for**  $i = 1$  **to**  $d - 1$  **do**
  - 5: Run Algorithm 1 at  $x$  with  $(a, b) = (v_i, q_{i+1})$  and  $T$  steps, and obtain  $y_i$ .
  - 6: Set  $v_{i+1} \leftarrow -\langle y_i, q_{i+1} \rangle v_i + \langle y_i, v_i \rangle q_{i+1}$ .
  - 7: **end for**
  - 8: If using fixed comparison radius, compare  $x$  with  $x + hv_d$ .
  - 9: If using adaptive comparison radius, use Algorithm 3 to update  $h$ , and compare  $x$  with  $x + hv_d$ .
  - 10: **If**  $x + hv_d \prec x$  **return**  $\hat{n} := -v_d$  **else return**  $\hat{n} := v_d$ .
- 

$v_{i+1} := -s_i v_i + c_i q_{i+1}$ , which is unit and satisfies  $\langle v_{i+1}, y_i \rangle = 0$ . Moreover,  $v_{i+1}$  is orthogonal to all previous  $y_j$  because it lies in  $\text{span}\{v_i, q_{i+1}\}$  and both  $v_i$  and  $q_{i+1}$  are orthogonal to  $\{y_1, \dots, y_{i-1}\}$ . Thus  $\{y_1, \dots, y_i, v_{i+1}\}$  is orthonormal. Finally, since both  $v_i$  and  $q_{i+1}$  lie in  $\text{span}\{q_1, \dots, q_{i+1}\}$ , so does  $v_{i+1}$ , completing the induction. Taking  $i = d - 1$  shows that  $\{y_1, \dots, y_{d-1}, v_d\}$  is an orthonormal basis of  $\mathcal{X}$ .

Since  $q_{i+1}$  is orthogonal to  $\text{span}\{q_1, \dots, q_i\}$ , it is also orthogonal to  $v_i$  and all previous  $y_j$ . Because  $\|q_{i+1}\| = \|v_i\| = 1$ , the pair  $(v_i, q_{i+1})$  is orthonormal.  $\square$

Now we are ready for showing the accuracy guarantee and the number of pairwise comparisons used by Algorithm 2.

**Theorem 3.5** (Normal direction estimation). *Assume the preference relation is plateau-free and  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ , and  $d \geq 2$ . In the regime of fixed comparison radius  $h > 0$ , the output  $\hat{n}$  of Algorithm 2 satisfies*

$$\|n_x - \hat{n}\| \leq 2\sqrt{d-1} \left( \frac{h}{r_x} + \frac{\pi}{2^{T+1}} \right). \quad (10)$$

and Algorithm 2 uses at most  $(d-1)(T+3) + 1$  pairwise comparisons.

*Proof.* If the right-hand side of (10) is at least 2, then the claim is trivial because both  $n_x$  and  $\hat{n}$  are unit vectors. Hence later we only consider the case  $h < r_x$ .

Let  $\tau := h/(2r_x)$  and  $\gamma := \tau + \pi/2^{T+2}$ . Lemma 3.3 implies that for each  $i \in \{1, \dots, d-1\}$ ,  $|\langle n_x, y_i \rangle| \leq \gamma$  almost surely. Using Lemma 3.4, we can write  $n_x = \alpha_0 v_d + \sum_{i=1}^{d-1} \beta_i y_i$  where  $\alpha_0 := \langle n_x, v_d \rangle$  and  $\beta_i := \langle n_x, y_i \rangle$ . Then  $\sum_{i=1}^{d-1} \beta_i^2 \leq (d-1)\gamma^2$ .

If  $\sqrt{d-1}\gamma \geq \frac{1}{2}$ , then the right-hand side of (10) is at least 2, while  $\|n_x - \hat{n}\| \leq 2$  for any unit vectors; hence (10) holds trivially.

Assume now that  $\sqrt{d-1}\gamma < \frac{1}{2}$ . Then  $\sum_{i=1}^{d-1} \beta_i^2 < \frac{1}{4}$  and therefore  $|\alpha_0| = \sqrt{1 - \sum_{i=1}^{d-1} \beta_i^2} > \sqrt{3}/2$ . In particular  $|\alpha_0| > \tau$  since  $\tau \leq \gamma < \frac{1}{2}$ . We claim that the final orientation step returns  $\hat{n} = \text{sign}(\alpha_0) v_d$ . Indeed, if  $\alpha_0 > \tau$  then  $x + hv_d \preceq x$  cannot occur (by Lemma 3.1(ii)), so the algorithm cannot output  $-v_d$ . Similarly, if  $\alpha_0 < -\tau$  then  $x \preceq x + hv_d$  cannot occur (by Lemma 3.1(i)), so the algorithm cannot output  $v_d$ . Thus  $\hat{n} = \text{sign}(\alpha_0) v_d$  and  $\langle n_x, \hat{n} \rangle = |\alpha_0|$ . Finally,

$$\|n_x - \hat{n}\|^2 = (1 - |\alpha_0|)^2 + \sum_{i=1}^{d-1} \beta_i^2 \leq 2 \sum_{i=1}^{d-1} \beta_i^2 \leq 2(d-1)\gamma^2,$$

so  $\|n_x - \hat{n}\| \leq 2\sqrt{d-1}\gamma \leq 4\sqrt{d-1}\gamma$ , proving (10).

Finally, each call to Algorithm 1 uses at most  $T+3$  comparisons, and the final orientation step uses one more. Overall, at most  $(d-1)(T+3)+1$  comparisons are used.  $\square$

The bound (10) separates the two sources of error:  $2\sqrt{d-1}\frac{h}{r_x}$  is the local-geometry-induced bias from finite-radius comparisons (Lemma 3.1), and  $2\sqrt{d-1}\frac{\pi}{2^{T+1}}$  is the algorithmic approximation error from terminating planar bisection after  $T$  steps. Intuitively, the smaller  $r_x$  is, the faster the local geometry can change, leading to a larger bias term in the error.

**Corollary 3.6.** *Under the assumptions of Theorem 3.5, for any  $\epsilon \in (0, 2]$ , let  $h \in (0, \frac{\epsilon r_x}{8\sqrt{d-1}}]$ , and  $T = \lceil \log_2\left(\frac{8\sqrt{d-1}\pi}{\epsilon}\right) \rceil - 2$ . Then  $\|n_x - \hat{n}\| \leq \epsilon$ , and Algorithm 2 uses at most*

$$N_\epsilon = (d-1) \left\lceil \log_2\left(\frac{51\sqrt{d-1}}{\epsilon}\right) \right\rceil + 1 \quad (11)$$

*pairwise comparisons.*

*Proof.* Substituting  $h \leq \frac{\epsilon r_x}{8\sqrt{d-1}}$  and  $T = \lceil \log_2\left(\frac{8\sqrt{d-1}\pi}{\epsilon}\right) \rceil - 2$  into (10) yields  $\|n_x - \hat{n}\| \leq \epsilon$ . Note that for  $\epsilon \in (0, 2]$ ,  $h > 0$  and  $T \in \mathbb{N}_{++}$  so they are valid parameters. The number of comparisons is at most

$$(d-1)(T+3)+1 = (d-1) \left( \left\lceil \log_2\left(\frac{8\sqrt{d-1}\pi}{\epsilon}\right) \right\rceil + 1 \right) + 1 \leq (d-1) \left\lceil \log_2\left(\frac{51\sqrt{d-1}}{\epsilon}\right) \right\rceil + 1,$$

where the last inequality uses  $\log_2(8\pi) + 1 = \log_2(16\pi) < \log_2(51)$ . This finishes the proof.  $\square$

Corollary 3.6 shows that, once the fixed comparison radius satisfies  $h \lesssim \epsilon r_x / \sqrt{d}$ , Algorithm 2 estimates the normal direction to accuracy  $\epsilon$  using  $O(d \log(d/\epsilon))$  comparisons. In Section 3.4, we will prove a minimax lower bound of order  $\Omega(d \log(1/\epsilon))$  for normal direction estimation. Hence the above complexity is nearly optimal. Making the range of proper comparison radii as large as possible is also preferable, as it would be easier to choose a suitable  $h$  in practice, and comparisons between very close points are less robust practically.

It should also be noted that the case  $d=1$  should be separately analyzed, because Algorithm 2 reduces to a single comparison between  $x$  and  $x + hv_1$ , where  $v_1 \in \mathbb{S}^0$ , and Algorithm 1 is never used.

**Proposition 3.7.** *When  $d=1$ , let  $v_1 \in \mathbb{S}^0$  be the unit direction used by Algorithm 2. The algorithm reduces to a single comparison between  $x$  and  $x + hv_1$ . Still, suppose the preference relation is plateau-free and  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ . The output  $\hat{n}$  is  $-v_1$  if  $x + hv_1 \prec x$  and  $v_1$  otherwise. In this case,  $\hat{n} = n_x$  if  $h < 2r_x$ . The whole procedure uses only one comparison.*

*Proof.* In one dimension,  $\mathbb{S}^0 = \{\pm e\}$ , so  $n_x, \hat{n} \in \{\pm e\}$ . Applying Lemma 3.1 with  $u = e$  and  $u = -e$  shows that if  $h < 2r_x$ , then  $x + he \preceq x$  implies  $\langle n_x, e \rangle \leq \frac{h}{2r_x} < 1$ , so  $n_x = -e$ ; and  $x \prec x + he$  implies  $\langle n_x, e \rangle \geq -\frac{h}{2r_x} > -1$ , so  $n_x = e$ .  $\square$

When a smooth function  $f$  realizes the preference relation, the normal direction coincides with the normalized gradient. There are existing works in estimating the normalized gradient using comparison oracles. The most common approach is to use the random-direction sign estimator of the form  $\mathcal{C}_f(x + hu, x)u$  where  $\mathcal{C}_f$  is the comparison oracle of  $f$  defined in (6). This estimator's

expectation is proportional to the normalized gradient when  $h$  is sufficiently small. See, for example, [35, 36, 33, 3], for this idea and its usage in sparse problems, Riemannian optimization and noisy oracle settings. However, the cost of simply taking the average of repetitions increases polynomially with  $d/\epsilon$  for a target estimation error  $\epsilon$ . A second approach, proposed in [16], estimates the normalized gradient by maintaining a cone of plausible directions and shrinking this cone using half-space cuts extracted from comparison feedback. However, the cost grows quadratically with the dimension  $d$  (see the proof of Theorems 1 and 2 in [16]).

A third approach, studied in [45], considers  $L$ -smooth functions and proposes a two-stage method: first finding a dominant gradient component that is the largest in magnitude, and then estimating the ratio of this component to each of the other components of the gradient vector. After translating its notation to ours, its method gets an  $\epsilon$ -accurate normalized gradient using  $O(d \log(d/\epsilon))$  comparisons, but requires the comparison radius to satisfy  $h \lesssim \epsilon \|\nabla f(x)\| / (Ld^{3/2})$ . Since Lemma 2.10 gives  $r_x \geq \|\nabla f(x)\| / L$ , this means  $h \lesssim \epsilon r_x / d^{3/2}$ , which is more restrictive by an additional factor of  $d$  than the admissible scale in Corollary 3.6. When preparing the first manuscript of this work, we realized a recent concurrent work [39] proposes to first rotate the basis towards a reference direction (which is similar to the estimator of the first approach) and then continue with [45]’s approach. This improves the required comparison radius to order  $\epsilon \|\nabla f(x)\| / (L\sqrt{d})$ , but the guarantee is only with constant success probability because the reference-direction construction is randomized.

All of the above normalized gradient estimation methods assume the existence of an underlying smooth objective function and require function-dependent quantities such as a smoothness constant  $L$  and a lower bound on  $\|\nabla f(x)\|$ . Our estimator is formulated directly in the function-free preference geometry. The next subsection addresses the remaining drawback of the need to pre-tune  $h$  for convex preference by introducing an adaptive strategy.

### 3.3 Normal direction estimation with adaptive comparison radius

In the normal direction estimation method in Section 3.2, the error bound of Theorem 3.5 contains a bias term of the form  $2\sqrt{d-1} \frac{h}{r_x}$ , in which  $r_x$  is the regularity radius of  $x$ . The regularity radius may vary substantially from point to point. As discussed in Section 2.4 (and later in Section 4.2), a small regularity radius may itself signal proximity to an optimal or near-stationary region. Consequently, when normal direction estimation is used inside an optimization algorithm, it may be unrealistic to fix a single comparison radius along the entire trajectory.

Therefore, in this subsection we develop a normal direction estimation method that can automatically adjust the comparison radius using line search. According to Lemma 3.2, if additionally assuming convexity of the current sublevel set  $\mathcal{S}_x$ , we may decrease the comparison radius until comparisons yield an exact halfspace certificate of  $n_x$ . With exact halfspace certificates, the normal direction estimation error can be controlled purely by the bisection depth  $T$ , even when  $r_x$  is unknown. The price of line search appears only through a logarithmic dependence on  $1/r_x$ .

The method follows the same outer orthogonal-subspace construction (Algorithm 2) and the same planar bisection method (Algorithm 1), but we implement their comparison steps through an additional subroutine (Algorithm 3) to potentially adjust the comparison radius  $h$ . The comparison radius  $h$  is stored in the global memory and shared across all calls to Algorithm 2, Algorithm 1, and Algorithm 3, and it is initialized once to  $h_0$  and may be decreased over time.

Before presenting the main theorem, we introduce three useful lemmas. The first lemma quantifies how many radius halvings are needed before termination of Algorithm 3.

**Lemma 3.8.** *Assume the preference relation is plateau-free and  $r_x$ -regular at  $x$  with  $r_x > 0$ . Fix  $u \in \mathbb{S}^{d-1}$  with  $\langle n_x, u \rangle \neq 0$ . Consider one call of Algorithm 3 entered with comparison radius  $h_{\text{in}} > 0$ .*

---

**Algorithm 3** Comparison with line search
 

---

- 1: Input: base point  $x$ , direction  $u \in \mathbb{S}^{d-1}$ , and shared comparison radius  $h$  initialized once to  $h_0$ .
  - 2: **while**  $x \prec x + hu$  **and**  $x \prec x - hu$  **do**
  - 3:     Halve the comparison radius  $h \leftarrow h/2$
  - 4: **end while**
  - 5: **Return** the updated  $h$  and whether  $x \prec x + hu$  and  $x \prec x - hu$ .
- 

Then the while-loop terminates after at most

$$\left\lceil \log_2 \left( \frac{h_{\text{in}}}{r_x |\langle n_x, u \rangle|} \right) \right\rceil_+ \quad (12)$$

halvings.

*Proof.* Let  $t_\star := \left\lceil \log_2 \left( \frac{h_{\text{in}}}{r_x |\langle n_x, u \rangle|} \right) \right\rceil_+$ , and  $h_\star := \frac{h_{\text{in}}}{2^{t_\star}}$ . Then  $h_\star \leq r_x |\langle n_x, u \rangle|$ , hence  $h_\star \leq r_x$ . Define the inward direction  $u_{\text{in}} := -\text{sign}(\langle n_x, u \rangle)u$  so that  $\langle n_x, u_{\text{in}} \rangle = -|\langle n_x, u \rangle|$ . Since  $h_\star \leq r_x |\langle n_x, u \rangle|$  and  $|\langle n_x, u \rangle| > 0$ , we have  $\langle n_x, u_{\text{in}} \rangle = -|\langle n_x, u \rangle| < -\frac{h_\star}{2r_x}$ . Therefore,

$$\|(x + h_\star u_{\text{in}}) - (x - r_x n_x)\|^2 = \|h_\star u_{\text{in}} + r_x n_x\|^2 = h_\star^2 + r_x^2 + 2h_\star r_x \langle u_{\text{in}}, n_x \rangle < r_x^2,$$

so  $x + h_\star u_{\text{in}} \in \text{int} B(x - r_x n_x, r_x) \subseteq \text{int}(\mathcal{S}_x)$ . By plateau-freeness, interior points cannot tie with  $x$ , hence  $x + h_\star u_{\text{in}} \prec x$ .

In particular,  $x \prec x + h_\star u_{\text{in}}$  is false, so at radius  $h_\star$  the while-condition  $x \prec x + hu$  and  $x \prec x - hu$  cannot hold. Thus the loop terminates after at most  $t_\star$  halvings, which yields (12).  $\square$

The second lemma is an anti-concentration bound for Haar directions which will be used to control how many halving steps are needed with high probability.

**Lemma 3.9.** *Let  $q$  be uniform on  $\mathbb{S}^{d-1}$  and fix any  $v \in \mathbb{S}^{d-1}$ . Then for every  $\rho \in [0, 1]$ ,*

$$\mathbb{P}(|\langle v, q \rangle| \leq \rho) \leq \sqrt{\frac{2d}{\pi}} \rho.$$

This result could be derived by a corollary of the cosine-measure bound of independent random vectors uniformly distributed on the unit sphere, as stated in Lemma B.2 of [11].

*Proof.* The cosine measure bound in Lemma B.2 of [11] (letting  $D = \{q\}$  and  $m = 1$ ) states that for any  $\tau \in [0, \sqrt{d}]$ , the cosine measure  $\text{cm}(D, v)$  (which essentially is  $\langle v, q \rangle$ ) satisfies

$$\mathbb{P}\left(\langle v, q \rangle \geq \frac{\tau}{\sqrt{d}}\right) \geq 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\pi}}\right) = \frac{1}{2} - \frac{\tau}{\sqrt{2\pi}}.$$

Letting  $\tau = \rho\sqrt{d}$  yields  $\mathbb{P}(\langle v, q \rangle \geq \rho) \geq \frac{1}{2} - \rho\sqrt{\frac{d}{2\pi}}$  for  $\rho \in [0, 1]$ . By symmetry  $\langle v, q \rangle$  has the same distribution as  $-\langle v, q \rangle$ , hence  $\mathbb{P}(\langle v, q \rangle \leq -\rho) = \mathbb{P}(\langle v, q \rangle \geq \rho)$ . Therefore,

$$\mathbb{P}(|\langle v, q \rangle| \leq \rho) = 1 - \mathbb{P}(\langle v, q \rangle \geq \rho) - \mathbb{P}(\langle v, q \rangle \leq -\rho) \leq 1 - 2 \left(\frac{1}{2} - \rho\sqrt{\frac{d}{2\pi}}\right) = \sqrt{\frac{2d}{\pi}} \rho.$$

This finishes the proof.  $\square$

The third lemma states that once every queried direction comes with the correct sign of  $\langle n_x, u \rangle$ , planar bisection becomes exact and the local-geometry-induced bias term disappears from the planar error bound (compared with Lemma 3.3).

