Approximating Convex Hulls via Range Queries

Thomas Schibler
University of California, Santa Barbara, CA, USA

Jie Xue
New York University Shanghai, China

Jiumu Zhu
New York University Shanghai, China

Abstract

Recently, motivated by the rapid increase of the data size in various applications, Monemizadeh [APPROX'23] and Driemel, Monemizadeh, Oh, Staals, and Woodruff [SoCG'25] studied geometric problems in the setting where the only access to the input point set is via querying a range-search oracle. Algorithms in this setting are evaluated on two criteria: (i) the number of queries to the oracle and (ii) the error of the output. In this paper, we continue this line of research and investigate one of the most fundamental geometric problems in the oracle setting, i.e., the convex hull problem.

Let P be an unknown set of points in $[0,1]^d$ equipped with a range-emptiness oracle. Via querying the oracle, the algorithm is supposed to output a convex polygon $C \subseteq [0,1]^d$ as an estimation of the convex hull $\mathcal{CH}(P)$ of P. The error of the output is defined as the volume of the symmetric difference $C \oplus \mathcal{CH}(P) = (C \setminus \mathcal{CH}(P)) \cup (\mathcal{CH}(P) \setminus C)$. We prove tight and near-tight tradeoffs between the number of queries and the error of the output for different variants of the problem, depending on the type of the range-emptiness queries and whether the queries are non-adaptive or adaptive.

- **Orthogonal emptiness queries in d-dimensional space:**
  We show that the minimum error a deterministic algorithm can achieve with q queries is $\Theta(q^{-1/d})$ if the queries are non-adaptive, and $\Theta(q^{-1/(d-1)})$ if the queries are adaptive. In particular, in 2D, the bounds are $\Theta(1/\sqrt{q})$ and $\Theta(1/q)$ for non-adaptive and adaptive queries, respectively.
- **Halfplane emptiness queries in 2D:**
  We show that the minimum error a deterministic algorithm can achieve with q queries is $\Theta(1/\sqrt{q})$ if the queries are non-adaptive, and $\widetilde{\Theta}(1/q^2)$ if the queries are adaptive. Here $\widetilde{\Theta}(\cdot)$ hides logarithmic factors.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms

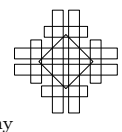
Keywords and phrases convex hull, range searching

Digital Object Identifier 10.4230/LIPIcs.SoCG.2026.89

Related Version *Full Version*: https://arxiv.org/abs/2603.20943

1 Introduction

Classic algorithms are designed to compute solutions to a problem instance by processing the *entire* input dataset. These algorithms can suffer two potential drawbacks. The first drawback concerns *efficiency*. As all of the input data has to be received and examined, the time complexity of such algorithms (even the most efficient ones) is at least linear in the input size. However, due to the rapidly growing size of the datasets involved in real-world applications nowadays, linear running time is already not satisfactory in many scenarios. The second drawback regards *data privacy*. For security reasons, sometimes the users would like to keep their data private and therefore cannot directly provide the exact dataset to the algorithm. Instead, they can only provide partial and implicit information of the dataset and ask the algorithm to give useful results based on the information provided. In this situation, the classic algorithms no longer work due to lack of full information of the dataset.



Motivated by removing these drawbacks, researchers have studied algorithms in the *oracle model*. In this model, the algorithm does not have direct access to the input dataset. Instead, a provided oracle built on the dataset can answer certain queries about the dataset. The algorithm is required to compute a good solution by performing queries to the oracle. Algorithms in this setting are usually evaluated on two criteria: (i) the number of queries to the oracle and (ii) the error of the output.

For geometric problems, various types of oracles have been considered in the literature [14, 16, 15, 17, 22]. When the input dataset is a set P of points in a Euclidean space $\mathbb{R}^d$, a natural type is the *range-search* oracle. A query to a range-search oracle is a *range* Q in the space $\mathbb{R}^d$ of a specific shape, and the oracle will return certain information about the points in $P \cap Q$. For example, a *range-counting* oracle returns $|P \cap Q|$ [1, 12], a *range-emptiness* oracle returns whether $P \cap Q = \emptyset$ or not [19, 26], a *range-reporting* oracle returns the set $P \cap Q$ itself [6, 23], etc. Range-search oracles have the following advantages. First, range search can usually be implemented very efficiently. As a fundamental topic in Computational Geometry, range search has been extensively studied over decades and many efficient data structures have been proposed for various range queries [4, 5, 9, 12, 13, 21]. Furthermore, most types of range search (except range reporting) do not reveal the exact data points inside the query range Q, and therefore data privacy is well guaranteed. Recently, Monemizadeh [22] and the authors of [17] studied multiple geometric problems with range-search oracles, following earlier work of Czumaj and Sohler [16] and the authors of [15]. Problems considered include facility location [22], clustering [16], Euclidean minimum spanning tree [15, 17], earth mover distance [17], etc. They show that for all these problems, one can obtain nontrivial approximation solutions via a small number of range-search queries.

In this paper, we continue this line of research and investigate one of the most fundamental geometric problems, the *convex hull* problem [7, 8, 10, 11, 24, 27], in the range-search oracle model. In the convex hull problem, the goal is to compute the *convex hull* of a set P of points in $\mathbb{R}^d$, denoted by $\mathcal{CH}(P)$, which by definition is the smallest convex body in $\mathbb{R}^d$ containing P. We study the problem with the simplest type of range-search oracles, i.e., *range-emptiness* oracles. We are interested in finding a convex body C^* as an approximation of $\mathcal{CH}(P)$ via a small number of range-emptiness queries. To this end, we need a measure for the error of the approximation. The most natural measure one can use is the relative error $\|C^* \oplus \mathcal{CH}(P)\| \,/\, \|\mathcal{CH}(P)\|$, where $\|\cdot\|$ denotes the volume and $C^* \oplus \mathcal{CH}(P) = (C^* \backslash \mathcal{CH}(P)) \cup (\mathcal{CH}(P) \backslash C^*)$ is the symmetric difference between C^* and $\mathcal{CH}(P)$. Unfortunately, one can easily see that it is impossible to approximate the convex hull with any bounded relative error no matter how many range-emptiness queries the algorithm performs[1]. Therefore, we shall instead consider the additive error $\|C^* \oplus \mathcal{CH}(P)\|$. Clearly, the additive error depends on the extent measure of P. So we need an extra normalization assumption: we require all points in P to lie in the unit hypercube $[0,1]^d$. Below we formulate the problem to be studied.