**Lemma 3.10.** *Assume  $\mathcal{S}_x$  is closed and convex, and the preference relation is plateau-free and  $r_x$ -regular at  $x$  with  $r_x > 0$ . Run Algorithm 1 to depth  $T$ , implementing Algorithm 3 before all comparisons. In the event that Algorithm 3 always terminates, the returned unit vector  $y$  almost surely satisfies*

$$|\langle n_x, y \rangle| \leq \frac{\pi}{2^{T+2}}. \quad (13)$$

*Proof.* Fix any direction  $u \in \mathbb{S}^{d-1}$  queried by Algorithm 1. Since Algorithm 3 terminates, at the final radius  $h$  we have at least one of  $x + hu \preceq x$  or  $x - hu \preceq x$ . Using the comparison outcomes already produced in the final check of the while-condition in Algorithm 3, we obtain a one-sided sign certificate. If  $\langle n_x, u \rangle > 0$  then Lemma 3.2 implies  $x + hu \notin \mathcal{S}_x$  and hence  $x \prec x + hu$ . If  $\langle n_x, u \rangle < 0$ , then Lemma 3.2 applied to  $-u$  implies  $x - hu \notin \mathcal{S}_x$ , hence  $x \prec x - hu$ . Since the line-search loop has terminated, by completeness,  $x + hu \preceq x$ .

Consequently, Algorithm 1 performs an exact bisection on the sign of the continuous function  $u \mapsto \langle n_x, u \rangle$  restricted to the unit circle in the plane  $\text{span}\{a, b\}$ . If  $\text{proj}_{\text{span}\{a, b\}}(n_x) = 0$  (although later we will prove it is with zero probability), then  $\langle n_x, y \rangle = 0$  for every returned  $y$  and the claim is trivial. Otherwise, there exists a tangential direction  $u^* \in \text{span}\{a, b\} \cap \mathbb{S}^{d-1}$  with  $\langle n_x, u^* \rangle = 0$ .

The initialization step using the orthonormal pair  $(\bar{a}, \bar{b})$  produces a bracket  $(u_-, u_+)$  lying in a quadrant, hence  $\angle(u_-, u_+) \leq \pi/2$ , and let  $u^*$  lie on the shorter arc between them. Each bisection step replaces an endpoint by the geodesic midpoint, halving the bracket arc length, while  $u^*$ . After  $T$  bisections, the final bracket length is at most  $\frac{\pi}{2^{T+1}}$ , so the returned midpoint  $y$  satisfies  $\angle(y, u^*) \leq \frac{\pi}{2^{T+2}}$ . Finally,

$$|\langle n_x, y \rangle| = |\langle n_x, y - u^* \rangle| \leq \|y - u^*\| \leq \angle(y, u^*) \leq \frac{\pi}{2^{T+2}},$$

which is (13). □

With Lemmas 3.8, 3.9 and 3.10, now we are ready to state the main complexity bound result for normal direction estimation with adaptive comparison radius.

**Theorem 3.11** (Adaptive normal direction estimation). *Assume  $d \geq 2$  and fix  $x \in \mathcal{X}$ . Suppose the preference relation is plateau-free and  $r_x$ -regular at  $x$  with regularity radius  $r_x > 0$ , and assume moreover that  $\mathcal{S}_x$  is convex. Fix  $\epsilon \in (0, \sqrt{2}]$  and define the bisection depth*

$$T_\epsilon := \left\lceil \log_2 \left( \frac{\pi\sqrt{d-1}}{2\epsilon} \right) \right\rceil. \quad (14)$$

*Run the outer construction of Algorithm 2 with depth  $T_\epsilon$ , initializing the shared radius to  $h_0$ . In the regime with adaptive comparison radius, the algorithm terminates almost surely and returns  $\hat{n}$  satisfying  $\|\hat{n} - n_x\| \leq \epsilon$ . Moreover, for every  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ ,*

$$N_\epsilon \leq 2(d-1) \left\lceil \log_2 \left( \frac{7\sqrt{d-1}}{\epsilon} \right) \right\rceil + 2 \left\lceil \log_2 \left( \frac{h_0 d^3}{r_x \epsilon} \right) \right\rceil_+ + 16 + 4 \left\lceil \log_2 \left( \frac{1}{\delta} \right) \right\rceil. \quad (15)$$

*Proof.* Let  $T$  denote  $T_\epsilon$  in this proof. Recall that the comparison radius  $h$  is shared globally, initialized at  $h_0$ , and is updated only by Algorithm 3.

We first prove the accuracy guarantee  $\|\hat{n} - n_x\| \leq \epsilon$ . By Lemma 3.4, the vectors  $\{y_1, \dots, y_{d-1}, v_d\}$  form an orthonormal basis. Write

$$n_x = \alpha_0 v_d + \sum_{i=1}^{d-1} \beta_i y_i, \quad \text{where } \beta_i := \langle n_x, y_i \rangle.$$

Lemma 3.10 implies that each call of Algorithm 1 returns a unit vector  $y_i$  satisfying  $|\langle n_x, y_i \rangle| \leq \frac{\pi}{2^{T+2}}$ , so  $\sqrt{\sum_{i=1}^{d-1} \beta_i^2} \leq \sqrt{d-1} \cdot \frac{\pi}{2^{T+2}}$ . The last step of Algorithm 2 chooses  $\hat{n} \in \{\pm v_d\}$  so that  $\langle n_x, \hat{n} \rangle = |\alpha_0|$ . Therefore,

$$\|n_x - \hat{n}\| = \sqrt{(1 - |\alpha_0|)^2 + \|\beta\|^2} \leq \sqrt{2}\|\beta\| \leq \sqrt{2d-2} \cdot \frac{\pi}{2^{T+2}}.$$

By (14),  $2^T \geq \frac{\pi\sqrt{d-1}}{2\epsilon}$ , so the right-hand side is at most  $\epsilon$ . Furthermore, note that as  $\epsilon \in (0, \sqrt{2}]$  and  $d \geq 2$ ,  $\left\lceil \log_2\left(\frac{\pi\sqrt{d-1}}{2\epsilon}\right) \right\rceil \geq \left\lceil \log_2\left(\frac{\pi}{2\sqrt{2}}\right) \right\rceil \geq 1$ , so  $T_\epsilon$  defined in (14) is in  $\mathbb{N}_{++}$ , and the bisection depth is valid. This proves that  $\|n_x - \hat{n}\| \leq \epsilon$ .

With the accuracy guarantee established, the rest of the proof counts the number of comparisons. Algorithm 2 makes  $d-1$  calls of Algorithm 1 and one final orientation comparison, and each call of Algorithm 1 uses at most  $T_\epsilon + 3$  comparisons (in lines 4, 8 and 13). Note that these comparisons can be directly read off from the final while-condition evaluation of Algorithm 3 without additional comparisons. Therefore, later we consider only the number of calls to Algorithm 3 and the number of halvings inside it. For each call of Algorithm 1, there are at most  $T_\epsilon + 2$  calls of Algorithm 3. Overall, at most  $Q := (d-1)(T_\epsilon + 2) + 1$  calls of Algorithm 3 are used in Algorithm 2.

Then let  $H$  be the total number of assignments  $h \leftarrow h/2$  made across all calls. A call with  $H_{\text{call}}$  halvings checks the condition  $H_{\text{call}} + 1$  times, making  $2H_{\text{call}} + 2$  comparisons. Summing over calls gives

$$N_\epsilon \leq 2Q + 2H = 2(d-1)(T_\epsilon + 2) + 2 + 2H. \quad (16)$$

It remains to control  $H$ .

Let  $\mathcal{U}$  be the set of all directions  $u$  ever passed to Algorithm 3. Lemma 3.8 implies that

$$H \leq \left\lceil \log_2\left(\frac{h_0}{r_x \cdot \min_{u \in \mathcal{U}} |\langle n_x, u \rangle|}\right) \right\rceil_+. \quad (17)$$

Therefore, in order to bound  $H$ , we need to first lower bound  $\min_{u \in \mathcal{U}} |\langle n_x, u \rangle|$  with high probability.

Fix the  $i$ -th call of Algorithm 1, the corresponding plane is  $\text{span}\{v_i, q_{i+1}\}$ . After the random rotation in line 2 of Algorithm 1, the algorithm uses an orthonormal basis, denoted by  $(\bar{a}_i, \bar{b}_i)$ , of this plane. Consider the equally spaced grid on the unit circle in this plane:

$$\mathcal{G}_i := \left\{ \cos\left(\frac{k\pi}{2^{T+1}}\right) \bar{a}_i + \sin\left(\frac{k\pi}{2^{T+1}}\right) \bar{b}_i : k = 0, 1, \dots, 2^{T+2} - 1 \right\}. \quad (18)$$

Up to depth  $T$ , Algorithm 1 only queries endpoints and dyadic midpoints of a quadrant arc, hence every direction queried in Algorithm 1 lies in  $\mathcal{G}_i$ . Therefore

$$\min_{u \in \mathcal{U}} |\langle n_x, u \rangle| \geq \min_{i=1, \dots, d-1} \min_{u \in \mathcal{G}_i} |\langle n_x, u \rangle|. \quad (19)$$

Next we will derive the lower bound for each  $\min_{u \in \mathcal{G}_i} |\langle n_x, u \rangle|$ .

Let  $\hat{u}_i$  denote  $\text{proj}_{\text{span}\{v_i, q_{i+1}\}}(n_x)$  and let  $u_i^*$  be any unit direction in  $\text{span}\{v_i, q_{i+1}\}$  satisfying  $\langle n_x, u_i^* \rangle = 0$  (equivalently,  $u_i^*$  is orthogonal to  $\hat{u}_i$ ). Define the smallest angular separation between the line of  $u_i^*$  and the grid endpoints as follows:

$$\Delta_i := \min_{g \in \mathcal{G}_i} \min\{\angle(g, u_i^*), \angle(g, -u_i^*)\}. \quad (20)$$

Then for any queried direction  $u \in \mathcal{G}_i$ ,

$$\begin{aligned} |\langle n_x, u \rangle| &= |\langle \hat{u}_i, u \rangle + \langle n_x - \hat{u}_i, u \rangle| = |\langle \hat{u}_i, u \rangle| = \|\hat{u}_i\| \cdot \|u\| \cdot |\cos(\angle(\hat{u}_i, u))| \\ &= \|\hat{u}_i\| \cdot \|u\| \cdot |\sin(\angle(u_i^*, u))| \geq \|\hat{u}_i\| \cdot \|u\| \cdot \sin(\Delta_i) = \|\hat{u}_i\| \cdot \sin(\Delta_i) \end{aligned} \quad (21)$$

where the second equality is because  $n_x - \hat{u}_i$  is orthogonal to the plane of  $u$ , the fourth equality is because  $\hat{u}_i$ ,  $u_i^*$  and  $u$  are in the same plane and  $\langle u_i^*, \hat{u}_i \rangle = 0$ , and the inequality is because of the definition of  $\Delta_i$  in (20).

Furthermore, since  $\Delta_i \leq \pi/2$ ,  $\sin(\Delta_i) \geq \frac{2}{\pi}\Delta_i$ . In addition, since  $v_i$  and  $q_{i+1}$  are orthonormal,  $\hat{u}_i = \text{proj}_{\text{span}\{v_i, q_{i+1}\}}(n_x) = \langle n_x, v_i \rangle v_i + \langle n_x, q_{i+1} \rangle q_{i+1}$ , we have  $\|\hat{u}_i\| \geq |\langle n_x, q_{i+1} \rangle|$ . Therefore, substituting these two lower bounds back to (21) gives a lower bound for each  $\min_{u \in \mathcal{G}_i} |\langle n_x, u \rangle|$ :

$$\min_{u \in \mathcal{G}_i} |\langle n_x, u \rangle| \geq |\langle n_x, q_{i+1} \rangle| \cdot \frac{2}{\pi} \Delta_i \quad (22)$$

Finally, using (19) and take the minimum over  $i = 1, \dots, d-1$  yields

$$\min_{u \in \mathcal{U}} |\langle n_x, u \rangle| \geq \frac{2}{\pi} \underbrace{\left( \min_{2 \leq j \leq d} |\langle n_x, q_j \rangle| \right)}_{=: A} \underbrace{\left( \min_{1 \leq i \leq d-1} \Delta_i \right)}_{=: B}. \quad (23)$$

Then we use  $A$  and  $B$  to denote the two product terms in the right-hand side of (23) for brevity.

To further lower bound  $\min_{u \in \mathcal{U}} |\langle n_x, u \rangle|$ , we now bound  $A$  and  $B$  from below with high probability. By Lemma 3.9 and a union bound,

$$\mathbb{P}(A \leq \alpha) \leq (d-1) \sqrt{\frac{2d}{\pi}} \alpha. \quad (24)$$

For  $B$ , fix  $i$  and condition on the input orthonormal vectors  $v_i$  and  $q_{i+1}$ , then the random in-plane rotation is controlled by the random variable  $\theta \sim \text{Unif}[0, 2\pi)$ . That rotation makes  $\mathcal{G}_i$  a uniform random rotation of a fixed equally spaced grid of spacing  $\frac{\pi}{2^{T+1}}$ . Equivalently, the position of  $u_i^*$  (and  $-u_i^*$ ) inside its nearest length- $\frac{\pi}{2^{T+1}}$  interval is uniform on  $[0, \frac{\pi}{2^{T+1}})$ . Hence for any  $\beta \in [0, \frac{1}{2}]$ ,

$$\mathbb{P}\left(B \leq \beta \cdot \frac{\pi}{2^{T+1}}\right) \leq \sum_{i=1}^{d-1} \mathbb{P}\left(\Delta_i \leq \beta \cdot \frac{\pi}{2^{T+1}}\right) \leq \sum_{i=1}^{d-1} 2\beta = 2(d-1)\beta. \quad (25)$$

With (24) and (25), choose  $\alpha := \frac{\delta}{2(d-1)} \sqrt{\frac{\pi}{2d}}$  and  $\beta := \frac{\delta}{4(d-1)}$ , then by a union bound  $\mathbb{P}(A \geq \alpha, B \geq \beta \cdot \frac{\pi}{2^{T+1}}) \geq 1 - \delta$ , and on this event (23) gives

$$\min_{u \in \mathcal{U}} |\langle n_x, u \rangle| \geq \frac{2}{\pi} \cdot \frac{\delta^2}{8(d-1)^2} \sqrt{\frac{\pi}{2d}} \cdot \frac{\pi}{2^{T+1}}$$

Plugging this into (17) and using  $2^T \leq \frac{\pi\sqrt{d-1}}{\epsilon}$  and  $(d-1)^{5/2}\sqrt{d} \leq d^3$  (for  $d \geq 2$ ), we obtain

$$H \leq \left[ \log_2 \left( \frac{32\sqrt{2\pi} h_0 d^3}{r_x \epsilon^2} \right) \right]_+ \quad (26)$$

with probability at least  $1 - \delta$ .

Finally, substituting (26) and  $T_\epsilon = \left\lceil \log_2 \left( \frac{\pi\sqrt{d-1}}{2\epsilon} \right) \right\rceil$  (by (14)) into (16), we have

$$\begin{aligned} N_\epsilon &\leq 2(d-1) \left( \left\lceil \log_2 \left( \frac{\pi\sqrt{d-1}}{2\epsilon} \right) \right\rceil + 2 \right) + 2 + 2 \left\lceil \log_2 \left( \frac{32\sqrt{2\pi} h_0 d^3}{r_x \epsilon \delta^2} \right) \right\rceil_+ \\ &\leq 2(d-1) \left\lceil \log_2 \left( \frac{7\sqrt{d-1}}{\epsilon} \right) \right\rceil + 16 + 2 \left\lceil \log_2 \left( \frac{h_0 d^3}{r_x \epsilon \delta^2} \right) \right\rceil_+ \end{aligned} \quad (27)$$

where the second inequality is because  $4 \cdot \frac{\pi}{2} \leq 7$  and  $\log_2(32\sqrt{2\pi}) \approx 6.33 \leq 7$ . Using the ceiling inequality  $\left\lceil \log_2 \left( \frac{C}{\delta^2} \right) \right\rceil_+ \leq \lceil \log_2(C) \rceil_+ + 2 \left\lceil \log_2 \left( \frac{1}{\delta} \right) \right\rceil$ , with  $C = \frac{h_0 d^3}{r_x \epsilon}$ , we obtain (15).  $\square$

Theorem 3.11 shows that the estimator is adaptive to the unknown local geometry. For any target accuracy  $\epsilon$ , the procedure does not require any prior knowledge of the regularity radius  $r_x$  or the curvature  $\kappa_x$ . Instead, it starts from an arbitrary initial comparison radius  $h_0$  and automatically shrinks the radius until the comparisons become informative at the scale required by the local geometry at  $x$ .

The dependence in the complexity bound (15) is also transparent. A larger initial  $h_0$  may cause more line-search halving steps. Likewise, a smaller regularity radius  $r_x$  corresponds to sharper local geometry and hence requires more comparisons. The good news is that all of these dependencies on  $h_0$  and  $r_x$  are only logarithmic.

The randomness comes from the internal randomness of the estimator in generating the Haar directions and uniform distribution. Its effect is also logarithmic through the term  $4 \lceil \log_2(1/\delta) \rceil$ , which means the total number of comparisons has an exponentially decaying tail and a finite expectation of the same order as the bound in (15).

When the dimension  $d$  is large, the leading contribution in (15) is the term  $2(d-1) \left\lceil \log_2 \left( \frac{7\sqrt{d-1}}{\epsilon} \right) \right\rceil$ . In this case, the adaptive estimator is about a factor of two more expensive than the fixed-comparison-radius estimator, whose complexity is given by (11) in Corollary 3.6. This extra factor is the price of adaptivity: during line search, each check requires comparisons in both the  $+u$  and  $-u$  directions, whereas the fixed-radius scheme does not need two-sided comparisons.

Finally, it should also be noted that the case  $d = 1$  should be separately analyzed. In this case Algorithm 1 is never used. See the proposition below.

**Proposition 3.12.** *Assume  $d = 1$ , in the same setting as Theorem 3.11, the output  $\hat{n}$  is exactly  $n_x$ , and the algorithm terminates after at most  $2 + 2 \lceil \log_2(h_0/r_x) \rceil_+$  comparisons.*

*Proof.* In one dimension,  $v_1 \in \mathbb{S}^0$  and  $n_x \in \{\pm v_1\}$ , so  $\langle n_x, v_1 \rangle \in \{\pm 1\}$  and in particular  $|\langle n_x, v_1 \rangle| = 1$ . Since the while-condition of Algorithm 3 is false at termination either  $x + hv_1 \preceq x$  or  $x - hv_1 \preceq x$ .

If  $x + hv_1 \preceq x$ , then by Lemma 3.2 we have  $\langle n_x, v_1 \rangle \leq 0$ , which forces  $\langle n_x, v_1 \rangle = -1$ , i.e.  $n_x = -v_1$ . Otherwise  $x + hv_1 \not\preceq x$ , so necessarily  $x - hv_1 \preceq x$ . Applying Lemma 3.2 to  $-v_1$  yields  $\langle n_x, -v_1 \rangle \leq 0$ , i.e.  $\langle n_x, v_1 \rangle \geq 0$ . Thus  $\langle n_x, v_1 \rangle = 1$  and hence  $n_x = v_1$ . Therefore  $\hat{n} = n_x$ .

For the comparison count, Lemma 3.8 gives that a single call to Algorithm 3 performs at most  $\lceil \log_2(h_0/(r_x |\langle n_x, v_1 \rangle|)) \rceil_+ = \lceil \log_2(h_0/r_x) \rceil_+$  radius halvings. Each while-check uses two comparisons, so the total number of comparisons is at most  $2 + 2 \lceil \log_2(h_0/r_x) \rceil_+$ .  $\square$

### 3.4 Lower bound of normal direction estimation methods

This subsection shows that for the task of estimating the normal direction, no algorithm can guarantee worst-case error  $\epsilon$  within  $o(d \cdot \log(1/\epsilon))$  comparisons without additional assumptions.

Consequently, both normal direction estimation methods (with fixed or adaptive comparison radii) proposed in this section are nearly optimal.

We establish the bound on preference relations realized by linear functions  $\langle c, x \rangle$  for  $c \in \mathbb{S}^{d-1}$ . They are plateau-free, convex and regular at all  $x \in \mathcal{X}$ , and  $c$  is the normal direction. Theorem 3.14 below shows that in this class of relations, there always exist two different normal directions that are hard to distinguish with a limited number of comparisons.

**Definition 3.13** (Algorithms with  $t$  comparisons). An algorithm with  $t$  comparisons proceeds sequentially and has access only to its past comparison history and its own private randomness. We represent all private randomness by a random seed  $\xi$  sampled before the run. At round  $i = 1, \dots, t$ , after observing the previous comparison history

$$\mathcal{T}_{i-1} := ((p_1, q_1, \sigma_1), \dots, (p_{i-1}, q_{i-1}, \sigma_{i-1})),$$

where  $\sigma_j \in \{-1, 0, +1\}$  is the comparison oracle response to the pair  $(p_j, q_j)$ , the algorithm chooses the next pair  $(p_i, q_i) \in \mathcal{X} \times \mathcal{X}$  as a measurable function of  $(\mathcal{T}_{i-1}, \xi)$ . After  $t$  rounds, its output is a measurable function of  $(\mathcal{T}_t, \xi)$ .

**Theorem 3.14** (Lower bound of normal direction estimation). *Fix  $d \geq 2$  and  $t \in \mathbb{N}_{++}$ . For each  $c \in \mathbb{S}^{d-1}$ , let  $\preceq_c$  denote the preference relation realized by  $f_c(x) := \langle c, x \rangle$ . For any algorithm with  $t$  comparisons in the sense of Definition 3.13, and for every realization of its private random seed  $\xi$ , there exist  $c, \hat{c} \in \mathbb{S}^{d-1}$  such that*

$$\angle(c, \hat{c}) > 2^{-(t+1)/(d-1)}, \quad (28)$$

and, under the same  $\xi$ , the algorithm receives the same comparison transcript under  $\preceq_c$  and  $\preceq_{\hat{c}}$ .

*Proof.* Fix an arbitrary realization of the private random seed  $\xi$ . Conditional on this seed, the algorithm's next comparison pair is a deterministic function of the previous comparison history.

For the linear preference relation  $\preceq_c$ , a comparison between  $p$  and  $q$  returns the sign of  $\langle c, q - p \rangle$ . If  $p = q$ , the comparison gives no information. Let  $D_0 := \mathbb{S}^{d-1}$  and let  $\text{area}(\cdot)$  denote surface area on  $\mathbb{S}^{d-1}$ . We adversarially construct a comparison history. Suppose that after  $i - 1$  replies, the set of directions consistent with the comparison history is a measurable set  $D_{i-1} \subseteq \mathbb{S}^{d-1}$ . Since the seed  $\xi$  is fixed, this comparison history uniquely determines the next comparison pair  $(p_i, q_i)$ . If  $p_i = q_i$ , set  $D_i := D_{i-1}$  and continue. Otherwise, set  $u_i := (q_i - p_i) / \|q_i - p_i\|$ . The two possible nonzero replies correspond to

$$D_{i-1}^+ := D_{i-1} \cap \{c : \langle c, u_i \rangle > 0\}, \quad D_{i-1}^- := D_{i-1} \cap \{c : \langle c, u_i \rangle < 0\}.$$

The boundary has surface area zero. Hence at least one of  $D_{i-1}^+$  and  $D_{i-1}^-$  has area at least  $\frac{1}{2} \text{area}(D_{i-1})$ . The adversary chooses the corresponding reply and sets  $D_i$  to that larger cell. Iterating yields

$$\text{area}(D_t) \geq 2^{-t} \text{area}(\mathbb{S}^{d-1}). \quad (29)$$

For  $\alpha \in [0, \pi]$  and  $v \in \mathbb{S}^{d-1}$ , define  $\text{Cap}(v, \alpha) := \{w \in \mathbb{S}^{d-1} : \angle(w, v) \leq \alpha\}$ . And we let  $\alpha_t \in (0, \pi)$  be the root of

$$\frac{\text{area}(\text{Cap}(v, \alpha_t))}{\text{area}(\mathbb{S}^{d-1})} = 2^{-(t+1)}, \quad (30)$$

which is independent of  $v$  by rotational symmetry. Choose any  $c \in D_t$ . By (29),  $D_t$  has area at least twice the area of  $\text{Cap}(c, \alpha_t)$ . Therefore  $D_t \setminus \text{Cap}(c, \alpha_t)$  has positive area, and we may choose  $\hat{c} \in D_t \setminus \text{Cap}(c, \alpha_t)$ .

Moreover,  $c$  and  $\hat{c}$  both lie in the same cell  $D_t$ , which is the intersection of the strict hemispheres selected by the constructed history. Therefore they give the same nonzero sign for every comparison along that history. Since the seed  $\xi$  is fixed, identical replies up to round  $i - 1$  force the algorithm to choose the same comparison pair at round  $i$ . By induction over  $i = 1, \dots, t$ , the full comparison history is identical under  $\preceq_c$  and  $\preceq_{\hat{c}}$ .

It remains to lower bound  $\alpha_t$ . By the spherical-cap formula,

$$\frac{\text{area}(\text{Cap}(v, \alpha_t))}{\text{area}(\mathbb{S}^{d-1})} = \frac{\int_0^{\alpha_t} \sin^{d-2} \theta d\theta}{\int_0^\pi \sin^{d-2} \theta d\theta} = 2^{-(t+1)}.$$

Using  $\sin \theta \leq \theta$  on  $[0, \alpha_t]$ , we get

$$\frac{\alpha_t^{d-1}}{d-1} \geq 2^{-(t+1)} \int_0^\pi \sin^{d-2} \theta d\theta. \quad (31)$$

Moreover, by the trigonometric representation of the Beta function,

$$\int_0^\pi \sin^{d-2} \theta d\theta = \sqrt{\pi} \frac{\Gamma(\frac{d-1}{2})}{\Gamma(\frac{d}{2})} = \sqrt{\pi} \frac{\Gamma(x)}{\Gamma(x + \frac{1}{2})}$$

for  $x := \frac{d-1}{2}$ . By Wendel's inequality, for every  $x > 0$  and every  $s \in (0, 1)$ ,  $\frac{\Gamma(x+s)}{\Gamma(x)} \leq x^s$ . Taking  $s = \frac{1}{2}$  yields  $\Gamma(x + \frac{1}{2})/\Gamma(x) \leq \sqrt{x}$  and hence

$$\int_0^\pi \sin^{d-2} \theta d\theta \geq \sqrt{\pi} \frac{1}{\sqrt{x}} = \sqrt{\frac{2\pi}{d-1}} \geq \sqrt{\frac{2\pi}{d}}.$$

Substituting this into (31) yields  $\alpha_t \geq \left(\sqrt{2\pi/d}(d-1)2^{-(t+1)}\right)^{1/(d-1)}$ . Since  $(\sqrt{2\pi/d}(d-1))^{1/(d-1)} \geq 1$  for all  $d \geq 2$ , we obtain  $\alpha_t \geq 2^{-(t+1)/(d-1)}$ . This proves (28).  $\square$

With this theorem, we can immediately obtain the following lower bound for normal direction estimation under the general class of preferences considered in this section.

**Corollary 3.15.** *Fix  $d \geq 2$  and  $\epsilon \in (0, 1/(2\pi))$ . Consider any algorithm with  $t$  comparisons in the sense of Definition 3.13. Suppose that, for every realization of  $\xi$  and every  $c \in \mathbb{S}^{d-1}$ , when the preference relation is  $\preceq_c$ , the algorithm outputs  $\hat{n} \in \mathbb{S}^{d-1}$  satisfying  $\|\hat{n} - c\| \leq \epsilon$ . Then*

$$t > (d-1) \log_2 \left( \frac{1}{\pi\epsilon} \right) - 1. \quad (32)$$

*Consequently, any comparison based algorithm that guarantees worst-case normal-direction error at most  $\epsilon$  over the class of plateau-free and regular preference relations must use  $\Omega(d \log(1/\epsilon))$  comparisons.*

*Proof.* The family  $\{\preceq_c: c \in \mathbb{S}^{d-1}\}$  is a subclass of the preference relations considered here. Fix any realization of the private random seed. By Theorem 3.14, there exist  $c, \hat{c} \in \mathbb{S}^{d-1}$  with identical comparison transcripts under this realization and  $\angle(c, \hat{c}) > 2^{-(t+1)/(d-1)}$ . Since the algorithm receives the same transcript, it produces the same output  $\hat{n}$  on both instances. The assumed guarantee gives  $\|\hat{n} - c\| \leq \epsilon$  and  $\|\hat{n} - \hat{c}\| \leq \epsilon$ . Therefore,  $\angle(c, \hat{c}) \leq \angle(c, \hat{n}) + \angle(\hat{n}, \hat{c}) \leq \pi\epsilon$ , where we used  $\|a - b\| = 2 \sin(\angle(a, b)/2)$  and  $\sin z \geq 2z/\pi$  for  $z \in [0, \pi/2]$ . Thus  $2^{-(t+1)/(d-1)} < \pi\epsilon$ , which rearranges to (32).  $\square$

Corollary 3.15 shows that to guarantee worst-case error smaller than  $\epsilon$  in normal direction estimation, any algorithm must use at least on the order of  $d \cdot \log(1/\epsilon)$  comparisons, which matches the scaling achieved by the proposed methods up to constants and logarithmic factors in  $d$  (Corollary 3.6; or additionally in  $h_0, 1/r_x$ , and  $1/\delta$  in Theorem 3.11).

During the preparation of the first version of this manuscript, the concurrent work [39] proves a lower bound  $\Omega(d \log(1/\epsilon))$  of comparison-based method for estimating a differentiable function's gradient direction up to  $\epsilon$ . Directly using their results also implies a lower bound for normal direction estimation. They use a different approach by first constructing a maximal  $\epsilon$ -net of the unit sphere and then showing that no method can distinguish an unknown point in the net in a certain number of comparisons.

### 3.5 From normal directions to cutting-plane methods

Recall that our goal is to solve (2), where we assume  $\mathcal{C}$  is convex and having access to efficient Euclidean projections onto  $\mathcal{C}$ . When the preference relation is convex, access to normal directions and Euclidean projections onto  $\mathcal{C}$  immediately yield valid separating halfspaces that contain the optimal set  $\mathcal{X}^*$ . Consequently, one may directly use cutting-plane methods, including the ellipsoid method and the analytic- or volumetric-center cutting-plane methods, to obtain logarithmic dependence on the target accuracy in its complexity. The main limitation is dimension: even for the best-known cutting-plane methods, the number of separating halfspaces scales linearly with  $d$ , and the per-iteration arithmetic and memory costs typically scale more steeply with  $d$ .

**Separating halfspaces from normal directions.** Assume throughout this subsection that the preference relation is plateau-free and convex, so every sublevel set  $\mathcal{S}_x$  is convex. A valid cut at a query point  $x$  is a closed halfspace  $H$  such that  $\mathcal{X}^* \subseteq H$  and  $x \notin \text{int}(H)$ . When  $x \in \mathcal{C} \setminus \mathcal{X}^*$  and the preference relation is  $r_x$ -regular at  $x$  with  $r_x > 0$ , the outward normal direction  $n_x$  defines the supporting halfspace

$$H_x := \{y \in \mathcal{X} : \langle n_x, y - x \rangle \leq 0\}. \quad (33)$$

By Lemma 3.2,  $\mathcal{S}_x \subseteq H_x$ . Moreover, every  $x^* \in \mathcal{X}^*$  satisfies  $x^* \preceq x$ , hence  $x^* \in \mathcal{S}_x$ . Therefore  $\mathcal{X}^* \subseteq \mathcal{S}_x \subseteq H_x$ . Thus  $H_x$  is a valid cut.

If instead  $x \notin \mathcal{C}$ , we can use a feasibility cut. Let  $\hat{x} = \Pi_{\mathcal{C}}(x)$  be the Euclidean projection of  $x$  onto  $\mathcal{C}$  and set  $v_x := \frac{x - \hat{x}}{\|x - \hat{x}\|}$ , then let

$$H_x := \{y \in \mathcal{X} : \langle v_x, y - \hat{x} \rangle \leq 0\}. \quad (34)$$

By convexity of  $\mathcal{C}$ ,  $\mathcal{C} \subseteq H_x$  and hence  $\mathcal{X}^* \subseteq \mathcal{C} \subseteq H_x$ . Therefore, whether the query point violates optimality or feasibility, one can construct a separating halfspace guaranteed to contain  $\mathcal{X}^*$ .

**Complexity of cutting plane methods.** Algorithm 4 is written in a generic form. Representatives include the ellipsoid method, analytic-center cutting-plane method, and volumetric-center cutting-plane method (see [17, 29]). Different cutting-plane methods share the same separation step and mainly differ in two places: how the query point  $x_t$  is chosen in line 3, and how the new localization region  $\Omega_{t+1}$  is represented after the new cut is added in line 6. It is often  $\Omega_{t+1} = \Omega_t \cap H_t$ , while ellipsoid methods update  $\Omega_{t+1}$  to a smaller outer ellipsoid of  $\Omega_t \cap H_t$ . As for the center selection (line 3), ellipsoid methods simply choose the center of the ellipsoid as it keeps  $\Omega_t$  as ellipsoids. Other cutting-plane methods keep  $\Omega_t$  as a polyhedron, so they may spend more efforts in finding the center. For example, the analytic-center cutting-plane method recenters to the analytic center of  $\Omega_t$  by solving a log-barrier problem, and the volumetric-center cutting-plane method recenters to the volumetric center of  $\Omega_t$  by solving a volumetric barrier problem.

---

**Algorithm 4** Generic cutting-plane methods for solving (2)

---

- 1: **Input:** a bounded localization set  $\Omega_0$  satisfying  $\mathcal{X}^* \subseteq \Omega_0$ .
  - 2: **for**  $t = 0, 1, 2, \dots$  **do**
  - 3:     *Center selection.* Choose  $x_t \in \Omega_t$ .
  - 4:     **If**  $\Omega_{t+1}$  is sufficiently small **return**  $x_t$ .
  - 5:     *Separation-oracle query.* Obtain a valid cut  $H_{x_t}$  via either (33) or (34).
  - 6:     *Update localization set.* Use  $H_t$  to update  $\Omega_{t+1}$  so that  $\mathcal{X}^* \subseteq \Omega_{t+1}$ .
  - 7: **end for**
- 

For a bounded initial region  $\Omega_0$  of diameter  $R$  and target accuracy  $\epsilon$  measured by the distance to  $\mathcal{X}^*$ , with a restatement of the complexity results, classical ellipsoid-type methods require  $O(d^2 \log(dR/\epsilon))$  separating-halfspace oracle calls, whereas volumetric-center based methods achieve sharper  $O(d \log(dR/\epsilon))$  dependence [17, 29, 20, 15]. In particular, [20] gives a cutting-plane method using expected  $O(d \log(dR/\epsilon))$  separating-halfspace oracle calls and  $O(d^3 \log^{O(1)}(dR/\epsilon))$  additional arithmetic time overall. Subsequently, [15] refines the additional overhead to essentially  $O(d^2)$  time per separating-halfspace oracle call while preserving the  $O(d \log(dR/\epsilon))$  separating-halfspace oracle complexity.