Approximate Convex Hull in $\mathbb{R}^d$ via range queries

Input: A (black-box) range-emptiness oracle $\mathcal{O}$ on a set P of points in $[0,1]^d$
Output: A convex body C^* that approximates $\mathcal{CH}(P)$

[1] Even in $\mathbb{R}^1$, there is no way to check whether P only contains one point, in which case $\|\mathcal{CH}(P)\| = 0$, or P contains at least two points, in which case $\|\mathcal{CH}(P)\| > 0$, via range-emptiness queries.

Regarding the above problem, a natural question concerns the tradeoff between the number of queries and the (additive) error of the output: if the algorithm is allowed to perform q queries to the oracle $\mathcal{O}$, what is the minimum error it can achieve (in the worst case)? The answer to this question depends on the following two features of the queries.

- **Shape of the query ranges.** Range queries with different shapes might behave very differently. The most commonly used queries are orthogonal queries [12], where the query ranges are axis-parallel rectangles/boxes. Besides these, well-studied range queries include halfplane/halfspace queries [2, 18], simplex queries [20], semi-algebraic queries [5, 26], etc.
- **Adaptivity of the queries.** In the non-adaptive query model, the algorithm must make all its queries at once, and then make its estimation based on the batched answer. In the adaptive query model, the algorithm is allowed to make queries at any time and the oracle will provide the answer immediately. In particular, the next query can be made after seeing the answers of the previous queries.

As the main contribution of this paper, we prove tight and near-tight tradeoffs between the number of queries and the error of the output for Approximate Convex Hull with orthogonal emptiness queries in $\mathbb{R}^d$ for any fixed d and halfplane emptiness queries in $\mathbb{R}^2$, in both non-adaptive and adaptive query models.

- **Orthogonal queries.** We show the minimum error a deterministic algorithm can achieve for Approximate Convex Hull in $\mathbb{R}^d$ with q orthogonal emptiness queries is $\Theta(q^{-1/d})$ if the queries are non-adaptive and is $\Theta(q^{-1/(d-1)})$ if the queries are adaptive. In particular, in 2D, the bounds are $\Theta(1/\sqrt{q})$ and $\Theta(1/q)$ for non-adaptive and adaptive queries, respectively.
- **Halfplane queries.** We show the minimum error a deterministic algorithm can achieve for Approximate Convex Hull in $\mathbb{R}^2$ with q halfplane emptiness queries is $\Theta(1/\sqrt{q})$ if the queries are non-adaptive and is $\widetilde{\Theta}(1/q^2)$ if the queries are adaptive[2].

Table 1 summarizes the tradeoffs we prove in this paper. All of our algorithms are deterministic and have offline time complexity (outside the oracle) polynomial in q. Our lower bounds similarly hold for deterministic algorithms.

Table 1 Minimum error one can achieve for Approximate Convex Hull with q queries.

Query shape	Space	Query type	Upper bound	Lower bound	Source
Orthogonal	$\mathbb{R}^d$	Non-adaptive	$O(q^{-1/d})$	$\Omega(q^{-1/d})$	Theorems 2 and 8
		Adaptive	$O(q^{-1/(d-1)})$	$\Omega(q^{-1/(d-1)})$	Theorems 12 and 13
Halfplane	$\mathbb{R}^2$	Non-adaptive	$O(1/\sqrt{q})$	$\Omega(1/\sqrt{q})$	Theorems 17 and 22
		Adaptive	$\widetilde{O}(1/q^2)$	$\Omega(1/q^2)$	Theorems 28 and 29

Related work. Both convex hulls and range search have been extensively studied in the literature. See [3, 25] for surveys of these topics. Problems related to convex hulls have also been considered in oracle models prior to this paper. For example, the celebrated work of Chazelle, Liu, and Magen [14] considered the problem of approximating the volume of the convex hull in 2D and 3D via a sampling oracle, which can uniformly sample an input point.

[2] The notation $\widetilde{\Theta}(\cdot)$ hides factors logarithmic in q.

Czumaj and Sohler [16] studied the problem of testing convex position via range queries. Here the goal is to check whether one can remove an ε-fraction of the points from the input point set to make it in convex position. The oracle used in [16] is slightly stronger than range-emptiness oracles – it can report one point in the query range.

2 Preliminaries

Basic notations. For a number $n \in \mathbb{N}$, we write $[n] = \{1, \ldots, n\}$. For a convex body C in $\mathbb{R}^d$, we denote by $\|C\|$ its volume. The notation $|\cdot|$ is used with different meanings depending on the context. For a segment σ, we use $|\sigma|$ to denote the length of σ. For an angle α, $|\alpha| \in [0, 2\pi)$ is the magnitude of α. Furthermore, for a vector $\vec{v}$, $|\vec{v}|$ denotes its magnitude.

Vectors and halfspaces. For a halfspace $H \subset \mathbb{R}^d$, denote by ∂H its bounding hyperplane. (More generally, ∂C denotes the boundary of any convex body C.) The *normal vector* of a halfspace H is a unit vector $\vec{v} \in \mathbb{S}^{d-1}$ perpendicular to ∂H so that for any $q \in \partial H$, $H = \{p \in \mathbb{R}^d : \langle p, \vec{v}\rangle \leq \langle q, \vec{v}\rangle\}$.

3 Approximating convex hulls via orthogonal queries

In this section, we present our results for Approximate Convex Hull in $\mathbb{R}^d$ via orthogonal emptiness queries. For non-adaptive queries (Section 3.1), the algorithm is very simple, while the lower bound proof is nontrivial and interesting. For adaptive queries (Section 3.2), the algorithm is more technical and the lower bound proof is simpler.

3.1 Non-adaptive orthogonal queries

We first present our algorithmic result with non-adaptive orthogonal emptiness queries. Let $P \subseteq [0,1]^d$ be an unknown set of points and $\mathcal{O}$ be the orthogonal emptiness oracle on P. For an axis-parallel box $\square$ in $\mathbb{R}^d$, let Query($\square$) denote the output of $\mathcal{O}$ when queried with $\square$, which is yes if $P \cap \square = \emptyset$ and is no if $P \cap \square \neq \emptyset$.

Algorithm 1 NonAdaptiveOrthogonal(q).