**Advantages and disadvantages.** Using cutting-plane methods as a direct solver based on normal direction estimation is straightforward. Normal direction estimation becomes a black-box separation halfspace oracle, and the optimization algorithm layer can directly leverage cutting-plane methods. Furthermore, cutting-plane methods usually exhibit polylogarithmic dependence on the target accuracy, especially suitable for high-accuracy requirements. However, there are two main drawbacks. First, even with the best cutting-plane method, the number of iterations scales linearly with  $d$ . In our setting, each iteration may require estimating a normal direction from comparisons, so the total comparison cost multiplies. Simply saying, if obtaining one reliable normal direction costs  $\tilde{O}(d)$  comparisons (here  $\tilde{O}$  omits logarithmic factors in accuracy and dimension), then the overall number of comparisons may be on the order of  $\tilde{O}(d^2 \log(dR/\epsilon))$ , again hiding only logarithmic factors in  $d$ ,  $R$ , and  $\epsilon^{-1}$ . Second, cutting-plane methods have to maintain information of the full-dimensional geometry via either polytopes or ellipsoids, so memory occupancy and arithmetic overhead costs scale at least quadratically in  $d$  per iteration, and can dominate in moderate-to-large dimensional problems.

A good algorithm should balance the trade-off between the dimension-dependent costs and the accuracy-dependent costs. Cutting-plane methods are ideal choice for getting high-accuracy solutions but incur high dimension-dependent costs. Once  $d$  becomes large, the costs in maintaining the full-dimensional geometry model and solving the centers can be prohibitive. Therefore, we still seek methods that do not maintain a full-dimensional geometry model and have smaller per-iteration overhead. In the next section, we will study two normal direction span-based methods, which are more scalable in high-dimensional settings, although they may exhibit worse dependence on the target accuracy.

## 4 Two normal direction span-based methods

In this section, we develop optimization methods based on the normal direction estimators from Section 3. Since normal directions point toward less-preferred regions, moving in their negatives is expected to produce more-preferred points. This leads to methods that update within the span

of previously estimated normal directions together with the projection corrections induced by the constraint set. We call them normal-direction span-based methods.

**Definition 4.1** (Normal-direction span-based methods). Fix  $K \in \mathbb{N}$ . A normal-direction span-based method maintains internal points  $z_1, \dots, z_{K+1} \in \mathcal{X}$  and the corresponding feasible points  $x_t := \Pi_{\mathcal{C}}(z_t)$  for  $t = 1, \dots, K + 1$ . At each iteration  $t = 1, \dots, K$ , the method receives the normal direction at  $x_t$  and receives  $n_t := n_{x_t}$ . It then chooses the next internal point so that

$$z_{t+1} \in z_1 + \text{span}\{n_1, \dots, n_t, x_1 - z_1, \dots, x_t - z_t\}. \quad (35)$$

After  $K$  iterations, it may output any  $\hat{x} \in \{x_1, \dots, x_{K+1}\}$ .

We study methods that use normal directions estimated by methods introduced in Section 3. Our analysis is entirely function-free: rather than relying on objective values or derivatives, we measure progress using the geometric quantities introduced in Section 2.4: the level-set optimality gap  $\Delta_{\text{LS}}(\cdot)$  and the regularity radius  $r_x$ .

We begin in Section 4.1 with a simple fixed-step normal direction descent method (NDD). Its guarantee is two-case: either the returned point has small level-set optimality gap, or it is preferred to an iterate with small regularity radius. Section 4.2 then introduces a local growth condition under which the second alternative can be converted into the first. Section 4.3 introduces an adaptive version, adaNDD, which removes the need to tune the comparison radius and step-size in advance. Finally, Section 4.4 shows a lower bound on the worst-case convergence rate of normal-direction span-based methods, which implies that NDD and adaNDD are nearly optimal in the number of normal directions.

#### 4.1 Normal direction descent method (NDD)

We now consider a basic normal direction descent scheme that repeatedly moves in the direction opposite to an estimated outward normal direction. At iteration  $k$ , given a feasible iterate  $x_k \in \mathcal{C}$ , we estimate the outward normal direction  $n_{x_k}$  using comparisons, and then take a fixed-length step in the negative estimated normal direction, followed by Euclidean projection onto  $\mathcal{C}$ :

$$x_{k+1} = \Pi_{\mathcal{C}}(x_k - \eta \hat{n}_k), \quad (36)$$

where  $\hat{n}_k$  is the output of Algorithm 2 run at the base point  $x_k$  with a fixed comparison radius  $h$  and  $T$  planar bisection steps.

When the preference relation is realized by a differentiable function  $f$ , then  $n_{x_k} = \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$ . This makes (36) precisely the normalized gradient descent (using an inexact normalized gradient estimator  $\hat{n}_k$ ). Normalized gradient descent methods have been studied in a range settings, including smooth convex and strongly convex objectives as well as smooth nonconvex problems; see, e.g., [28, 22, 30, 25].

However, we do not assume the existence of the objective function, and we focus on function-free optimization using (36) and measure progress using the level-set optimality gap and regularity radius introduced in Section 2. Algorithm 5 gives the full procedure. In addition to the parameters  $(h, T)$  for Algorithm 2, the method takes a stepsize  $\eta > 0$  and a horizon  $K$ . The algorithm maintains a best-so-far point using one additional comparison per iteration.

If the normal direction estimates  $\hat{n}_k$  are exact, Algorithm 5 is a normal-direction span-based method. Indeed, its internal update is  $z_{t+1} = x_t - \eta n_t$ , where  $x_t = \Pi_{\mathcal{C}}(z_t)$ . Since  $x_t = z_t + (x_t - z_t)$ ,  $z_{t+1} \in z_1 + \text{span}\{n_1, \dots, n_t, x_1 - z_1, \dots, x_t - z_t\}$ . The update satisfies the span restriction in Definition 4.1, and the best-so-far output is one of the visited feasible points.

---

**Algorithm 5** Normal direction descent (NDD) method
 

---

- 1: **Input:** initial point  $x_1 \in \mathcal{C}$ , stepsize  $\eta > 0$ , comparison radius  $h > 0$ , planar bisection steps  $T \in \mathbb{N}_{++}$ , horizon  $K \in \mathbb{N}$ .
  - 2: Initialize  $\hat{x} \leftarrow x_1$ .
  - 3: **for**  $k = 1, 2, \dots, K$  **do**
  - 4:   Compute normal direction estimate  $\hat{n}_k$  of  $x_k$  by Algorithm 2 with fixed parameters  $(h, T)$ .
  - 5:    $x_{k+1} \leftarrow \Pi_{\mathcal{C}}(x_k - \eta \hat{n}_k)$ .
  - 6:   **If**  $x_{k+1} \prec \hat{x}$  **then** set  $\hat{x} \leftarrow x_{k+1}$ .
  - 7: **end for**
  - 8: **return**  $\hat{x}$ .
- 

Before showing the main guarantees of Algorithm 5, we present two lemmas that are key to the analysis. The first lemma relates the level-set optimality gap  $\Delta_{\text{LS}}(x)$  to a supporting-hyperplane gap at  $x$ , and the second lemma shows that  $\Delta_{\text{LS}}$  is monotone with respect to the preference relation.

**Lemma 4.2.** *Assume the preference relation  $\preceq$  is convex and plateau-free. Fix  $x \in \mathcal{C}$  with  $x \notin \mathcal{X}^*$ , and let  $n_x$  be the outward unit normal at  $x \in \partial \mathcal{S}_x$ . Then for any  $x^* \in \mathcal{X}^*$ ,*

$$\Delta_{\text{LS}}(x) \leq \langle n_x, x - x^* \rangle. \quad (37)$$

*Proof.* By convexity and plateau-freeness of the preference relation,  $\mathcal{S}_x$  is convex and closed. Since  $\mathcal{S}_x$  is convex and admits an outward unit normal  $n_x$  at  $x \in \partial \mathcal{S}_x$ , it is contained in the supporting halfspace

$$\mathcal{S}_x \subseteq \{y \in \mathcal{X} : \langle n_x, y - x \rangle \leq 0\}. \quad (38)$$

Fix  $x^* \in \mathcal{X}^*$ . By definition of  $\mathcal{X}^*$ , we have  $x^* \preceq x$ , hence  $x^* \in \mathcal{S}_x$ . Let  $s_0 := \sup\{s \geq 0 : x^* + s n_x \in \mathcal{S}_x\}$  and set  $v := x^* + s_0 n_x$ . Then  $v \in \partial \mathcal{S}_x = \mathcal{L}_x$ . Due to (38), we have  $\langle n_x, v - x \rangle = \langle n_x, x^* + s_0 n_x - x \rangle = \langle n_x, x^* - x \rangle + s_0 \leq 0$ , which implies  $s_0 \leq \langle n_x, x - x^* \rangle$ . Therefore,

$$\Delta_{\text{LS}}(x) = \text{dist}(\mathcal{X}^*, \mathcal{L}_x) \leq \text{dist}(x^*, \mathcal{L}_x) \leq \|v - x^*\| = s_0 \leq \langle n_x, x - x^* \rangle.$$

This proves (37). □

**Lemma 4.3.** *Assume the preference relation  $\preceq$  is plateau-free. If  $y \preceq x$ , then*

$$\Delta_{\text{LS}}(y) \leq \Delta_{\text{LS}}(x). \quad (39)$$

*Proof.* If any of  $x$  and  $y$  is optimal, then the inequality holds trivially. We therefore assume  $x, y \notin \mathcal{X}^*$ . If  $y \preceq x$  then  $\mathcal{S}_y \subseteq \mathcal{S}_x$ , hence  $\mathcal{X} \setminus \mathcal{S}_x \subseteq \mathcal{X} \setminus \mathcal{S}_y$ . Since  $\mathcal{X}^* \subseteq \mathcal{S}_y$  and plateau-freeness implies  $\partial \mathcal{S}_z = \mathcal{L}_z$  and that  $\mathcal{S}_z$  is closed for all  $z$ , we have  $\text{dist}(\mathcal{X}^*, \mathcal{L}_z) = \text{dist}(\mathcal{X}^*, \mathcal{X} \setminus \mathcal{S}_z)$  for  $z \in \{x, y\}$ . Thus, enlarging the complement cannot increase its distance to  $\mathcal{X}^*$ , which yields (39). □

Now we are ready to state the main guarantee for Algorithm 5. Theorem 4.4 below shows that the method either returns a point with small level-set optimality gap, or reaches an iterate with sufficiently small regularity radius.

**Theorem 4.4** (Normal direction descent method). *Assume the preference relation  $\preceq$  is convex and plateau-free. Assume  $\mathcal{C} \subseteq \mathcal{X}$  is nonempty, closed, and convex, and  $d \geq 2$ . Assume furthermore that the preference relation is  $r_x$ -regular at every  $x \in \mathcal{C} \setminus \mathcal{X}^*$ , with regularity radius  $r_x > 0$ . Assume*

$x_1 \notin \mathcal{X}^*$  and write  $D_1 := \text{dist}(x_1, \mathcal{X}^*)$ . Run Algorithm 5 for  $K \in \mathbb{N}_{++}$  iterations with stepsize  $\eta = D_1/\sqrt{K}$ , comparison radius  $h$ , and  $T$  planar bisection steps, starting from  $x_1 \in \mathcal{C}$ . Then Algorithm 5 uses at most

$$N \leq K \left( (d-1)(T+3) + 2 \right)$$

pairwise comparisons. Let  $\tilde{D} := \min \{ D_1 + \frac{\eta}{2}(K-1), \text{diam}(\mathcal{C}) \}$ . Then at least one of the following holds:

(i)

$$\Delta_{\text{LS}}(\hat{x}) \leq \hat{\epsilon} := \frac{2D_1}{\sqrt{K}} + \frac{\pi\sqrt{d-1}\tilde{D}}{2^{T-1}}; \quad (40)$$

(ii) there exists  $k \in \{1, \dots, K\}$  such that

$$r_{x_k} < \frac{4\sqrt{d-1}\tilde{D}h}{\hat{\epsilon}}. \quad (41)$$

In this case,  $\hat{x} \preceq x_k$ .

*Proof.* If there exists some  $k \in \{1, \dots, K\}$  such that  $x_k \in \mathcal{X}^*$ , then the best-so-far rule implies  $\hat{x} \in \mathcal{X}^*$ , and hence alternative (i) holds trivially. We therefore assume  $x_k \notin \mathcal{X}^*$  for all  $k \in \{1, \dots, K\}$ .

Fix a run of Algorithm 5. For each  $k$ , write  $D_k := \text{dist}(x_k, \mathcal{X}^*)$ , choose  $x_k^* \in \arg \min_{z \in \mathcal{X}^*} \|x_k - z\|$ , and let  $n_k := n_{x_k}$ . Since  $\hat{x} \preceq x_k$  for every  $k$  by the best-so-far update, Lemmas 4.3 and 4.2 imply

$$\Delta_{\text{LS}}(\hat{x}) \leq \Delta_{\text{LS}}(x_k) \leq \langle n_k, x_k - x_k^* \rangle, \quad k = 1, \dots, K. \quad (42)$$

Using the update (36) with  $\eta = D_1/\sqrt{K}$ , the nonexpansiveness of Euclidean projection, and the fact that  $x_k^* \in \mathcal{X}^* \subseteq \mathcal{C}$ , we obtain

$$\begin{aligned} D_{k+1}^2 &= \text{dist}(x_{k+1}, \mathcal{X}^*)^2 \leq \|x_{k+1} - x_k^*\|^2 \leq \left\| x_k - \frac{D_1}{\sqrt{K}} \hat{n}_k - x_k^* \right\|^2 \\ &= D_k^2 - \frac{2D_1}{\sqrt{K}} \langle \hat{n}_k, x_k - x_k^* \rangle + \frac{D_1^2}{K}, \end{aligned} \quad (43)$$

Rearranging and using  $\|x_k - x_k^*\| = D_k$ , we obtain

$$\langle n_k, x_k - x_k^* \rangle \leq \frac{\sqrt{K}}{2D_1} \left( D_k^2 - D_{k+1}^2 + \frac{D_1^2}{K} \right) + D_k \|\hat{n}_k - n_k\|. \quad (44)$$

Summing (44) over  $k = 1, \dots, K$ , telescoping the  $D_k^2$  terms, and using  $D_{K+1}^2 \geq 0$ , we get

$$\sum_{k=1}^K \langle n_k, x_k - x_k^* \rangle \leq D_1 \sqrt{K} + \sum_{k=1}^K D_k \|\hat{n}_k - n_k\|. \quad (45)$$

Averaging (42) over  $k = 1, \dots, K$  and combining the result with (45) yields

$$\Delta_{\text{LS}}(\hat{x}) \leq \frac{D_1}{\sqrt{K}} + \frac{1}{K} \sum_{k=1}^K D_k \|\hat{n}_k - n_k\|. \quad (46)$$

Next we bound the error term on the right-hand side of (46). Since  $\|x_{k+1} - x_k\| \leq \eta = D_1/\sqrt{K}$ , the triangle inequality gives  $D_{k+1} \leq D_k + \eta$ , hence  $D_k \leq D_1 + (k-1)\eta$ . Also,  $D_k \leq \text{diam}(\mathcal{C})$  because  $x_k, x_k^* \in \mathcal{C}$ . Therefore

$$\frac{1}{K} \sum_{k=1}^K D_k \leq \min \left\{ D_1 + \frac{\eta}{2}(K-1), \text{diam}(\mathcal{C}) \right\} = \tilde{D}. \quad (47)$$

Assume first that (41) fails for all  $k \in \{1, \dots, K\}$ . Then Theorem 3.5 implies

$$\|\hat{n}_k - n_k\| \leq 2\sqrt{d-1} \frac{h}{r_{x_k}} + \frac{\pi\sqrt{d-1}}{2^T} \leq \frac{\hat{\epsilon}}{2\tilde{D}} + \frac{\pi\sqrt{d-1}}{2^T}, \quad (48)$$

Multiplying (48) by  $D_k$ , averaging over  $k$ , and using (47), we have

$$\frac{1}{K} \sum_{k=1}^K D_k \|\hat{n}_k - n_k\| \leq \frac{\hat{\epsilon}}{2} + \frac{\pi\sqrt{d-1}\tilde{D}}{2^T}. \quad (49)$$

Substituting (49) into (46) and using the definition of  $\hat{\epsilon}$  in (40), we obtain

$$\Delta_{\text{LS}}(\hat{x}) \leq \frac{D_1}{\sqrt{K}} + \frac{\hat{\epsilon}}{2} + \frac{\pi\sqrt{d-1}\tilde{D}}{2^T} = \hat{\epsilon}. \quad (50)$$

Thus alternative (i) follows from (50).

If instead (41) holds for some  $k \in \{1, \dots, K\}$ , then  $\hat{x} \preceq x_k$  by the best-so-far rule, so alternative (ii) holds.

Finally, Theorem 3.5 shows that each call to Algorithm 2 uses at most  $(d-1)(T+3)+1$  pairwise comparisons, and each iteration uses one additional comparison to update the best-so-far point. Therefore the claimed comparison bound follows.  $\square$

*Remark 4.5* (Comparison complexity). Fix  $\epsilon \in (0, D_1)$ . If one only asks that the bound in alternative (i) of Theorem 4.4 be at most  $\epsilon$ , one may take  $K = \lceil 16D_1^2/\epsilon^2 \rceil$  and  $T = \lceil \log_2(4\pi\sqrt{d}\tilde{D}/\epsilon) \rceil$ . For this choice of  $K$ , we also have  $\tilde{D} \leq D_1 + 2D_1^2/\epsilon$ . Then the whole algorithm uses at most  $\tilde{O}\left(\frac{dD_1^2}{\epsilon^2}\right)$  comparisons, where  $\tilde{O}$  hides only logarithmic factors of  $d$ ,  $D_1$  and  $\epsilon^{-1}$ . By contrast, making the threshold in alternative (ii) small only requires taking  $h$  sufficiently small, and this does not change the complexity bound.