1: $r \leftarrow \lfloor q^{1/d} \rfloor$
2: $\Gamma \leftarrow$ Partition($[0,1]^d, r$)
3: $A_\square \leftarrow$ Query($\square$) for all $\square \in \Gamma$
4: $\Gamma_1 = \{\square \in \Gamma : A_\square = \text{no}\}$
5: $C^* \leftarrow \mathcal{CH}(\bigcup_{\square \in \Gamma_1} \square)$
6: **return** C^*

Our algorithm for approximating $\mathcal{CH}(P)$ is very simple (presented in Algorithm 1) and is similar to the algorithm of Czumaj and Sohler [16] for testing convex position. Let $r = \lfloor q^{1/d} \rfloor$. In line 2, the sub-routine Partition($[0,1]^d, r$) partitions $[0,1]^d$ evenly into r^d *cells* each of which is a hypercube of side-length $\frac{1}{r}$; let Γ be the set of the r^d cells. Then we query $\mathcal{O}$ with the boxes in Γ, and let $A_\square =$ Query($\square$) for $\square \in \Gamma$. Define Γ_1 as the set of cells $\square \in \Gamma$ with $A_\square =$ no (which are just the cells containing at least one point in P). Finally, the algorithm simply returns $C^* = \mathcal{CH}(\bigcup_{\square \in \Gamma_1} \square)$. Clearly, the number of queries to $\mathcal{O}$ is at most q, and they are non-adaptive as the algorithm performs them at the same time.

To bound the error of our algorithm, we need the following lemma about the volume of the Minkowski sum of a convex body and a ball; we include a proof in the full version. For two convex bodies X and Y in $\mathbb{R}^d$, denote by $X + Y = \{x + y : x \in X \text{ and } y \in Y\}$ their Minkowski sum.

▶ **Lemma 1.** *For constant d, let $C \subseteq [0,1]^d$ be a convex body and B_δ be the ball with radius $\delta \in [0,1]$ centered at the origin of $\mathbb{R}^d$. Then $\|(C + B_\delta)\backslash C\| = O(\delta)$ for all $\delta \in [0,1]$.*

By construction, we have $\mathcal{CH}(P) \subseteq C^*$. On the other hand, we observe that $C^* \subseteq \mathcal{CH}(P) + B$, where B is the ball in $\mathbb{R}^d$ centered at the origin with radius $\frac{d}{r}$. As C^* is a polytope, it suffices to show that every vertex of C^* lies in $\mathcal{CH}(P) + B$. A vertex z of C^* is a corner of some $\square \in \Gamma_1$. Since $A_\square = \mathsf{no}$, there exists a point $p \in P \cap \square$. The distance between z and p is at most $\frac{d}{r}$, as the side-length of $\square$ is $\frac{1}{r}$. Thus, $z \in p + B$ and $z \in \mathcal{CH}(P) + B$. It follows that $\|C^*\backslash\mathcal{CH}(P)\| \leq \|(\mathcal{CH}(P) + B)\backslash\mathcal{CH}(P)\|$. By Lemma 1, $\|(\mathcal{CH}(P) + B)\backslash\mathcal{CH}(P)\| = O(\frac{1}{r})$. So we have $\|C^*\backslash\mathcal{CH}(P)\| = O(\frac{1}{r}) = O(q^{-1/d})$.

▶ **Theorem 2.** *There exists an algorithm for* Approximate Convex Hull *in $\mathbb{R}^d$ that performs q non-adaptive orthogonal emptiness queries and has error $O(q^{-1/d})$.*

Interestingly, the above algorithm, while being very simple, is already the best one can hope. Specifically, we show that any deterministic algorithm for Approximate Convex hull with q non-adaptive orthogonal emptiness queries has error $\Omega(q^{-1/d})$.

Consider such an algorithm $\mathbf{A}$. Since the q queries $\mathbf{A}$ performs are non-adaptive, these queries are independent of the point set P (as well as the range-emptiness oracle). Let $\square_1, \ldots, \square_q$ be these queries, each of which is a box in $\mathbb{R}^d$. Without loss of generality, we may assume $\square_1, \ldots, \square_q \subseteq [0,1]^d$. Set $n = \lceil q^{1/d} \rceil + 1$ and $\delta = q^{-1/d}/c$ where c is a sufficiently large constant (which only depends on d). Define a sequence $H_0, H_1, \ldots, H_n$ of parallel hyperplanes in $\mathbb{R}^d$ where the equation of H_i is $x_1 + \cdots + x_d = 1 + \delta i$. For $i \in [n]$, we say a box $\square$ is *i-good* if $H_i \cap \square \neq \emptyset$ and $H_{i-1} \cap \square = \emptyset$. See figure 1.

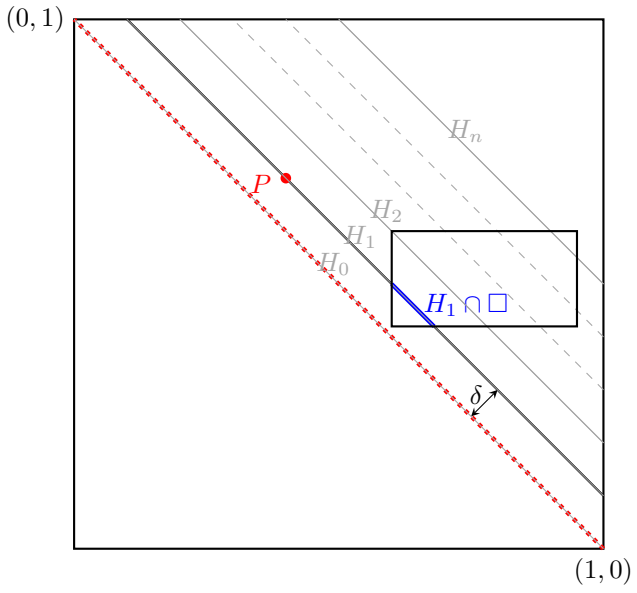


Figure 1 The lower bound construction in $d = 2$ dimensions for non-adaptive orthogonal queries with underlying points P (red) along the main diagonal H_0. The depicted query $\square$ is 1-good since its lower left corner lies between H_0 and H_1. The length of $H_1 \cap \square$ (blue) is $O(\delta^{d-1})$, so $\Omega(1/\delta^{d-1})$ 1-good queries are needed to cover H_1 in order to determine whether P contains a point on H_1. In total, $\Omega(n\delta^{d-1})$ i-good queries are needed to cover all $H_1, \cdots, H_n$.