Here we focus only on the case  $d \geq 2$  because when  $d = 1$ , a cutting-plane method as described in Section 3.5 is already efficient. Theorem 4.4 gives a two-case guarantee, but the two alternatives play different roles. The first alternative is the main near-optimality guarantee, stated directly in terms of the level-set optimality gap  $\Delta_{\text{LS}}(\hat{x})$ . The second alternative shows that the output  $\hat{x}$  is preferred to an iterate with small regularity radius. As discussed in Section 2 (and later in the next subsection), small regularity radius is itself a meaningful geometric certificate and is often associated with near-stationarity. It is not the part that drives the comparison complexity, because its threshold is proportional to the comparison radius  $h$ , so it can be made arbitrarily small by taking  $h$  sufficiently small. In this sense, the dominant part is alternative (i).

*Remark 4.6* (Normal direction complexity). If the normal direction  $\hat{n}_k$  used in the algorithm is equal to the real normal direction  $n_k$ , (46) implies that  $\Delta_{\text{LS}}(\hat{x}) \leq \frac{D_1}{\sqrt{K}}$ . This guarantee remains only if the preference is plateau-free and convex, and does not require any regularity condition.

When the preference relation is realized by a differentiable function  $f$ , we have  $n_{x_k} = \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$ , so Algorithm 5 is precisely projected normalized gradient descent with an inexact normalized gradient oracle. If assuming  $\hat{n}_k$  is equal to  $n_k$ , our proof up until (46) follows the classic proof of normalized subgradient method by [37]. The new ingredient here is that the normal direction is obtained only through comparisons, so the analysis additionally controls the accumulated error. See also [30, 12] for analyses of normalized gradient methods.

Furthermore, because the normalized gradient descent method also works for nonconvex optimization problems and NDD has the same update rule, we believe that NDD may also apply to preference relations without convexity assumption. We do not yet have a function-free analysis but the existing guarantees for normalized gradient descent methods on the underlying objective function may still apply to NDD; see, e.g., [21, 26].

## 4.2 A local growth condition of regularity radius

The second alternative in Theorem 4.4 is geometric. To interpret it, we now discuss a relation that arises in many common preference models: a small regularity radius often forces the current level-set optimality gap to be small. A convenient way to summarize this relation is the following local growth condition.

**Condition 4.7** (Local growth condition of regularity radius). For the problem (2), there exist constants  $\gamma_1 > 0$  and  $\gamma_2 \in (0, +\infty]$  such that for every  $x \in \mathcal{C} \setminus \mathcal{X}^*$ ,

$$r_x \geq \min \{ \gamma_1 \cdot \Delta_{\text{LS}}(x), \gamma_2 \}. \quad (51)$$

In words, as  $x$  approaches  $\mathcal{X}^*$  in terms of  $\Delta_{\text{LS}}(x)$ , the regularity radius  $r_x$  can shrink at most linearly in  $\Delta_{\text{LS}}(x)$ , and once  $x$  is sufficiently far from  $\mathcal{X}^*$ , the regularity radius is uniformly lower bounded by  $\gamma_2$ . Since  $\Delta_{\text{LS}}(x) \leq \text{dist}(x, \mathcal{X}^*)$  for every  $x \in \mathcal{X}$ , any stronger distance-based condition of the form  $r_x \geq \min \{ \gamma_1 \text{dist}(x, \mathcal{X}^*), \gamma_2 \}$  automatically implies Condition 4.7. This condition summarizes a relation that holds for many common preference relations in optimization. Below we present several examples of functions that are often seen as objective functions of optimization problems and realize preference relations satisfying Condition 4.7. These examples are stated for the unconstrained case  $\mathcal{C} = \mathcal{X}$  and the constrained case  $\mathcal{C} \subset \mathcal{X}$  will be discussed later.

*Example 4.8* (Convex quadratic functions). Let  $f(x) := \frac{1}{2}(x - \hat{x})^\top Q(x - \hat{x})$  be a convex quadratic function, for symmetric matrix  $Q \in \mathbb{R}^{d \times d}$  and vector  $\hat{x} \in \mathbb{R}^d$ . Then the preference relation realized by  $f$  satisfies Condition 4.7 with  $(\gamma_1, \gamma_2) = (\sqrt{\lambda_{\min}^+(Q)/\lambda_{\max}^+(Q)}, +\infty)$ , where  $\lambda_{\max}^+(Q)$  and  $\lambda_{\min}^+(Q)$  are the largest and smallest positive eigenvalues of  $Q$ , respectively.

*Proof.* Let  $\mathcal{N} := \ker(Q)$ . Then  $\mathcal{X}^* = \hat{x} + \mathcal{N}$ . Fix any  $x \notin \mathcal{X}^*$  and write  $c := f(x) > 0$ . For any  $y \in \mathcal{L}_x$ , write  $y - \hat{x} = v + w$  with  $v \in \mathcal{N}^\perp$  and  $w \in \mathcal{N}$ . Then  $c = f(y) = \frac{1}{2}v^\top Qv \leq \frac{\lambda_{\max}^+(Q)}{2} \|v\|_2^2 = \frac{\lambda_{\max}^+(Q)}{2} \text{dist}(y, \mathcal{X}^*)^2$ , so  $\text{dist}(y, \mathcal{X}^*) \geq \sqrt{2c/\lambda_{\max}^+(Q)}$ . Equality is attained by taking  $y - \hat{x}$  along an eigenvector corresponding to  $\lambda_{\max}^+(Q)$ , and hence  $\Delta_{\text{LS}}(x) = \sqrt{2c/\lambda_{\max}^+(Q)}$ . On the other hand, writing  $x - \hat{x} = v + w$  with  $v \in \mathcal{N}^\perp$  and  $w \in \mathcal{N}$ , we have  $\nabla f(x) = Qv$  and therefore  $\|\nabla f(x)\|_2^2 = v^\top Q^2v \geq \lambda_{\min}^+(Q) v^\top Qv = 2\lambda_{\min}^+(Q)c = \lambda_{\min}^+(Q)\lambda_{\max}^+(Q)\Delta_{\text{LS}}(x)^2$ . Since  $f$  is  $\lambda_{\max}^+(Q)$ -smooth, Lemma 2.10 gives  $r_x \geq \|\nabla f(x)\|_2/\lambda_{\max}^+(Q) \geq \sqrt{\lambda_{\min}^+(Q)/\lambda_{\max}^+(Q)} \Delta_{\text{LS}}(x)$ . This proves the claim.  $\square$

*Example 4.9* (Distance to a closed convex set). Let  $C_0 \subset \mathcal{X}$  be nonempty, closed, and convex, and let the preference relation be realized by  $f(x) := \text{dist}(x, C_0)$ . Then Condition 4.7 holds with  $(\gamma_1, \gamma_2) = (1, +\infty)$ .

*Proof.* Fix  $x \notin C_0$  and let  $\hat{x} := \Pi_{C_0}(x)$  be the Euclidean projection. Let  $t := \|x - \hat{x}\| = \text{dist}(x, C_0) = \text{dist}(x, \mathcal{X}^*) > 0$ . The sublevel set  $\mathcal{S}_x$  is  $\{u + tv : u \in C_0, v \in B(0, 1)\}$ . Since  $\hat{x} \in C_0$ , we have  $B(\hat{x}, t) \subseteq \mathcal{S}_x$  and  $x \in \partial B(\hat{x}, t)$ , so  $B(\hat{x}, t)$  is an interior tangent ball to  $\mathcal{S}_x$  at  $x$ . Moreover, since  $\mathcal{S}_x$  is convex, there exists an exterior tangent ball of radius  $t$  at  $x$ . Therefore,  $r_x \geq t$ . Since  $\Delta_{\text{LS}}(x) = \text{dist}(x, C_0) = t$ , Condition 4.7 holds with  $(\gamma_1, \gamma_2) = (1, +\infty)$ .  $\square$

*Example 4.10* (Convex  $L$ -smooth with local strong convexity). Assume the preference relation is realized by a convex differentiable function  $f : \mathcal{X} \rightarrow \mathbb{R}$  such that

1. ( *$L$ -smoothness*)  $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$  for all  $x, y \in \mathcal{X}$ ;
2. (*local strong convexity*) there exist a minimizer  $x^* \in \mathcal{X}^*$  and constants  $\mu > 0$  and  $\rho > 0$  such that  $f$  is  $\mu$ -strongly convex on the ball  $B(x^*, \rho)$ , i.e., for all  $x, y \in B(x^*, \rho)$ ,  $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2$ .

Then Condition 4.7 holds with  $(\gamma_1, \gamma_2) = (\frac{\mu}{L}, \frac{\mu\rho}{L})$ . Here  $\rho$  could be  $+\infty$  and then  $\gamma_2 = +\infty$ .

*Proof.* For  $x \notin \mathcal{X}^*$ , we give a lower bound on  $\|\nabla f(x)\|$ . If  $\|x - x^*\| \leq \rho$ , then strong convexity implies  $\langle \nabla f(x), x - x^* \rangle \geq \mu\|x - x^*\|^2$  and hence  $\|\nabla f(x)\| \geq \mu\|x - x^*\| = \mu \text{dist}(x, \mathcal{X}^*)$ .

If  $\|x - x^*\| \geq \rho$ , write  $u := (x - x^*)/\|x - x^*\|$  and define the one-dimensional restriction  $g(t) := f(x^* + tu)$ . Convexity of  $f$  implies  $g$  is convex, hence  $g'$  is nondecreasing. Let  $x_\rho := x^* + \rho u \in B(x^*, \rho)$ . Local strong convexity with  $(x, y) = (x_\rho, x^*)$  yields  $\langle \nabla f(x_\rho), x_\rho - x^* \rangle \geq \mu\rho^2$ , i.e.,  $g'(\rho) \geq \mu\rho$ . Therefore  $g'(t) \geq g'(\rho) \geq \mu\rho$  for all  $t \geq \rho$ , and in particular  $\|\nabla f(x)\| \geq \langle \nabla f(x), u \rangle = g'(\|x - x^*\|) \geq \mu\rho$ .

Applying Lemma 2.10 in both cases gives  $r_x \geq \frac{\|\nabla f(x)\|}{L} \geq \min\left\{\frac{\mu \text{dist}(x, \mathcal{X}^*)}{L}, \frac{\mu\rho}{L}\right\} \geq \min\left\{\frac{\mu \Delta_{\text{LS}}(x)}{L}, \frac{\mu\rho}{L}\right\}$ . This completes the proof.  $\square$

*Example 4.11* ( $L$ -smooth with PL inequality and quadratic growth). Assume the preference relation is realized by a differentiable function  $f : \mathcal{X} \rightarrow \mathbb{R}$ . Let  $f^* := \min_{z \in \mathcal{X}} f(z)$  be the optimal value. Assume further that  $f$  satisfies:

1. ( *$L$ -smoothness*)  $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$  for all  $x, y \in \mathcal{X}$ ;
2. (*PL inequality*) there exists  $\mu_{\text{PL}} > 0$  such that for all  $x \in \mathcal{X}$ ,  $\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu_{\text{PL}}(f(x) - f^*)$ .
3. (*Quadratic growth*) there exists  $\mu_{\text{QG}} > 0$  such that for all  $x \in \mathcal{X}$ ,  $f(x) - f^* \geq \frac{\mu_{\text{QG}}}{2} \text{dist}(x, \mathcal{X}^*)^2$ .

This function does not have to be convex. Then Condition 4.7 holds with  $(\gamma_1, \gamma_2) = \left(\frac{\sqrt{\mu_{\text{PL}}\mu_{\text{QG}}}}{L}, +\infty\right)$ .

*Proof.* Combining PL and quadratic growth gives  $\|\nabla f(x)\|^2 \geq \mu_{\text{PL}}\mu_{\text{QG}} \text{dist}(x, \mathcal{X}^*)^2$ , hence  $\|\nabla f(x)\| \geq \sqrt{\mu_{\text{PL}}\mu_{\text{QG}}} \text{dist}(x, \mathcal{X}^*)$ . Lemma 2.10 then yields  $r_x \geq \|\nabla f(x)\|/L \geq \sqrt{\mu_{\text{PL}}\mu_{\text{QG}}} \text{dist}(x, \mathcal{X}^*)/L \geq (\sqrt{\mu_{\text{PL}}\mu_{\text{QG}}}/L) \cdot \Delta_{\text{LS}}(x)$ .  $\square$

Nelder–Mead method is a classic derivative-free optimization method that relies solely on function comparisons. It is widely used in many real-world applications and for a while people believed for convex optimization it always converged to the optimal solution. However, McKinnon [24] showed that the method may converge to a nonstationary point and constructs a function example. We show that the induced preference relation of the example of Figure 2.2 in [24] satisfies Condition 4.7.

*Example 4.12* (McKinnon’s Nelder–Mead example). Consider the example function of [24]:  $f(x, y) = 360x^2 + y + y^2$  for  $x \leq 0$ , and  $f(x, y) = 6x^2 + y + y^2$  for  $x \geq 0$ . Then the preference relation realized by  $f$  satisfies Condition 4.7 with  $(\gamma_1, \gamma_2) = (1/360, +\infty)$ .

*Proof.* The unique minimizer is  $(0, -1/2)$ . Also,  $f$  is the sum of the globally 2-strongly convex function  $6x^2 + y + y^2$  and the convex function  $354(\max\{-x, 0\})^2$ , so  $f$  is globally 2-strongly convex. Moreover,  $\nabla f(x, y) = (720x, 1 + 2y)$  for  $x \leq 0$  and  $\nabla f(x, y) = (12x, 1 + 2y)$  for  $x \geq 0$ , so  $f$  is globally 720-smooth. Therefore Example 4.10 applies with  $L = 720$ ,  $\mu = 2$ , and  $\rho = +\infty$ , which gives  $(\gamma_1, \gamma_2) = (\mu/L, +\infty) = (1/360, +\infty)$ .  $\square$

There are other function examples in Section 3 of [24]. Among other examples that have smooth function level sets, it is still true that the regularity radius approaches 0 only when a point approaches the optimal solution.

The above examples show that Condition 4.7 holds for a wide range of preference relations in the unconstrained case  $\mathcal{C} = \mathcal{X}$ . They also apply to constrained problems  $\mathcal{C} \subset \mathcal{X}$  whenever the optimal solutions coincide with the optimal solutions of the corresponding unconstrained problem, because the regularity radius and the level-set optimality gap are not changing if enforcing the constraint does not change the optimal solution set.

The next example covers directly a constrained optimization situation.

*Example 4.13* (Bounded feasible set with nonvanishing gradient). Consider the problem (2). Assume that the preference relation is realized by an  $L$ -smooth differentiable function  $f : \mathcal{X} \rightarrow \mathbb{R}$  with  $L > 0$  and suppose that the gradient of  $f$  is bounded away from zero on  $\mathcal{C}$ , i.e.,  $g_{\mathcal{C}} := \inf_{x \in \mathcal{C}} \|\nabla f(x)\| > 0$ . Then Condition 4.7 holds for any  $\gamma_1 > 0$  and  $\gamma_2 = \frac{g_{\mathcal{C}}}{L}$ .

*Proof.* Fix any  $x \in \mathcal{C} \setminus \mathcal{X}^*$ . By Lemma 2.10,  $r_x \geq \frac{\|\nabla f(x)\|}{L} \geq \frac{g_{\mathcal{C}}}{L}$ , which proves Condition 4.7 with any  $\gamma_1 > 0$  and  $\gamma_2 = \frac{g_{\mathcal{C}}}{L}$ .  $\square$

This example is especially relevant when the optimal solution is attained on the boundary of  $\mathcal{C}$ . In such cases, the gradient of a smooth realization need not vanish at the optimum. If the gradient norm is uniformly bounded away from zero on  $\mathcal{C}$ , then the regularity radius is uniformly positive there, and Condition 4.7 follows immediately.

It should also be noted that a given preference relation can be realized by multiple functions that are monotone reformulations of each other (Lemma 2.8), and a preference relation satisfying Condition 4.7 need not be realized by any of the classes above.

We now return to Theorem 4.4. Under Condition 4.7, the small regularity radius alternative in Theorem 4.4 can be converted into a bound on the level-set optimality gap  $\Delta_{\text{LS}}(x)$ . This yields the following corollary, which recovers a pure  $\epsilon$ -level-set guarantee for a concrete choice of parameters.

**Corollary 4.14.** *Under the assumptions of Theorem 4.4, suppose that Condition 4.7 holds with parameters  $(\gamma_1, \gamma_2)$ . Fix  $\epsilon \in (0, D_1)$ , where  $D_1 := \text{dist}(x_1, \mathcal{X}^*)$ . Choose*

$$K := \left\lceil \frac{6D_1^2}{\epsilon^2} \right\rceil, \quad T := \left\lceil \log_2 \left( 14\pi\sqrt{d} \left( 1 + \frac{D_1}{\epsilon} \right)^2 \right) \right\rceil, \quad 0 < h \leq \frac{\min\{\epsilon\gamma_1, \gamma_2\}}{7\sqrt{d} \left( 1 + \frac{D_1}{\epsilon} \right)^2}.$$

*Then Algorithm 5 returns  $\hat{x}$  such that  $\Delta_{\text{LS}}(\hat{x}) \leq \epsilon$ . Moreover, the total number of comparisons satisfies*

$$N \leq 7d \left( 1 + \frac{D_1^2}{\epsilon^2} \right) \log_2 \left( 224\pi \cdot d \left( 1 + \frac{D_1}{\epsilon} \right)^2 \right). \quad (52)$$

*Proof.* Let  $z := D_1/\epsilon > 1$  and  $\eta = D_1/\sqrt{K}$ . Since  $K = \lceil 6z^2 \rceil$ , we have  $K - 1 < 6z^2$ ,  $\sqrt{K} \geq \sqrt{6}z$ , and  $K < 7z^2$ . Consequently,

$$\tilde{D} \leq D_1 + \frac{\eta}{2}(K - 1) < \epsilon \left( z + \frac{\sqrt{6}}{2}z^2 \right) \leq \frac{\sqrt{6}}{2} \epsilon(1 + z)^2. \quad (53)$$

Also, by the choice of  $T$ , we have  $2^{T-1} \geq 7\pi\sqrt{d}(1+z)^2$ . Combining this with (53) and  $\sqrt{d-1} \leq \sqrt{d}$  yields

$$\hat{\epsilon} = \frac{2D_1}{\sqrt{K}} + \frac{\pi\sqrt{d-1}\tilde{D}}{2^{T-1}} \leq \frac{2}{\sqrt{6}}\epsilon + \frac{\sqrt{6}}{14}\epsilon < \epsilon. \quad (54)$$

Moreover, since  $\hat{\epsilon} \geq 2D_1/\sqrt{K}$ , (53) and  $K < 7z^2$  imply  $\frac{4\sqrt{d-1}\tilde{D}h}{\hat{\epsilon}} \leq \frac{2\sqrt{d}\tilde{D}h\sqrt{K}}{D_1} < \sqrt{42}\sqrt{d}(1+z)^2h$ . Using the choice of  $h$  and the fact that  $\sqrt{42} < 7$ , we obtain

$$\frac{4\sqrt{d-1}\tilde{D}h}{\hat{\epsilon}} < \min\{\epsilon\gamma_1, \gamma_2\}. \quad (55)$$

We now apply Theorem 4.4. If alternative (i) holds, then (54) gives  $\Delta_{\text{LS}}(\hat{x}) \leq \epsilon$ . Otherwise, alternative (ii) and (55) yield some  $k \in \{1, \dots, K\}$  such that  $\hat{x} \preceq x_k$  and  $r_{x_k} < \min\{\epsilon\gamma_1, \gamma_2\}$ . Since  $x_k \in \mathcal{C} \setminus \mathcal{X}^*$ , Condition 4.7 then implies  $\Delta_{\text{LS}}(x_k) < \epsilon$ . By Lemma 4.3,  $\Delta_{\text{LS}}(\hat{x}) \leq \Delta_{\text{LS}}(x_k) < \epsilon$ . Thus  $\Delta_{\text{LS}}(\hat{x}) \leq \epsilon$  in either case.

Finally, Theorem 4.4 gives  $N \leq K((d-1)(T+3)+2) \leq Kd(T+3)$ . Since  $K \leq 7(1+D_1^2/\epsilon^2)$  and  $T+3 \leq \log_2(14\pi\sqrt{d}(1+\frac{D_1}{\epsilon})^2) + 4 \leq \log_2(224\pi d(1+\frac{D_1}{\epsilon})^2)$ , we obtain (52).  $\square$

Corollary 4.14 shows that, under the local growth condition of the regularity radius, the two-case guarantee in Theorem 4.4 can be converted into a pure level-set guarantee. In particular, to obtain  $\Delta_{\text{LS}}(\hat{x}) \leq \epsilon$ , it suffices to take  $\tilde{O}(dD_1^2/\epsilon^2)$  comparisons, where  $\tilde{O}(\cdot)$  hides only logarithmic factors of  $d$ ,  $D_1$ , and  $\epsilon^{-1}$ .

### 4.3 Adaptive normal direction descent method (adaNDD)

Although we have already provided the NDD method in Section 4.1, it requires tuning the stepsize  $\eta$  and the comparison radius  $h$ . In particular, the parameter choices must depend on problem-related quantities such as  $\text{dist}(x_1, \mathcal{X}^*)$  and the local growth parameters  $(\gamma_1, \gamma_2)$  from Condition 4.7. In this subsection, we present an adaptive alternative that is called adaNDD. It avoids explicit tuning by combining: (i) Algorithm 2 implemented with the adaptive normal direction estimation routine from Section 3.3, and (ii) an adaptive normal direction descent method based on coin-betting.

The method maintains an internal sequence  $\{z_k\}_{k \geq 1} \subseteq \mathcal{X}$  and the corresponding feasible projected iterates  $x_k := \Pi_{\mathcal{C}}(z_k)$ . At each iteration  $k$ , we call Algorithm 2 with adaptive comparison radius at the feasible point  $x_k$  and request a target normal direction accuracy  $\epsilon_k$ . To achieve this accuracy, we set a particular planar bisection depth  $T_k$  so that the returned direction  $\hat{n}_k$  satisfies  $\|\hat{n}_k - n_{x_k}\| \leq \epsilon_k$ .

The number of comparisons used by the adaptive normal direction estimator can vary across iterations. To control the overall comparison complexity, we use a budgeted implementation in which the  $k$ -th call to Algorithm 2 with line search is allowed at most

$$B_k := 2(d-1) \left\lceil \log_2 \left( \frac{7\sqrt{d-1}}{\epsilon_k} \right) \right\rceil + 2 \left\lceil \log_2 \left( \frac{h_0 d^3}{r^* \epsilon_k} \right) \right\rceil_+ + 16 + 4 \left\lceil \log_2 \left( \frac{\pi^2 k^2}{6\delta} \right) \right\rceil \quad (56)$$

comparisons, for target regularity radius  $r^* > 0$  (as an alternative optimality criterion rather than the level-set optimality gap) and initial radius  $h_0 > 0$ . If the budget is exhausted before the normal-estimation call terminates, the method stops immediately and returns the current best-so-far point.

Overall, the adaNDD method is presented in Algorithm 6. It does not require advance knowledge of  $\text{dist}(x_1, \mathcal{X}^*)$  or Condition 4.7 parameters to choose a stepsize or a fixed comparison radius. The user specifies only the initial comparison radius  $h_0$ , the target regularity radius  $r^*$ , the confidence level  $\delta$ , and the maximal iteration budget  $K$ .

---

**Algorithm 6** Adaptive normal direction descent method (adaNDD)

---

- 1: **Input:** initial point  $x_1 \in \mathcal{C}$ , initial comparison radius  $h_0 > 0$ , target regularity radius  $r^* > 0$ , confidence level  $\delta \in (0, 1)$ , horizon  $K \in \mathbb{N}$ .
  - 2: Initialize shared radius  $h \leftarrow h_0$ ,  $S_0 \leftarrow 0$ ,  $A_0 \leftarrow 0$ ,  $\hat{x} \leftarrow x_1$ , and  $z_1 \leftarrow x_1$ .
  - 3: **for**  $k = 1, 2, \dots, K$  **do**
  - 4: Set  $\epsilon_k \leftarrow \min\left\{\frac{1}{2}, \frac{1}{\sqrt{k}(1+\|x_k-x_1\|)}\right\}$ ,  $T_k \leftarrow \left\lceil \log_2\left(\frac{\pi\sqrt{d-1}}{2\epsilon_k}\right) \right\rceil$  and  $B_k$  as in (56).
  - 5: Compute normal direction estimate  $\hat{n}_k$  of  $x_k$  by Algorithm 2 with adaptive comparison radius initialized at radius  $h$ , planar bisection depth  $T_k$ , and a budget  $B_k$  of comparisons.
  - 6: **If** the budget is exhausted before line 5 terminates **then return**  $\hat{x}$ .
  - 7: **If**  $z_k \in \mathcal{C}$  **then**  $g_k \leftarrow \hat{n}_k$ ; **else**  $s_k \leftarrow \frac{z_k - x_k}{\|z_k - x_k\|}$  and  $g_k \leftarrow \hat{n}_k + \max\{0, -\langle \hat{n}_k, s_k \rangle\} s_k$ .
  - 8:  $S_k \leftarrow S_{k-1} + g_k$ .
  - 9:  $A_k \leftarrow A_{k-1} + \langle g_k, z_k - x_1 \rangle$ .
  - 10:  $z_{k+1} \leftarrow x_1 - \frac{1 - A_k}{k+1} S_k$ .
  - 11:  $x_{k+1} \leftarrow \Pi_{\mathcal{C}}(z_{k+1})$ .
  - 12: **If**  $x_{k+1} \prec \hat{x}$  **then set**  $\hat{x} \leftarrow x_{k+1}$ .
  - 13: **end for**
  - 14: **return**  $\hat{x}$ .
- 

Algorithm 6 is also a normal direction span-based method if the normal direction estimates are exact. Indeed, if  $z_t \in \mathcal{C}$  then  $g_t = n_t$ , while if  $z_t \notin \mathcal{C}$  then  $g_t = n_t + \beta_t s_t$  with  $s_t$  proportional to  $z_t - x_t$ . Thus  $g_t \in \text{span}\{n_t, x_t - z_t\}$ . Since  $z_{t+1} = x_1 - \frac{1-A_t}{t+1} \sum_{i=1}^t g_i$ , it follows that  $z_{t+1} \in x_1 + \text{span}\{n_1, \dots, n_t, x_1 - z_1, \dots, x_t - z_t\}$ , which is precisely the span restriction in Definition 4.1. The scalar factor  $\frac{1-A_k}{k+1}$  is updated online through the cumulative inner products  $\langle g_i, z_i - x_1 \rangle$ . This is exactly the KT coin-betting update (Algorithm 7) for online linear optimization written in anchored form; see, e.g., [31].

---

**Algorithm 7** KT coin-betting for OLO over a Hilbert space  $H$  ([31, Algorithm 1])

---

- 1: **Input:** parameter  $d_0 > 0$ .
  - 2: **for**  $t = 1, 2, \dots$  **do**
  - 3: Predict with  $w_t \leftarrow \frac{1}{t} \left( d_0 + \sum_{i=1}^{t-1} \langle g_i, w_i \rangle \right) \sum_{i=1}^{t-1} g_i$   $\triangleright$  empty sums for  $t = 1$  give  $w_1 = 0$
  - 4: Receive  $g_t \in H$  such that  $\|g_t\| \leq 1$ .
  - 5: **end for**
- 

The above KT coin-betting algorithm enjoys a sublinear regret guarantee in online linear optimization, which we restate below.

**Lemma 4.15** (KT coin-betting regret bound [31, Corollary 5]). *Let  $H$  be a Hilbert space and let  $(g_t)_{t=1}^K \subseteq H$  satisfy  $\|g_t\| \leq 1$ . In online linear optimization, on each round  $t$  the learner chooses  $w_t \in H$ , then observes  $g_t$ . Then Algorithm 7 guarantees that for all  $u \in H$  and all  $K \geq 1$ ,*

$$\sum_{t=1}^K \langle g_t, u - w_t \rangle \leq \|u\| \sqrt{K \ln\left(1 + \frac{24K^2\|u\|^2}{d_0^2}\right)} + d_0 \left(1 - \frac{1}{e\sqrt{\pi K}}\right), \quad \forall d_0 > 0. \quad (57)$$

With Lemma 4.15, we can now derive an upper bound on the cumulative inner products  $\sum_{t=1}^K \langle g_t, z_t - x^* \rangle$  for Algorithm 6. This idea has been also used by [30] to design a parameter-free normalized gradient descent method for unconstrained convex optimization.

**Lemma 4.16.** *Consider a run of Algorithm 6 that completes  $K$  iterations. For every  $x^* \in \mathcal{X}$ , it holds that  $\|g_t\| \leq 1$  for all  $t$  and*

$$\sum_{t=1}^K \langle g_t, z_t - x^* \rangle \leq \|x^* - x_1\| \sqrt{K \ln(1 + 24 K^2 \|x^* - x_1\|^2)} + 1. \quad (58)$$

*Proof.* First,  $\|g_k\| \leq 1$  for every  $k$ . If  $z_k \in \mathcal{C}$ , then  $g_k = \hat{n}_k$ , so this follows from  $\|\hat{n}_k\| = 1$ . If  $z_k \notin \mathcal{C}$ , then  $g_k = \hat{n}_k + \beta_k s_k$  with  $\beta_k = \max\{0, -\langle \hat{n}_k, s_k \rangle\}$ . If  $\langle \hat{n}_k, s_k \rangle \geq 0$ , then  $g_k = \hat{n}_k$ ; otherwise  $g_k = \hat{n}_k - \langle \hat{n}_k, s_k \rangle s_k$ , and hence  $\|g_k\|^2 = \|\hat{n}_k\|^2 - \langle \hat{n}_k, s_k \rangle^2 \leq 1$ .

Define  $w_t := z_t - x_1 \in \mathcal{X}$  so that  $w_1 = 0$ , and set  $r_t := -g_t$ . Then the recursion on  $(w_t)$  coincides with the KT update in Algorithm 7 run with parameter  $d_0 = 1$  and feedback  $r_t$ . Apply Lemma 4.15 in  $H = \mathcal{X}$  with comparator  $u := x^* - x_1$  to obtain

$$\sum_{t=1}^K \langle r_t, u - w_t \rangle \leq \|u\| \sqrt{K \ln(1 + 24 K^2 \|u\|^2)} + \left(1 - \frac{1}{e\sqrt{\pi K}}\right) \leq \|u\| \sqrt{K \ln(1 + 24 K^2 \|u\|^2)} + 1.$$

Finally,

$$\sum_{t=1}^K \langle r_t, u - w_t \rangle = \sum_{t=1}^K \langle -g_t, (x^* - x_1) - (z_t - x_1) \rangle = \sum_{t=1}^K \langle g_t, z_t - x^* \rangle,$$

which yields (58).  $\square$

Furthermore, Lemma 4.17 below shows that each  $\langle g_k, z_k - x^* \rangle$  provides an upper bound for  $\langle \hat{n}_k, x_k - x^* \rangle$ . This step is critical in bridging the internal iterates  $z_k$  and the projected iterates  $x_k$ .

**Lemma 4.17.** *Assume  $\mathcal{C} \subseteq \mathcal{X}$  is nonempty, closed, and convex. Consider any iteration  $k$  of Algorithm 6 at which  $g_k$  is defined. For every  $x^* \in \mathcal{C}$ , it holds that*

$$\langle \hat{n}_k, x_k - x^* \rangle \leq \langle g_k, z_k - x^* \rangle. \quad (59)$$

*Proof.* If  $z_k \in \mathcal{C}$ , then  $x_k = z_k$  and  $g_k = \hat{n}_k$  by Algorithm 6, so (59) holds with equality. Assume now that  $z_k \notin \mathcal{C}$ . By Algorithm 6,  $s_k = (z_k - x_k)/\|z_k - x_k\|$  and  $g_k = \hat{n}_k + \beta_k s_k$ , where  $\beta_k := \max\{0, -\langle \hat{n}_k, s_k \rangle\}$ . Write  $z_k = x_k + \alpha_k s_k$ , where  $\alpha_k := \|z_k - x_k\| > 0$ . Projection optimality gives  $\langle s_k, z - x_k \rangle \leq 0$  for all  $z \in \mathcal{C}$ , and hence  $\langle s_k, x_k - x^* \rangle \geq 0$  for all  $x^* \in \mathcal{C}$ . Moreover,  $\langle g_k, s_k \rangle = \langle \hat{n}_k, s_k \rangle + \beta_k = \max\{\langle \hat{n}_k, s_k \rangle, 0\} \geq 0$ . Therefore,

$$\langle g_k, z_k - x^* \rangle = \langle g_k, x_k - x^* \rangle + \alpha_k \langle g_k, s_k \rangle \geq \langle \hat{n}_k, x_k - x^* \rangle,$$

where the last inequality uses  $g_k - \hat{n}_k = \beta_k s_k$ ,  $\beta_k \geq 0$ , and  $\langle s_k, x_k - x^* \rangle \geq 0$ . This completes the proof.  $\square$

Additionally, it can be proven that the internal iterates  $z_k$  (and  $x_k = \Pi_{\mathcal{C}}(z_k)$ ) generated by Algorithm 6 remain within a bounded distance from the initial point  $x_1$ . The boundedness of the iterates is used in analyzing the effect of the error of normal direction estimates.

**Lemma 4.18.** *Consider a run of Algorithm 6 that completes  $K$  iterations, and assume  $x_k \notin \mathcal{X}^*$  for all  $k \leq K$  (otherwise the best-so-far output is already optimal). Let  $D_1 := \text{dist}(x_1, \mathcal{X}^*)$ . Then for all  $k \in \{1, \dots, K + 1\}$ ,*

$$\|x_k - x_1\| \leq \|z_k - x_1\| \leq 1 + (k - 1)D_1 + 2\sqrt{k - 1}. \quad (60)$$

*Proof.* Fix  $x^* \in \mathcal{X}^*$  such that  $\|x^* - x_1\| = D_1$  and write  $w_k := z_k - x_1$ . Recall from Algorithm 6 that  $w_{k+1} = -\frac{1-A_k}{k+1}S_k$ , where  $S_k = \sum_{i=1}^k g_i$  and  $A_k = \sum_{i=1}^k \langle g_i, w_i \rangle$ . By Lemma 4.16 with comparator  $x^* = x_1$  and horizon  $k$ , we have  $A_k \leq 1$ , and hence  $(1 - A_k)/(k + 1) \geq 0$ . Moreover,  $\|g_i\| \leq 1$  implies  $\|S_k\| \leq k$ . Therefore

$$\|w_{k+1}\| = \frac{1 - A_k}{k + 1} \|S_k\| \leq \frac{1 - A_k}{k + 1} k \leq 1 - A_k. \quad (61)$$

We now upper bound  $1 - A_k$ . For each  $i \leq k$ , let  $n_i := n_{x_i}$ . First consider the case  $z_i \in \mathcal{C}$ . Then  $x_i = z_i$ ,  $g_i = \hat{n}_i$ , and

$$\langle g_i, w_i \rangle = \langle n_i, z_i - x_1 \rangle + \langle \hat{n}_i - n_i, w_i \rangle. \quad (62)$$

Since  $x^* \in \mathcal{X}^* \subseteq \mathcal{S}_{x_i}$  and  $n_i$  is an outward normal at  $x_i \in \partial\mathcal{S}_{x_i}$ , the supporting-halfspace property yields  $\langle n_i, x_i - x^* \rangle \geq 0$ . Because  $z_i = x_i$  in this case,

$$\langle n_i, z_i - x_1 \rangle = \langle n_i, x_i - x^* \rangle + \langle n_i, x^* - x_1 \rangle \geq -D_1. \quad (63)$$

Also, Theorem 3.11 and the definition of  $\epsilon_i$  in Algorithm 6 give  $\|\hat{n}_i - n_i\| \leq \epsilon_i \leq \frac{1}{\sqrt{i}(1+\|x_i - x_1\|)} = \frac{1}{\sqrt{i}(1+\|w_i\|)}$ , so by Cauchy–Schwarz,

$$\langle \hat{n}_i - n_i, w_i \rangle \geq -\|\hat{n}_i - n_i\| \|w_i\| \geq -\frac{1}{\sqrt{i}}. \quad (64)$$

Combining (62), (63), and (64) yields  $\langle g_i, w_i \rangle \geq -D_1 - \frac{1}{\sqrt{i}}$ .