▶ **Lemma 3.** *There exists $i \in [n]$ such that at most $\frac{q}{n}$ boxes in $\{\square_1, \ldots, \square_q\}$ are i-good.*

Proof. Observe that for different $i, j \in [n]$, no box in $\mathbb{R}^d$ can be both i-good and j-good. To see this, suppose $i < j$ without loss of generality. Let $\square$ be a j-good box. Then $H_j \cap \square \neq \emptyset$ and $H_{j-1} \cap \square = \emptyset$. So H_j and $\square$ are on the same side of H_{j-1}. If $i = j - 1$, then $H_i \cap \square = \emptyset$ and thus $\square$ is not i-good. Otherwise, $i \leq j - 2$. In this case, H_i and $\square$ are on the opposite sides of H_{j-1}, which implies $H_i \cap \square = \emptyset$ and $\square$ is not i-good. Therefore, any box in $\mathbb{R}^d$ can be i-good for at most one $i \in [n]$. Let t_i be the number of i-good boxes in $\{\square_1, \ldots, \square_q\}$. It follows that $\sum_{i=1}^{n} t_i \leq q$ and hence there exists $i \in [n]$ satisfying $t_i \leq q/n$. ◀

Let $i \in [n]$ such that at most $\frac{q}{n}$ boxes in $\{\square_1, \ldots, \square_q\}$ are i-good. Without loss of generality, assume $\square_1, \ldots, \square_t$ are i-good and $\square_{t+1}, \ldots, \square_q$ are not i-good, where $t \leq \frac{q}{n}$. Each $H_i \cap \square_j$ is a $(d-1)$-dimensional convex polytope, and let $\|H_i \cap \square_j\|'$ denote its $(d-1)$-dimensional volume.

▶ **Lemma 4.** *For all $j \in [t]$, $\|H_i \cap \square_j\|' = O(\delta^{d-1})$.*

Proof. Suppose $\square_j = [x_1^-, x_1^+] \times \cdots \times [x_d^-, x_d^+]$ and $\square_j' = [x_1^-, +\infty) \times \cdots \times [x_d^-, \infty)$. Since $\square_j$ is i-good, we have $H_i \cap \square_j \neq \emptyset$ and $H_{i-1} \cap \square_j = \emptyset$, which implies $1 + \delta i - \delta \leq x_1^- + \cdots + x_d^- \leq 1 + \delta i$. So the distance from the point $(x_1^-, \ldots, x_d^-) \in \mathbb{R}$ to H_i is $O(\delta)$. This implies $\|H_i \cap \square_j'\|' = O(\delta^{d-1})$, and thus $\|H_i \cap \square_j\|' = O(\delta^{d-1})$. ◀

▶ **Lemma 5.** $H_i \cap [0,1]^d \nsubseteq \bigcup_{j=1}^{t} \square_j$.

Proof. Equivalently, we prove that $H_i \cap [0,1]^d \nsubseteq \bigcup_{j=1}^{t} (H_i \cap \square_j)$. As $H_i \cap [0,1]^d$ and $H_i \cap \square_1, \ldots, H_i \cap \square_t$ are all $(d-1)$-dimensional convex polytopes, it suffices to show that $\|H_i \cap [0,1]^d\|' > \sum_{j=1}^{t} \|H_i \cap \square_j\|'$. Since $1 + \delta i = 1 + O(\frac{1}{c})$ and c is sufficiently large, $\|H_i \cap [0,1]^d\|'$ is lower bounded by a constant depending only on d. By Lemma 4, $\sum_{j=1}^{t} \|H_i \cap \square_j\|' = O(\delta^{d-1} t) = O(\frac{1}{c})$. Therefore, we have $\|H_i \cap [0,1]^d\|' > \sum_{j=1}^{t} \|H_i \cap \square_j\|'$, for sufficiently large c. ◀

Next, we construct two sets Z and Z' of points in $\mathbb{R}^d$. We include in Z the vertices of the polytope $H_{i-1} \cap [0,1]^d$. Furthermore, for each $j \in [q]$ such that $H_{i-1} \cap \square_j \neq \emptyset$, we include in Z a point $p_j \in H_{i-1} \cap \square_j$. Note that all points in Z lie on $H_{i-1} \cap [0,1]^d$ and we have $\mathcal{CH}(Z) = H_{i-1} \cap [0,1]^d$, so $\|\mathcal{CH}(Z)\| = 0$. To further construct Z', we pick a point $p \in (H_i \cap [0,1]^d) \backslash (\bigcup_{j=1}^{t} \square_j)$, which exists by Lemma 5. Then we define $Z' = Z \cup \{p\}$.

▶ **Lemma 6.** *For all $j \in [q]$, $Z \cap \square_j = \emptyset$ iff $Z' \cap \square_j = \emptyset$.*

Proof. If $p \notin \square_j$, then $Z \cap \square_j = Z' \cap \square_j$ and we are done. So assume $p \in \square_j$. Then $j \notin [t]$ by the choice of p. Thus, $\square_j$ is not i-good and we have either $H_i \cap \square_j = \emptyset$ or $H_{i-1} \cap \square_j \neq \emptyset$. But $H_i \cap \square_j \neq \emptyset$ since $p \in H_i \cap \square_j$. So we must have $H_{i-1} \cap \square_j \neq \emptyset$. By construction, we include in Z the point $p_j \in H_{i-1} \cap \square_j$, which implies $Z \cap \square_j \neq \emptyset$ and $Z' \cap \square_j \neq \emptyset$. ◀

The above lemma shows that the algorithm $\mathbf{A}$ cannot distinguish Z and Z', that is, it returns the same convex body when running on $P = Z$ and $P = Z'$.

▶ **Lemma 7.** $\|\mathcal{CH}(Z') \backslash \mathcal{CH}(Z)\| = \Omega(\delta)$.

Let C^* be the output of $\mathbf{A}$ when run on $P = Z$ or $P = Z'$. We have $\|\mathcal{CH}(Z') \backslash \mathcal{CH}(Z)\| \leq \|C^* \oplus \mathcal{CH}(Z)\| + \|C^* \oplus \mathcal{CH}(Z')\|$. Therefore, the above lemma implies either $\|C^* \oplus \mathcal{CH}(Z)\| = \Omega(\delta)$ or $\|C^* \oplus \mathcal{CH}(Z')\| = \Omega(\delta)$. As $\delta = \Theta(q^{-1/d})$, the error of $\mathbf{A}$ is $\Omega(q^{-1/d})$.

▶ **Theorem 8.** *Any deterministic algorithm for* Approximate Convex Hull *in $\mathbb{R}^d$ that performs q non-adaptive orthogonal emptiness queries has error $\Omega(q^{-1/d})$.*

3.2 Adaptive orthogonal queries

We now consider adaptive orthogonal emptiness queries, and show how to approximate $\mathcal{CH}(P)$ with error $O(q^{-1/(d-1)})$ via $O(q)$ queries. We shall define 2^d supersets of $\mathcal{CH}(P)$, one-to-one corresponding to the vectors in $\{-1,1\}^d$. Our algorithm independently approximates these supersets and then merges them together. For each $\vec{v} = (v_1, \ldots, v_d) \in \{-1,1\}^d$, let O_v be the orthant $\{(x_1, \ldots, x_d) : x_i v_i \geq 0 \text{ for all } i \in [d]\}$. We define $\mathcal{CH}_{\vec{v}}(P) = (\mathcal{CH}(P) + O_{-\vec{v}}) \cap [0,1]^d$. The following observation relates $\mathcal{CH}(P)$ with the supersets $\mathcal{CH}_{\vec{v}}(P)$ (proof in full version).