Now consider the case  $z_i \notin \mathcal{C}$ . Let  $s_i = (z_i - x_i)/\|z_i - x_i\|$ , set  $\beta_i = \max\{0, -\langle \hat{n}_i, s_i \rangle\}$ , and define  $g_i = \hat{n}_i + \beta_i s_i$ . By projection optimality and  $x_1 \in \mathcal{C}$ , we have  $\langle s_i, x_i - x_1 \rangle \geq 0$  and  $\langle g_i, s_i \rangle = \max\{\langle \hat{n}_i, s_i \rangle, 0\} \geq 0$ . Since  $w_i = z_i - x_1 = (x_i - x_1) + \|z_i - x_i\|s_i$ , it follows that

$$\langle g_i, w_i \rangle = \langle g_i, x_i - x_1 \rangle + \|z_i - x_i\| \langle g_i, s_i \rangle \geq \langle \hat{n}_i, x_i - x_1 \rangle. \quad (65)$$

Using  $\langle \hat{n}_i, x_i - x_1 \rangle = \langle n_i, x_i - x_1 \rangle + \langle \hat{n}_i - n_i, x_i - x_1 \rangle$ , together with (63) and the same Cauchy–Schwarz bound as above, we obtain  $\langle \hat{n}_i, x_i - x_1 \rangle \geq -D_1 - \frac{1}{\sqrt{i}}$ . Combining this with (65) yields  $\langle g_i, w_i \rangle \geq -D_1 - \frac{1}{\sqrt{i}}$ .

Hence  $\langle g_i, w_i \rangle \geq -D_1 - \frac{1}{\sqrt{i}}$  for every  $i \leq k$ . Therefore  $1 - A_k = 1 - \sum_{i=1}^k \langle g_i, w_i \rangle \leq 1 + kD_1 + \sum_{i=1}^k \frac{1}{\sqrt{i}} \leq 1 + kD_1 + 2\sqrt{k}$ . Plugging this into (61) gives  $\|w_{k+1}\| \leq 1 + kD_1 + 2\sqrt{k}$ . Reindexing yields the second inequality of (60). Finally, by nonexpansiveness of projection,  $\|x_k - x_1\| = \|\Pi_{\mathcal{C}}(z_k) - \Pi_{\mathcal{C}}(x_1)\| \leq \|z_k - x_1\|$ , which gives the first inequality of (60).  $\square$

With the above lemmas in place, we are ready to state the main computational guarantees of the adaNDD method.

**Theorem 4.19** (Adaptive normal direction descent method). *Assume the preference relation  $\preceq$  is convex and plateau-free. Assume  $\mathcal{C} \subseteq \mathcal{X}$  is nonempty, closed, and convex, and  $d \geq 2$ . Assume furthermore that the preference relation is  $r_x$ -regular for every  $x \in \mathcal{C} \setminus \mathcal{X}^*$  with  $r_x > 0$ . Run Algorithm 6 for  $K \geq 1$  iterations, and write  $D_1 := \text{dist}(x_1, \mathcal{X}^*)$ . The total number of comparisons has the following upper bound:*

$$N \leq 2Kd \left\lceil \log_2 \left( 7d^{1/2} K^{3/2} (D_1 + 3) \right) \right\rceil + 2K \left( \left\lceil \log_2 \left( \frac{h_0 d^{5/2}}{r^*} \right) \right\rceil_+ + 2 \left\lceil \log_2 \left( \frac{2K^2}{\delta} \right) \right\rceil + 9 \right). \quad (66)$$

If the method completes  $K$  iterations, then its output  $\hat{x}$  satisfies

$$\Delta_{\text{LS}}(\hat{x}) \leq \frac{D_1 \sqrt{\ln(1 + 24 K^2 D_1^2)} + 2D_1 + 3}{\sqrt{K}}. \quad (67)$$

If the method stops early because the budget is exhausted at some iteration  $\tau$  during the normal-estimation call at  $x_\tau$ , then for every  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$  (over the internal randomness of Algorithm 2),

$$r_{x_\tau} < r^*. \quad (68)$$

If, in addition, Condition 4.7 holds with parameters  $(\gamma_1, \gamma_2)$  and  $r^* \leq \gamma_2$ , then on the same event,

$$\Delta_{\text{LS}}(\hat{x}) \leq \frac{r^*}{\gamma_1}. \quad (69)$$

*Proof.* We first show the proof of (67) and (66), the case the method completes  $K$  iterations.

If there exists some  $k \leq K + 1$  such that  $x_k \in \mathcal{X}^*$ , then the best-so-far rule implies  $\hat{x} \in \mathcal{X}^*$  and hence  $\Delta_{\text{LS}}(\hat{x}) = 0$ , so (67) holds trivially. We therefore assume  $x_k \notin \mathcal{X}^*$  for all  $k \leq K + 1$ , so that  $n_k := n_{x_k}$  exists and is unique.

Fix  $x^* \in \mathcal{X}^*$  such that  $\|x^* - x_1\| = D_1$ . By Lemma 4.2 and Lemma 4.3,

$$\begin{aligned} \Delta_{\text{LS}}(\hat{x}) &\leq \min_{1 \leq k \leq K} \Delta_{\text{LS}}(x_k) \leq \frac{1}{K} \sum_{k=1}^K \Delta_{\text{LS}}(x_k) \leq \frac{1}{K} \sum_{k=1}^K \langle n_k, x_k - x^* \rangle \\ &= \frac{1}{K} \underbrace{\sum_{k=1}^K \langle \hat{n}_k, x_k - x^* \rangle}_{\text{denoted by I}} - \frac{1}{K} \underbrace{\sum_{k=1}^K \langle \hat{n}_k - n_k, x_k - x^* \rangle}_{\text{denoted by II}} \end{aligned} \quad (70)$$

Below we derive an upper bound on I and a lower bound on II separately, and then combine them to obtain (67).

By Lemma 4.17, we have  $\langle \hat{n}_k, x_k - x^* \rangle \leq \langle g_k, z_k - x^* \rangle$ . Summing and applying Lemma 4.16 yields

$$\text{I} = \sum_{k=1}^K \langle \hat{n}_k, x_k - x^* \rangle \leq \sum_{k=1}^K \langle g_k, z_k - x^* \rangle \leq D_1 \sqrt{K \ln(1 + 24 K^2 D_1^2)} + 1. \quad (71)$$

By Theorem 3.11 and the choice of  $(\epsilon_k, T_k)$  in Algorithm 6,  $\|\hat{n}_k - n_k\| \leq \epsilon_k \leq \frac{1}{\sqrt{k} (1 + \|x_k - x_1\|)}$ . Note that  $\|x_k - x^*\| \leq \|x_k - x_1\| + \|x_1 - x^*\| \leq \|x_k - x_1\| + D_1$ , we obtain

$$|\langle \hat{n}_k - n_k, x_k - x^* \rangle| \leq \|\hat{n}_k - n_k\| \|x_k - x^*\| \leq \frac{\|x_k - x_1\| + D_1}{\sqrt{k} (1 + \|x_k - x_1\|)} \leq \frac{1 + D_1}{\sqrt{k}}.$$

Summing over  $k$  and using  $\sum_{k=1}^K k^{-1/2} \leq 2\sqrt{K}$  yields  $|\text{II}| \leq 2\sqrt{K}(1 + D_1)$ . Combining (70), (71), and  $|\text{II}| \leq 2\sqrt{K}(1 + D_1)$  gives (67).

Now we count the number of comparisons used by the method. Each completed iteration uses one additional comparison to update the best-so-far point, and the  $k$ -th normal-estimation call is allocated budget  $B_k$ . Therefore,  $N \leq K + \sum_{k=1}^K B_k$ . For each  $k \leq K$ , Lemma 4.18 yields  $1 + \|x_k - x_1\| \leq K(\text{dist}(x_1, \mathcal{X}^*) + 3)$ . Hence  $\epsilon_k^{-1} \leq K^{3/2}(\text{dist}(x_1, \mathcal{X}^*) + 3)$  and  $\frac{7\sqrt{d-1}}{\epsilon_k} \leq 7d^{1/2}K^{3/2}(\text{dist}(x_1, \mathcal{X}^*) + 3)$  and also

$$\frac{h_0 d^3}{r^* \epsilon_k} \leq \frac{h_0 d^3}{r^*} K^{3/2}(\text{dist}(x_1, \mathcal{X}^*) + 3) \leq (h_0 d^{5/2}/r^*) \cdot (d^{1/2}K^{3/2}(\text{dist}(x_1, \mathcal{X}^*) + 3)).$$

Moreover, since  $k \leq K$  and  $\pi^2/6 \leq 2$ ,  $\log_2\left(\frac{\pi^2 k^2}{6\delta}\right) \leq \log_2\left(\frac{2K^2}{\delta}\right)$ . Substituting these bounds into (56) and using  $\lceil \log_2(ab) \rceil_+ \leq \lceil \log_2(a) \rceil_+ + \lceil \log_2(b) \rceil$  (for  $b \geq 1$ ) yields, for all  $k \leq K$ ,

$$B_k \leq 2d \left\lceil \log_2\left(7d^{1/2}K^{3/2}(\text{dist}(x_1, \mathcal{X}^*) + 3)\right) \right\rceil + 2 \left\lceil \log_2\left(\frac{h_0 d^{5/2}}{r^*}\right) \right\rceil_+ + 16 + 4 \left\lceil \log_2\left(\frac{2K^2}{\delta}\right) \right\rceil.$$

Therefore, we obtain (66). If the method stops early at iteration  $\tau \leq K$ , then it uses at most  $\tau + \sum_{k=1}^{\tau} B_k \leq K + \sum_{k=1}^K B_k$  comparisons, so (66) also holds.

Next we show the proof of (68) and (69). Now we consider the case if the method stops early at iteration  $\tau \leq K$ . Fix  $\delta \in (0, 1)$  and define  $\delta_k := \frac{6\delta}{\pi^2 k^2}$  so that  $\sum_{k=1}^{\infty} \delta_k = \delta$ . Condition on the history up to iteration  $k$  (so  $x_k$  and  $z_k$  are fixed), and let  $N_k$  denote the number of pairwise comparisons that the unbudgeted call of Algorithm 2 at  $x_k$  would require before termination. By Theorem 3.11 (applied at  $x_k$  with target accuracy  $\epsilon_k$  and confidence  $\delta_k$ ), whenever  $x_k \notin \mathcal{X}^*$ ,

$$\mathbb{P}\left(N_k > \bar{N}_k \mid x_k\right) \leq \delta_k, \quad (72)$$

where

$$\bar{N}_k := 2(d-1) \left\lceil \log_2\left(\frac{7\sqrt{d-1}}{\epsilon_k}\right) \right\rceil + 2 \left\lceil \log_2\left(\frac{h_0 d^3}{r_{x_k} \epsilon_k}\right) \right\rceil_+ + 16 + 4 \left\lceil \log_2\left(\frac{1}{\delta_k}\right) \right\rceil.$$

Now assume  $r_{x_k} \geq r^*$ , then we have  $\frac{h_0 d^3}{r_{x_k} \epsilon_k} \leq \frac{h_0 d^3}{r^* \epsilon_k}$ . By monotonicity of  $z \mapsto \lceil \log_2(z) \rceil_+$ , this implies  $\bar{N}_k \leq B_k$ , with  $B_k$  defined in (56). Therefore,  $\mathbb{P}(N_k > B_k \mid x_k) \leq \delta_k$  whenever  $r_{x_k} > r^*$ .

A union bound over  $k \leq K$  shows that, except on an event of probability at most  $\delta$ , for every  $k \leq K$  we have the implication: if  $N_k > B_k$  then  $r_{x_k} < r^*$ . In particular, if the algorithm stops early at iteration  $\tau \leq K$ , then necessarily  $N_\tau > B_\tau$ , and hence (68) holds.

Finally, assume Condition 4.7 holds with parameters  $(\gamma_1, \gamma_2)$  and  $r^* \leq \gamma_2$ . On the same event we have  $r_{x_\tau} < r^* \leq \gamma_2$ , so (51) implies  $\Delta_{\text{LS}}(x_\tau) \leq \frac{r_{x_\tau}}{\gamma_1} < \frac{r^*}{\gamma_1}$ . Since  $\hat{x}$  is best-so-far and  $\Delta_{\text{LS}}(\cdot)$  is monotone under preference improvement (Lemma 4.3), we conclude  $\Delta_{\text{LS}}(\hat{x}) \leq \Delta_{\text{LS}}(x_\tau) \leq \frac{r^*}{\gamma_1}$ , which is (69).  $\square$

The theorem remains a two-case guarantee. If the method completes  $K$  iterations, then (67) gives a direct level-set optimality gap guarantee. If it stops early, then (68) certifies, with probability at least  $1 - \delta$ , that the current iterate has regularity radius smaller than  $r^*$ . This high-probability statement is inherited from the adaptive normal direction estimator.

Unlike NDD (Algorithm 5), adaNDD (Algorithm 6) does not require prior knowledge of problem-dependent quantities such as the distance to the optimal solutions  $\text{dist}(x_1, \mathcal{X}^*)$ , the diameter of  $\mathcal{C}$ , or the local growth condition parameters. Users only need to choose the initial comparison radius  $h_0$ , the target regularity radius  $r^*$ , and the confidence level  $\delta$ . In particular, no stepsize or comparison radius needs to be tuned. The level-set guarantee (67) is unaffected by the choice of  $(h_0, r^*, \delta)$ , while the total number of comparison bound (66) depends on  $(h_0, r^*, \delta)$  only through the logarithmic factors, which do not scale with  $d$ , and thus are often negligible compared with the leading term  $O(Kd \log(Kd(D_1 + 3)))$  of (66).

The guarantee (67) of completing  $K$  iterations holds for all  $K \geq 1$ . The bound automatically improves as  $K$  grows, without any need to preselect a target accuracy  $\epsilon$ . Compared with NDD with exact normal directions (see Remark 4.6), adaNDD achieves a similar dependence of (67) in  $K$  with only an additional logarithmic factor.

Condition 4.7 is used only after this point, to convert the small regularity radius certificate into the level-set optimality gap guarantee (69). Choosing the parameter  $r^*$  smaller affects the

comparison complexity (66) by a logarithmic factor. If Condition 4.7 holds with parameters  $(\gamma_1, \gamma_2)$ , then choosing any  $r^* \leq \min\{\epsilon\gamma_1, \gamma_2\}$  is enough to ensure  $\Delta_{\text{LS}}(\hat{x}) \leq \epsilon$  with probability at least  $1 - \delta$  on the event of stopping early.

#### 4.4 Lower bound of normal direction span-based methods

The convergence rate of the level-set optimality gap in NDD and adaNDD scale as  $O(\text{dist}(x_1, \mathcal{X}^*)/\sqrt{K})$  (or with a logarithmic factor) in the number of normal direction estimates  $K$  (see Remark 4.6 and Theorem 4.19). We now show that this dependence on  $K$  is unimprovable for the broad class of normal direction span-based methods. We will show that there exists a family of problems of the form (2) in which  $\mathcal{C}$  is nonempty, closed and convex, and the preference relation  $\preceq$  is convex, plateau-free and regular at every  $x \in \mathcal{C} \setminus \mathcal{X}^*$ , such that no normal direction span-based method can guarantee  $\Delta_{\text{LS}}(\hat{x}) < \frac{3 \text{dist}(x_1, \mathcal{X}^*)}{4\sqrt{K+1}}$  in  $K$  iterations for all these problems.

The preference relation is realized by a function  $f$  defined below. We first let

$$h(x) := \max_{1 \leq i \leq d} x_i. \quad (73)$$

And then let  $\rho : \mathbb{R}^d \rightarrow \mathbb{R}$  be a nonnegative  $C^\infty$  function satisfying

$$\int_{\mathbb{R}^d} \rho(z) dz = 1 \quad \text{and} \quad \{z \in \mathbb{R}^d : \rho(z) > 0\} \subseteq \{z \in \mathbb{R}^d : 0 < z_d < z_{d-1} < \dots < z_1 < 1\}. \quad (74)$$

Set  $\chi := D_1/(4\sqrt{d})$  and define  $\rho_\chi(z) := \chi^{-d}\rho(z/\chi)$ , so that  $\int_{\mathbb{R}^d} \rho_\chi(z) dz = 1$ . Then  $\{z : \rho_\chi(z) > 0\} \subseteq \{z \in \mathbb{R}^d : 0 < z_d < z_{d-1} < \dots < z_1 < \chi\}$ . With  $h$  and  $\rho_\chi$  defined, let  $f$  be the convolution of  $h$  with  $\rho_\chi$ , namely,

$$f(x) := \int_{\mathbb{R}^d} h(x+z)\rho_\chi(z) dz. \quad (75)$$

For a normal direction span-based method with  $K$  iterations, we consider the problem below.

**Problem 1.** *Let the problem be of the form (2), in which  $d = K + 1$  and  $\mathcal{C} = B(0, D_1) := \{x \in \mathbb{R}^d : \|x\|_2 \leq D_1\}$ , and the preference relation  $\preceq$  is realized by  $f$  in (75), namely,  $x \preceq y$  if and only if  $f(x) \leq f(y)$ . Let the initial iterate  $x_1$  be the origin  $0$  of  $\mathbb{R}^d$ .*

There are some basic regularity properties of  $f$ .