▶ **Lemma 9.** $\mathcal{CH}(P) = \bigcap_{\vec{v} \in \{-1,1\}^d} \mathcal{CH}_{\vec{v}}(P)$.

Algorithm 2 presents how to approximate $\mathcal{CH}_{\vec{v}}(P)$ for each $\vec{v} \in \{-1,1\}^d$. It essentially "sandwiches" the hull between two convex boundaries, refining the estimate at each iteration. Figure 2 illustrates one such refinement step. The subroutine SUBDIVIDE(R) evenly partitions the hypercube R into 2^d smaller hypercubes (each of which has side-length half of the side-length of R) and returns the set of these 2^d hypercubes. For a hypercube R and a vector $\vec{v} \in \{-1,1\}^d$, we define $\mathsf{cor}(R, \vec{v}) = \arg\max_{p \in R} \langle p, \vec{v} \rangle$ as the corner of R in the direction $\vec{v}$. The algorithm starts with a set $\mathcal{R}$ of hypercubes which initially consists of only $[0,1]^d$, and runs in $\lceil \frac{\log q}{d-1} \rceil$ rounds. In each round, we perform emptiness queries for the hypercubes in $\mathcal{R}$, and let $\mathcal{R}' \subseteq \mathcal{R}$ consist of the nonempty ones. Then define $U = \mathcal{CH}_{\vec{v}}(\{\mathsf{cor}(R, \vec{v}) : R \in \mathcal{R}'\})$ and $L' = \mathcal{CH}_{\vec{v}}(\{\mathsf{cor}(R, -\vec{v}) : R \in \mathcal{R}'\})$. Let $L = L' \setminus \partial L'$ be the interior of L'. This is simply to ensure that $\partial \mathcal{CH}_{\vec{v}}(P)$ is contained in $U \setminus L$. We then take all hypercubes in $\mathcal{R}$ that intersect $U \setminus L$, partition them using the subroutine SUBDIVIDE, and let the new $\mathcal{R}$ be the set of the resulting smaller hypercubes. (Technically, we only need to consider the nonempty queries in $\mathcal{R}'$, but this is equivalent in the worst case, and this way nicely ensures that our queries continue to cover all of $\partial CH_{\vec{v}}(P)$). Finally, we return the convex body U in the last iteration as an approximation of $\mathcal{CH}_{\vec{v}}(P)$.

Algorithm 2 ADAPTIVEORTHOGONAL($q, \vec{v}$).

```
1: R ← {[0,1]^d}
2: for i = 1 to ⌈log q / (d−1)⌉ do
3:     R' ← {R ∈ R : QUERY(R) = no}
4:     U ← CH_v({cor(R, v) : R ∈ R'})
5:     L' ← CH_v({cor(R, −v) : R ∈ R'})
6:     L ← L' \ ∂L'
7:     R'' ← {R ∈ R : R ∩ (U \ L) ≠ ∅}
8:     R ← ⋃_{R∈R''} SUBDIVIDE(R)
9: return U
```

At a high level, the algorithm maintains an upper and lower bound (the sets U and L respectively) on the unknown convex body $\mathcal{CH}_{\vec{v}}(P)$. At each iteration of the loop, we reduce the error volume $\|(U \setminus \mathcal{CH}_{\vec{v}}(P)) \cap [0,1]^d\|$ by half. Let $t = \lceil \frac{\log q}{d-1} \rceil$ be the number of iterations. For $i \in [t]$, denote by $\mathcal{R}_i$ the set $\mathcal{R}$ at the end of the i-th iteration of the loop, and likewise for L_i, U_i, $\mathcal{R}'_i$, $\mathcal{R}''_i$. Without loss of generality, we assume that $\vec{v} = (1, \ldots, 1)$. The following two lemmas are used to bound the error and number of queries; their proofs are included in the full version.

▶ **Lemma 10.** *For all $i \in [t]$, $L_i \subset \mathcal{CH}_{\vec{v}}(P) \subseteq U_i$ and $\partial \mathcal{CH}_{\vec{v}}(P) \subseteq \bigcup_{R \in \mathcal{R}_i} R$.*

▶ **Lemma 11.** *The number of queries at the i-th iteration is at most $4^d d! \cdot (2^i + 1)^{d-1}$.*

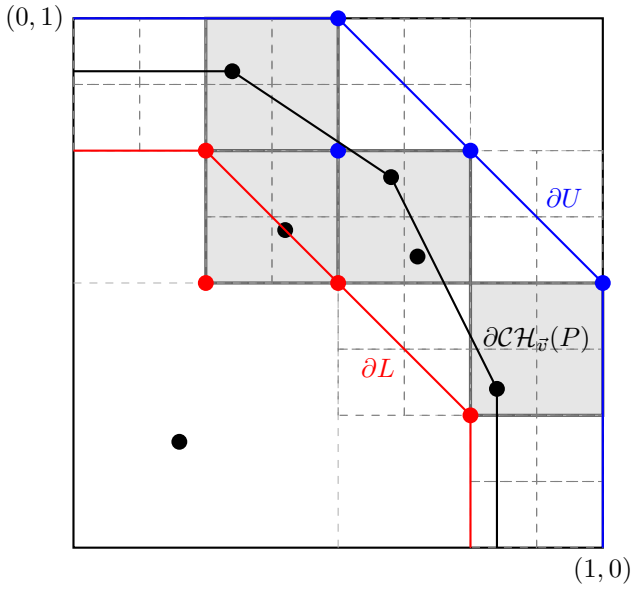


Figure 2 One iteration of Algorithm 2 for $d = 2, \vec{v} = (1, 1)$. The four queries in $\mathcal{R}'$ are filled in gray. The upper right corners of these queries are used to compute the set U with boundary shown in blue, while the lower left corners produce L in red. The boundary of $\mathcal{CH}_{\vec{v}}(P)$ lies in $U \setminus L$. The smallest dashed queries form the set $\mathcal{R}$ for the next iteration; their union covers $U \setminus L$.

As d is a constant, Lemma 11 implies the number of queries in the i-th iteration is $O((2^i+1)^{d-1})$. So the total number of queries is bounded by $\sum_{i=1}^{t}(2^i+1)^{d-1} = O(q)$. Since each query box in the i-th iteration has volume 2^{-id}, we have

$$\left\| \bigcup_{R \in \mathcal{R}_t} R \right\| \leq 2^{-td}|\mathcal{R}_t| = O(2^{-d \log q/(d-1)} q) = O(q^{-1/(d-1)}).$$

By Lemma 10, this further implies $\|(U_t \backslash \mathcal{CH}_{\vec{v}}(P)) \cap [0,1]^d\| = O(q^{-1/(d-1)})$, simply because $(U_t \backslash \mathcal{CH}_{\vec{v}}(P)) \cap [0,1]^d \subseteq (U_t \backslash L_t) \cap [0,1]^d \subseteq \bigcup_{R \in \mathcal{R}_t} R$.

We run Algorithm 2 for all $\vec{v} \in \{-1, 1\}^d$ and let $U_{\vec{v}}$ be the output for $\vec{v}$. Finally, we return $C^* = (\bigcap_{\vec{v} \in \{-1,1\}^d} U_{\vec{v}}) \cap [0,1]^d$ as the approximation of $\mathcal{CH}(P)$. The total number of queries is still $O(q)$. To bound the error $\|C^* \oplus \mathcal{CH}(P)\|$, we first observe $\mathcal{CH}(P) \subseteq C^*$. Indeed, since $\mathcal{CH}_{\vec{v}}(P) \subseteq U_{\vec{v}}$ for all $\vec{v} \in \{-1, 1\}^d$, we have $\mathcal{CH}(P) \subseteq \bigcap_{\vec{v} \in \{-1,1\}^d} U_{\vec{v}}$ by Lemma 9, which implies $\mathcal{CH}(P) \subseteq C^*$ because $P \subseteq [0,1]^d$. Again, by Lemma 9, we have

$$\|C^* \backslash \mathcal{CH}(P)\| \leq \sum_{\vec{v} \in \{-1,1\}^d} \|(U_{\vec{v}} \backslash \mathcal{CH}_{\vec{v}}(P)) \cap [0,1]^d\| = O(q^{-1/(d-1)}).$$

Therefore, the error of our algorithm is $O(q^{-1/(d-1)})$.

▶ **Theorem 12.** *There exists an algorithm for* Approximate Convex Hull *in* $\mathbb{R}^d$ *that performs* $O(q)$ *adaptive orthogonal emptiness queries and has error* $O(q^{-1/(d-1)})$.

We complement this result by showing that any deterministic algorithm with q adaptive orthogonal queries has error $\Omega(q^{-1/(d-1)})$. The argument is similar to and simpler than that of Theorem 8, so we include the precise details in the full version. The key difference is that Lemma 3 no longer holds for adaptive queries, but we can still apply the same argument for a single pair of hyperplanes and only lose a factor $n = q^{1/d}$ in the resulting error bound.

▶ **Theorem 13.** *Any deterministic algorithm for* Approximate Convex Hull *in* $\mathbb{R}^d$ *that performs* q *adaptive orthogonal emptiness queries has error* $\Omega(q^{-1/(d-1)})$.

4 Approximating convex hulls via halfplane queries

In this section, we present our results for Approximate Convex Hull in $\mathbb{R}^2$ via halfplane emptiness queries. Instead of working on halfplane emptiness queries directly, it is more convenient to consider *extreme halfplane queries*, which we formalize below. For two parallel lines ℓ and ℓ', let $\mathsf{dist}(\ell, \ell')$ denote the distance between them.

▶ **Definition 14** (extreme halfplane oracle)**.** *An* ***extreme halfplane oracle*** *on P takes a unit vector $\vec{v} \in \mathbb{S}^1$ as input and returns a halfplane H with normal vector $\vec{v}$ such that $P \subseteq H$. We say the oracle is* ***δ-accurate*** *for a number $\delta \geq 0$ if for every $\vec{v} \in \mathbb{S}^1$, the halfplane H returned by the oracle for query $\vec{v}$ satisfies the condition $\mathsf{dist}(\partial H, \partial H^*) \leq \delta$, where H^* is the minimal halfplane with normal vector $\vec{v}$ satisfying $P \subseteq H^*$.*

It is easy to see that a 0-accurate extreme halfplane oracle is stronger than a halfplane emptiness oracle. On the other hand, one can simulate an extreme halfplane query with accuracy δ via $O(1/\delta)$ evenly spaced non-adaptive halfplane emptiness queries. The query number can be reduced to $O(\log(1/\delta))$ via binary search in the adaptive model. The proofs of the following two lemmas are included in the full version.

▶ **Lemma 15.** *Given access to a halfplane emptiness oracle $\mathcal{O}$ on an unknown set P of points in the plane, one can build a δ-accurate extreme halfplane oracle on P which performs $O(\frac{1}{\delta})$ non-adaptive queries to $\mathcal{O}$ or performs $O(\log \frac{1}{\delta})$ adaptive queries to $\mathcal{O}$.*

We will also need the following lemma to argue that the error introduced by querying a δ-accurate extreme halfplane oracle, as opposed to a perfectly accurate one, is only $O(\delta)$.

▶ **Lemma 16.** *Let $H_1, \ldots, H_r$ be halfplanes, $C = \bigcap_{i=1}^r H_i$, and $\delta \in [0,1]$. Suppose $C \subseteq [0,1]^2$. For each $i \in [r]$, let H_i' be a halfplane parallel to H_i such that $C \subseteq H_i$ and $\mathsf{dist}(\partial H_i', \partial H_i) \leq \delta$. Define $C' = \bigcap_{i=1}^r H_i'$. Then $\|(C' \cap [0,1]^2) \oplus C\| = O(\delta)$.*

In what follows, we discuss the results for non-adaptive halfplane queries and adaptive halfplane queries individually. Same as before, for non-adaptive queries (Section 4.1), the algorithm is simple and the lower bound proof is nontrivial, while for adaptive queries (Section 4.2), the algorithm is more technical and the lower bound proof is simpler.

4.1 Non-adaptive halfplane queries

We first present our algorithmic result with non-adaptive halfplane emptiness queries. Let $P \subseteq [0,1]^2$ be a (unknown) set of points. We shall show how to approximate $\mathcal{CH}(P)$ with error $O(\frac{1}{q} + \delta)$ via q non-adaptive queries to a δ-accurate extreme halfplane oracle $\mathcal{E}$ on P. Setting $\delta = \frac{1}{q}$ and applying Lemma 15 yields an algorithm with error $O(1/q)$ that makes q^2 non-adaptive halfplane emptiness queries. Equivalently, q queries achieve error $O(1\sqrt{q})$.

Algorithm 3 NonAdaptiveHalfplane(q).