**Lemma 4.20.** *Consider  $f$  defined in (75). The function  $f$  is convex and there exists  $L > 0$  such that  $f$  is  $L$ -smooth on  $\mathcal{C}$ . For every point  $x \in \mathcal{C}$ ,  $\frac{1}{\sqrt{d}} \leq \|\nabla f(x)\|_2 \leq 1$ .*

*Proof.* Since  $h$  is convex and 1-Lipschitz, and since  $\rho_\chi$  is a nonnegative kernel with total mass 1, standard properties of convolution imply that  $f$  is also convex and 1-Lipschitz. Moreover, note that the convolution can be rewritten as  $f(x) = \int_{\mathbb{R}^d} h(y)\rho_\chi(y-x) dy$ . From standard properties of convolution,  $f$  is  $C^\infty$ . In addition, since  $h$  is globally Lipschitz and  $\rho_\chi$  is smooth and compactly supported, the second-order derivatives of  $f$  are bounded on  $\mathbb{R}^d$ . Hence,  $f$  is  $L$ -smooth for some  $L > 0$ .

At every point where  $h$  is differentiable, its gradient is one of the basis vectors  $\{e_1, \dots, e_d\}$  that correspond to a maximizing coordinate. Since  $h$  is Lipschitz, it is differentiable almost everywhere, and differentiating (75) under the integral sign yields  $\nabla f(x) \in \text{conv}\{e_1, \dots, e_d\}$  for all  $x \in \mathbb{R}^d$ . Hence  $\nabla f(x)$  is a probability vector. Therefore, for all  $x \in \mathbb{R}^d$

$$\frac{1}{\sqrt{d}} \leq \|\nabla f(x)\|_2 \leq 1, \quad (76)$$

and in particular  $\nabla f(x) \neq 0$  everywhere in  $\mathcal{C}$ . □

Lemma 4.21 below shows that Problem 1 belongs to the class of problems we have considered in this section.

**Lemma 4.21.** *Consider Problem 1. The constraint set  $\mathcal{C}$  is nonempty, closed, and convex, and the preference relation  $\preceq$  is convex, plateau-free, and regular at every  $x \in \mathcal{C}$ . The optimal solution set is on the boundary of  $\mathcal{C}$  and  $\text{dist}(x_1, \mathcal{X}^*) = D_1$ .*

*Proof.* Recall the properties of  $f$  established in Lemma 4.20. Convexity of  $\preceq$  follows from convexity of  $f$ . Since  $\nabla f(x) \neq 0$  at every point, each nonoptimal function level set  $\{u : f(u) = f(x)\}$ , equivalently each induced preference level set  $\mathcal{L}_x$ , is a smooth hypersurface. In particular it has empty interior, so the preference is plateau-free. Also, because  $f$  is  $L$ -smooth for some  $L > 0$  and  $\nabla f(x) \neq 0$  for every  $x \in \mathcal{C}$ , Lemma 2.10 implies that  $\preceq$  is regular at every  $x \in \mathcal{C}$ . Finally,  $f$  has no interior minimizer on  $\mathcal{C} = B(0, D_1)$ , since any interior minimizer of the differentiable convex function  $f$  would satisfy  $\nabla f(x) = 0$ , which is impossible. Hence every optimal point lies on  $\partial B(0, D_1)$ , and therefore  $\text{dist}(x_1, \mathcal{X}^*) = \text{dist}(0, \mathcal{X}^*) = D_1$ .  $\square$

Furthermore, Lemma 4.22 below shows a certain pattern of zero components of normal directions that will be key in the lower bound analysis.

**Lemma 4.22.** *Consider Problem 1. For all  $\bar{x} \in \mathcal{C}$  such that  $\bar{x}_r = \bar{x}_{r+1} = \dots = \bar{x}_d = 0$  for some  $r < d$ , the corresponding normal directions satisfy  $(n_{\bar{x}})_j = 0$  for all  $j > r$ .*

*Proof.* For every  $j > r$  and every  $y$  satisfying  $\rho_{\chi}(y) > 0$  we have  $(\bar{x} + y)_r = \bar{x}_r + y_r = y_r > y_j = \bar{x}_j + y_j = (\bar{x} + y)_j$ . Here the inequality is due to the support of  $\rho_{\chi}(\cdot)$  and the condition (74). Therefore, for almost every such  $y$ , the  $j$ th component of  $\nabla h(\bar{x} + y)$  is zero. Computing the derivative of (75) yields for all  $j > r$ ,  $(\nabla f(\bar{x}))_j = 0$ . By Lemma 4.20, we have  $\|\nabla f(x)\|_2 > 0$  for all  $x \in \mathcal{C}$ . Since the preference relation is realized by the differentiable function  $f$ , the outward normal satisfies  $n_x = \frac{\nabla f(x)}{\|\nabla f(x)\|_2}$  for all  $x \in \mathcal{C}$ . Combining this with  $(\nabla f(\bar{x}))_j = 0$  for  $j > r$  gives  $(n_{\bar{x}})_j = 0$  for all  $j > r$ , which completes the proof.  $\square$

Below we show the lower bound for normal direction span-based methods on Problem 1.

**Theorem 4.23** (Lower bound of normal direction span-based methods). *For Problem 1, every  $K$ -step normal direction span-based method (Definition 4.1) starting from  $x_1 = z_1 = 0$  satisfies*

$$\Delta_{\text{LS}}(\hat{x}) \geq \frac{3 \text{dist}(x_1, \mathcal{X}^*)}{4\sqrt{K+1}}. \quad (77)$$

*Proof.* We first prove that for the iterates generated by any normal direction span-based method on this instance: for every  $t = 1, \dots, K+1$ ,

$$x_t, z_t \in \text{span}\{e_1, \dots, e_{t-1}\}. \quad (78)$$

The base case  $t = 1$  is immediate since  $z_1 = x_1 = 0$ . Suppose (78) holds for every  $i = 1, \dots, t$ . Then  $x_t \in \text{span}\{e_1, \dots, e_{t-1}\}$ , so its coordinates  $t, \dots, d$  are zero. By Lemma 4.22, we have  $n_t \in \text{span}\{e_1, \dots, e_t\}$ . More generally, for every  $i \leq t$ , the induction hypothesis gives  $x_i \in \text{span}\{e_1, \dots, e_{i-1}\}$ , so Lemma 4.22 implies  $n_i \in \text{span}\{e_1, \dots, e_i\} \subseteq \text{span}\{e_1, \dots, e_t\}$ . Also, for every  $i \leq t$ , the induction hypothesis implies that both  $z_i$  and  $x_i$  belong to  $\text{span}\{e_1, \dots, e_{t-1}\}$ , hence so does the correction direction  $x_i - z_i$ . The update rule (35) therefore yields  $z_{t+1} \in \text{span}\{e_1, \dots, e_t\}$ . Since  $x_{t+1} = \Pi_{\mathcal{C}}(z_{t+1})$  is a radial rescaling of  $z_{t+1}$ , we also have  $x_{t+1} \in \text{span}\{e_1, \dots, e_t\}$ . This completes the induction and proves (78).

In particular,  $(x_t)_d = 0$  for every  $t$ . For any  $z$  satisfying  $\rho_\chi(z) > 0$ , due to (74) we have  $z_d > 0$ , and therefore  $h(x_t + z) \geq (x_t + z)_d = z_d > 0$ . Averaging over  $z$  gives

$$f(x_t) > 0 \quad \text{for every } t = 1, \dots, K + 1. \quad (79)$$

Hence any output  $\hat{x} \in \{x_1, \dots, x_{K+1}\}$  satisfies  $f(\hat{x}) > 0$ .

Now set  $y := (-\frac{D_1}{\sqrt{d}}, \dots, -\frac{D_1}{\sqrt{d}})$ . Then  $\|y\|_2 = D_1$ , so  $y \in \mathcal{C}$ . If  $\rho_\chi(z) > 0$ , then  $0 < z_i < \chi = \frac{D_1}{4\sqrt{d}}$  for every  $i$ , and therefore

$$h(y + z) = \max_{1 \leq i \leq d} (y_i + z_i) \leq -\frac{D_1}{\sqrt{d}} + \chi = -\frac{3D_1}{4\sqrt{d}}.$$

By (75), we obtain  $f(y) \leq -\frac{3D_1}{4\sqrt{d}}$ . If  $f^* := \min_{u \in \mathcal{C}} f(u)$ , then  $f^* \leq f(y)$ , and together with (79) this yields

$$f(\hat{x}) - f^* \geq \frac{3D_1}{4\sqrt{d}}. \quad (80)$$

Finally, because  $f$  is globally 1-Lipschitz, for every  $u \in \mathcal{L}_{\hat{x}}$  and every  $x^* \in \mathcal{X}^*$  we have  $f(\hat{x}) - f^* = f(u) - f(x^*) \leq \|u - x^*\|_2$ . Taking the infimum over  $u \in \mathcal{L}_{\hat{x}}$  and  $x^* \in \mathcal{X}^*$  gives  $\Delta_{\text{LS}}(\hat{x}) \geq f(\hat{x}) - f^*$ . Combining this with (80) and using  $d = K + 1$  and  $\text{dist}(x_1, \mathcal{X}^*) = D_1$  proves  $\Delta_{\text{LS}}(\hat{x}) \geq \frac{3D_1}{4\sqrt{K+1}}$ , as claimed.  $\square$

Theorem 4.23 shows that the  $O(\text{dist}(x_1, \mathcal{X}^*)/\sqrt{K})$  dependence in Theorem 4.4 is sharp for normal direction span-based methods, and that Theorem 4.19 is sharp up to the logarithmic factor in (67). Given that this lower bound remains for exact normal direction oracles, one may conclude that if the normal directions are estimated with some error (for example, using Algorithm 2), then the  $O(\text{dist}(x_1, \mathcal{X}^*)/\sqrt{K})$  dependence is still unimprovable.

This hard instance is inspired by the classic lower-bound constructions for first-order methods on convex Lipschitz functions; see, e.g., [28]. In both cases, the key mechanism is a zero chain: each query of normal direction reveals at most one new coordinate. Our results differ in two ways. First, the guarantee here is formulated in terms of the level-set optimality gap  $\Delta_{\text{LS}}$  rather than in terms of a function-value gap. Second, the classical nonsmooth hard instance does not automatically fit our regularity assumption of preference level sets. The convolution (75) using kernel  $\rho_\chi$  preserves the zero-chain structure while producing a preference relation whose level sets are regular.

This instance also satisfies the local growth condition. Let  $L$  be a smoothness constant of  $f$  in  $\mathcal{C}$ . By Lemma 4.20,  $\|\nabla f(x)\|_2 \geq 1/\sqrt{d}$  for all  $x \in \mathcal{C}$ . This is an instance of Example 4.13, so Condition 4.7 holds with, for example,  $(\gamma_1, \gamma_2) = (1, \frac{1}{L\sqrt{d}})$ .

## 5 Summary, remarks and future directions

This paper proposes a function-free optimization framework for optimizing under preference relations. Instead of assuming an application-given objective function, we study the preference relation and its level-set geometry directly. We introduced the level-set optimality gap as a measure of optimality and the regularity radius as a measure of stationarity. Under a regularity condition on the preference relation, we showed how to estimate normal directions from pairwise comparisons with near-optimal comparison complexity. Under convexity and a local growth condition on the regularity radius, these normal direction estimates lead to NDD and adaNDD, whose dependence on the number of normal direction steps nearly matches lower bounds for normal-direction span-based methods up to logarithmic factors. The resulting comparison complexity is  $\tilde{O}(dD^2/\epsilon^2)$ .

Although the paper is established under Euclidean space  $\mathcal{X}$ , we expect the same theory holds for any finite-dimensional real Hilbert space. The main objects used in the paper, including preference level sets, distances, normal directions, angles, separating halfspaces and orthonormal bases, depend only on the inner product and its induced norm, and hold also under real Hilbert space.

Several questions remain open for future research. One direction is to study NDD and adaNDD for preference relations without convexity assumptions. More broadly, it would be interesting to develop a nonconvex function-free optimization theory that does not rely on a particular objective function. For example, one may ask under what comparison-level assumptions and within how many comparisons an algorithm can guarantee a point that is preferred to some feasible point with sufficiently small regularity radius.

Another direction is to study comparison complexity lower bounds for function-free optimization. The lower bounds established in this paper are for normal direction estimation and for the number of normal direction steps among normal-direction span-based methods. It remains open to determine the minimax number of pairwise comparisons required for function-free optimization, and develop a method that achieves this comparison complexity.

## Statements and Declarations

This work was partially supported by ONR Award #N00014-25-1-2088. The authors have no competing interests to declare that are relevant to the content of this article.

## References

- [1] Charles Audet and J. E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.
- [2] El Houcine Bergou, Eduard Gorbunov, and Peter Richtárik. Stochastic three points method for unconstrained smooth minimization. *SIAM Journal on Optimization*, 30(4):2726–2749, 2020.
- [3] HanQin Cai, Daniel McKenzie, Wotao Yin, and Zhenliang Zhang. A one-bit, comparison-based gradient estimator. *Applied and Computational Harmonic Analysis*, 60:242–266, 2022.
- [4] Peter Chen, Xi Chen, Wotao Yin, and Tianyi Lin. ComPO: Preference alignment via comparison oracles. In *Advances in Neural Information Processing Systems*, volume 38, 2025.
- [5] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [6] Taha El Bakkali, El Mahdi Chayti, and Omar Saadi. Nonsmooth optimization with zeroth order comparison feedback. *arXiv preprint arXiv:2602.05622*, 2026.
- [7] Herbert Federer. *Geometric Measure Theory*, volume 153 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, Heidelberg, New York, 1969.
- [8] Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine Learning*, 89(1–2):123–156, 2012.
- [9] Daniel Golovin, John Karro, Greg Kochanski, Chansoo Lee, Xingyou Song, and Qiuyi Zhang. Gradientless descent: High-dimensional zeroth-order optimization. In *International Conference on Learning Representations*, 2020.
- [10] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1282–1291. PMLR, 2017.

- [11] Serge Gratton, Clément W. Royer, Luís N. Vicente, and Zaikun Zhang. Direct search based on probabilistic descent. *SIAM Journal on Optimization*, 25(3):1515–1541, 2015.
- [12] Elad Hazan, Kfir Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. In *Advances in Neural Information Processing Systems*, volume 28, 2015.
- [13] Robert Hooke and T. A. Jeeves. “Direct Search” solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
- [14] Kevin G. Jamieson, Robert D. Nowak, and Benjamin Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, volume 25, 2012.
- [15] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 944–953. ACM, 2020.
- [16] Mustafa O. Karabag, Cyrus Neary, and Ufuk Topcu. Smooth convex optimization using sub-zeroth-order oracles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3815–3822, 2021.
- [17] Leonid G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.
- [18] Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.
- [19] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.
- [20] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 1049–1065. IEEE, 2015.
- [21] Kfir Y. Levy. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016.
- [22] Kfir Y. Levy. Online to offline conversions, universality and adaptive minibatch sizes. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [23] Aleksandr Lobanov, Alexander Gasnikov, and Andrei Krasnov. Acceleration exists! optimization problems when oracle can only compare objective function values. In *Advances in Neural Information Processing Systems*, volume 37, pages 21428–21460, 2024.
- [24] Ken I. M. McKinnon. Convergence of the Nelder–Mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.
- [25] Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7555–7564. PMLR, 2021.
- [26] Ryan Murray, Brian Swenson, and Soumya Kar. Revisiting normalized gradient descent: Fast evasion of saddle points. *IEEE Transactions on Automatic Control*, 64(11):4818–4824, 2019.
- [27] John A. Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [28] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.
- [29] Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2nd edition, 2018.
- [30] Francesco Orabona. Normalized gradients for all. *arXiv preprint arXiv:2308.05621*, 2023.

- [31] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- [32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.
- [33] Yuxuan Ren, Abhishek Roy, and Shiqian Ma. Riemannian dueling optimization. *arXiv preprint arXiv:2603.00023*, 2026.
- [34] Aadirupa Saha, Vitaly Feldman, Yishay Mansour, and Tomer Koren. Faster convergence with MultiWay preferences. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 433–441. PMLR, 2024.
- [35] Aadirupa Saha, Tomer Koren, and Yishay Mansour. Dueling convex optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9245–9254. PMLR, 2021.
- [36] Aadirupa Saha, Tomer Koren, and Yishay Mansour. Dueling convex optimization with general preferences. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 52552–52564. PMLR, 2025.
- [37] Naum Zuselevich Shor. *Minimization Methods for Non-Differentiable Functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 1985.
- [38] Zhiwei Tang, Dmitry Rybin, and Tsung-Hui Chang. Zeroth-order optimization meets human feedback: Provable learning via ranking oracles. In *International Conference on Learning Representations*, 2024.
- [39] Xiwen Tao, Chenyi Zhang, Helin Wang, Yexin Zhang, and Tongyang Li. Gradient testing and estimation by comparisons. *arXiv preprint arXiv:2405.11454v2 (Version 2)*, 2026.
- [40] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.
- [41] Maegan Tucker, Ellen Novoseller, Claudia Kann, Yanan Sui, Yisong Yue, Joel W. Burdick, and Aaron D. Ames. Preference-based learning for exoskeleton gait optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2351–2357. IEEE, 2020.
- [42] Luís N. Vicente. Worst case complexity of direct search. *EURO Journal on Computational Optimization*, 1(1–2):143–153, 2013.
- [43] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The K-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.
- [44] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.
- [45] Chenyi Zhang and Tongyang Li. Comparisons are all you need for optimizing smooth functions. *arXiv preprint arXiv:2405.11454v1 (Version 1)*, 2024.
- [46] Zixi Zhao, Aliya Ahmadi, Caleb Hoover, Logan Grado, Nicholas Peterson, Xinran Wang, David Freeman, Thomas Murray, Andrew Lamperski, David Darrow, and Theoden I. Netoff. Optimization of spinal cord stimulation using Bayesian preference learning and its validation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:1987–1997, 2021.