1: $V \leftarrow \{(\sin \frac{2i\pi}{q}, \cos \frac{2i\pi}{q}) : i \in [q]\}$
2: $H_{\vec{v}} \leftarrow \textsc{Query}(\vec{v})$ for all $\vec{v} \in V$
3: $C^* \leftarrow \bigcap_{\vec{v} \in V} H_{\vec{v}}$
4: **return** $C^* \cap [0,1]^2$

For a unit vector $\vec{v} \in \mathbb{S}^1$, let $\textsc{Query}(\vec{v})$ denote the output of $\mathcal{E}$ when queried with $\vec{v}$. Our algorithm for approximating $\mathcal{CH}(P)$ is very simple, and is presented in Algorithm 3. Let $V = \{(\sin \frac{2i\pi}{q}, \cos \frac{2i\pi}{q}) : i \in [q]\}$ be q unit vectors uniformly picked on $\mathbb{S}^1$. We query $\mathcal{E}$

with the vectors in V, and let $H_{\vec{v}} = \textsc{Query}(\vec{v})$ for all $\vec{v} \in V$. After that, we simply return $C^* \cap [0,1]^2$ where $C^* = \bigcap_{\vec{v} \in V} H_{\vec{v}}$. Clearly, the number of queries to $\mathcal{E}$ is q, and they are non-adaptive as the algorithm performs them at the same time. We bound the error to $O(1/\sqrt{q})$ in the full version.

▶ **Theorem 17.** *There exists an algorithm for* Approximate Convex Hull *in* $\mathbb{R}^2$ *that performs* q *non-adaptive halfplane emptiness queries and has error* $O(1/\sqrt{q})$.

Next, we show that any deterministic algorithm that performs q non-adaptive halfplane queries must have error $\Omega(1/\sqrt{q})$. Consider an algorithm $\mathbf{A}$. As the q queries performed by $\mathbf{A}$ are non-adaptive, they are independent of the point set P. Let $H_1, \ldots, H_q$ be these queries, each of which is a halfplane in $\mathbb{R}^2$. For $x, y \in (0,1)$, denote by $\ell_{x,y}$ the line that goes through the points $(x,0)$ and $(0,y)$. Also, for $X, Y \subseteq (0,1)$, define $\mathcal{L}_{X,Y} = \{\ell_{x,y} : x \in X \text{ and } y \in Y\}$. Fix a parameter $\delta = \frac{1}{10\sqrt{q}}$, and let $A = \{i\delta : i \in \mathbb{Z}\} \cap [\frac{1}{3}, \frac{2}{3}]$.

▶ **Lemma 18.** *There exists* $a, b \in A$ *such that* $\partial H_i \notin \mathcal{L}_{[a,a+\delta),[b,b+\delta)}$ *for all* $i \in [q]$.

Proof. Observe that for different $a, a' \in A$, the intervals $[a, a+\delta)$ and $[a', a'+\delta)$ are disjoint. Therefore, $\mathcal{L}_{[a,a+\delta),[b,b+\delta)}$ and $\mathcal{L}_{[a',a'+\delta),[b',b'+\delta)}$ are disjoint, for $(a,b), (a',b') \in A \times A$ with $(a,b) \neq (a',b')$. It follows that for each $i \in [q]$, there exists at most one pair $(a,b) \in A \times A$ such that $\partial H_i \in \mathcal{L}_{[a,a+\delta),[b,b+\delta)}$; we charge i to (a,b) if this is the case. By construction, $|A| \geq 3\sqrt{q}$ and thus $|A \times A| \geq 9q > q$. As such, there exists a pair $(a,b) \in A \times A$ such that no number in $[q]$ is charged to (a,b). We then have $\partial H_i \notin \mathcal{L}_{[a,a+\delta),[b,b+\delta)}$ for all $i \in [q]$. ◀

Let $a, b \in A$ be as in the above lemma, and $p^* = (\frac{a}{2} + \frac{\delta}{10}, \frac{b}{2} + \frac{\delta}{10})$. We define two sets $Z = \{(a,0), (0,b)\}$ and $Z' = \{(a,0), (0,b), p^*\}$; see figure 3.

▶ **Lemma 19.** *The point* p^* *is below the lines* $\ell_{a+\delta,b}$ *and* $\ell_{a,b+\delta}$.

Proof. Without loss of generality, we only need to show p^* is below $\ell_{a+\delta,b}$. The equation of $\ell_{a+\delta,b}$ is $bx + (a+\delta)y - b(a+\delta) = 0$. Since $a, b \in A$, we have $a, b \in [\frac{1}{3}, \frac{2}{3}]$. So we have

$$b\left(\frac{a}{2} + \frac{\delta}{10}\right) + (a+\delta)\left(\frac{b}{2} + \frac{\delta}{10}\right) - b(a+\delta) = \left(\frac{a+b+\delta}{10} - \frac{b}{2}\right)\delta < 0,$$

which implies that that p^* is below $\ell_{a+\delta,b}$. ◀

▶ **Corollary 20.** *For all* $i \in [q]$, $Z \cap H_i = \emptyset$ *iff* $Z' \cap H_i = \emptyset$.

Proof. If $p^* \notin H_i$, then $Z \cap H_i = Z' \cap H_i$ and we are done. So suppose $p^* \in H_i$. We claim that either $(a,0) \in H_i$ or $(0,b) \in H_i$. It suffices to show that at least one of $(a,0)$ and $(0,b)$ lies on the same side of ∂H_i as p^*. Assume both $(a,0)$ and $(0,b)$ are on the opposite side of ∂H_i from p^*. Then p^* and the segment $\ell_{a,b} \cap [0,1]^2$ must lie on the opposite side of ∂H_i. But by our construction, p^* is sufficiently close to $\ell_{a,b} \cap [0,1]^2$, which forces $\partial H_i = \ell_{x,y}$ for some $x \in [a,1]$ and $y \in [b,1]$. By Lemma 18, we cannot have $x \in [a, a+\delta)$ and $y \in [b, b+\delta)$ at the same time. So either $x \geq a+\delta$ or $y \geq b+\delta$. Without loss of generality, assume $x \geq a+\delta$. By Lemma 19, p^* is below $\ell_{a+\delta,b}$. Since $x \geq a+\delta$ and $y \geq b$, p^* is below ∂H_i as well. But both $(a,0)$ and $(0,b)$ are also below ∂H_i, contradicting our assumption. As such, we have either $(a,0) \in H_i$ or $(0,b) \in H_i$. It follows that $Z \cap H_i \neq \emptyset$ and $Z' \cap H_i \neq \emptyset$. ◀

The rest of the proof is almost the same as that of Theorem 8. The above lemma shows that the algorithm $\mathbf{A}$ cannot distinguish Z and Z', that is, it returns the same convex polygon when running on $P = Z$ and $P = Z'$. We only need to observe the following simple fact.

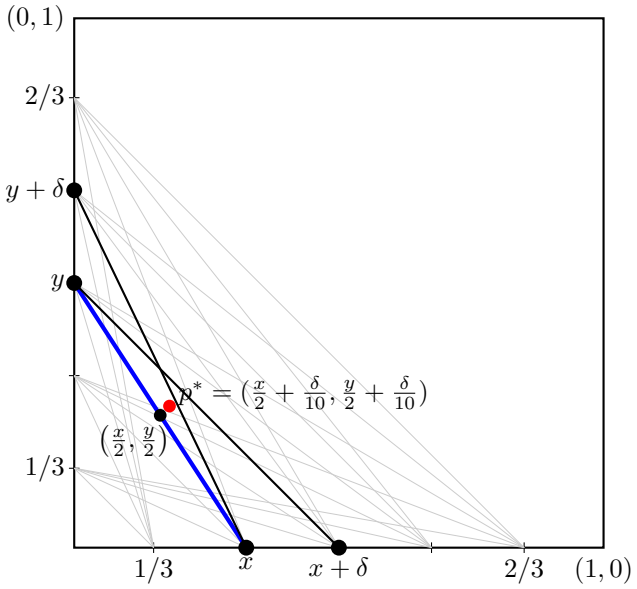


Figure 3 Lower bound construction for non-adaptive halfplane queries. The point p^* (red) is offset from the midpoint $(x/2, y/2)$ of $l_{x,y}$ to lie sufficiently far above $l_{x,y}$ (blue) but below both $l_{x+\delta,y}$ and $l_{x,y+\delta}$. Only a segment with both x-intercept in the interval $[x, x+\delta]$ and y-intercept in the interval $[y, y+\delta]$ can separate both $(x, 0), (0, y)$ from p^*. Doing so for all $\Theta(1/\delta^2) > q$ paired choices of x, y (gray lines) necessitates as many queries. Range [1/3, 2/3] enlarged for clarity.

▶ **Lemma 21.** $\|\mathcal{CH}(Z')\backslash\mathcal{CH}(Z)\| = \Omega(\delta)$.

Let C^* be the output of $\mathbf{A}$ when running on $P = Z$ or $P = Z'$. We have $\|\mathcal{CH}(Z')\backslash\mathcal{CH}(Z)\| \leq \|C^* \oplus \mathcal{CH}(Z)\| + \|C^* \oplus \mathcal{CH}(Z')\|$. Thus, the above lemma implies either $\|C^* \oplus \mathcal{CH}(Z)\| = \Omega(\delta)$ or $\|C^* \oplus \mathcal{CH}(Z')\| = \Omega(\delta)$. As $\delta = \Theta(1/\sqrt{q})$, the error of $\mathbf{A}$ is $\Omega(1/\sqrt{q})$.

▶ **Theorem 22.** *Any deterministic algorithm for* Approximate Convex Hull *in* $\mathbb{R}^2$ *that performs* q *non-adaptive halfplane emptiness queries incurs error* $\Omega(1/\sqrt{q})$.

4.2 Adaptive halfplane queries

We first present our algorithm with adaptive halfplane emptiness queries. Let $P \subseteq [0,1]^2$ be a (unknown) set of points. We shall show how to approximate $\mathcal{CH}(P)$ with error $O(\frac{1}{q^2} + q^2\delta)$ via $O(q \log q)$ adaptive queries to a δ-accurate extreme halfplane oracle $\mathcal{E}$ on P. Combining this with Lemma 15 will give us the desired result.

Same as before, for a unit vector $\vec{v} \in \mathbb{S}^1$, let Query$(\vec{v})$ denote the output of $\mathcal{E}$ when queried with $\vec{v}$. Our algorithm for approximating $\mathcal{CH}(P)$ is presented in Algorithm 4. We begin with a set V of unit vectors which is initially $\{(1,0),(0,1),(-1,0),(0,-1)\}$. We query $\mathcal{E}$ with the vectors in V and let $H_{\vec{v}} = \text{Query}(\vec{v})$. Then we iteratively add more vectors to V and query $\mathcal{E}$ with these vectors (the repeat-until loop in the algorithm). In each round, we first compute $C^* = \bigcap_{\vec{v}\in V} H_{\vec{v}}$ for the current V (line 4). Then line 5 sorts the vectors in the current V in clockwise order; let $\vec{v}_1, \ldots, \vec{v}_n$ be the sorted sequence. Let $\vec{v}_0 = \vec{v}_n$ and $\sigma_i = C^* \cap \partial H_{\vec{v}_i}$, which is the edge of C^* corresponding to the halfplane $H_{\vec{v}_i}$. We consider each pair $(\vec{v}_{i-1}, \vec{v}_i)$ of adjacent vectors satisfying $\mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i) \cdot |\sigma_i| > \frac{1}{q^2}$, where $\mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i)$ denotes the magnitude of the angle from $\vec{v}_{i-1}$ to $\vec{v}_i$. For each such pair $(\vec{v}_{i-1}, \vec{v}_i)$, we add a

new unit vector $\vec{v}$ to V, which is the bisector of $\vec{v}_{i-1}$ and $\vec{v}_i$, and query $\mathcal{E}$ with $\vec{v}$ to obtain the halfplane $H_{\vec{v}} = \textsc{Query}(\vec{v})$. After this, we proceed to the next round. The procedure terminates when no more vectors are added to V in a round. At the end, the algorithm returns the convex polygon $C^* \cap [0,1]^2$ as its output.

Algorithm 4 AdaptiveHalfplane(q).

$V \leftarrow \{(1,0),(0,1),(-1,0),(0,-1)\}$
$H_{\vec{v}} \leftarrow \textsc{Query}(\vec{v})$ for all $\vec{v} \in V$
repeat
  $C^* \leftarrow \bigcap_{\vec{v}\in V} H_{\vec{v}}$
  $(\vec{v}_1,\ldots,\vec{v}_n) \leftarrow \textsc{Sort}(V)$
  $\vec{v}_0 \leftarrow \vec{v}_n$
  $\sigma_i \leftarrow C^* \cap \partial H_{\vec{v}_i}$ for $i \in [n]_0$
  $I \leftarrow \{i \in [n] : \mathsf{ang}(\vec{v}_{i-1},\vec{v}_i) \cdot |\sigma_i| > \frac{1}{q^2}\}$
  for every $i \in I$ **do**
    $\vec{v} \leftarrow (\vec{v}_{i-1}+\vec{v}_i)/\|\vec{v}_{i-1}+\vec{v}_i\|_2$
    $V \leftarrow V \cup \{\vec{v}\}$
    $H_{\vec{v}} \leftarrow \textsc{Query}(\vec{v})$
until $I = \emptyset$
return $C^* \cap [0,1]^2$

Bounding the number of queries. Clearly, the number of queries we made in Algorithm 4 is just equal to the number of vectors in V at the end of the algorithm. Suppose the repeat-until loop in Algorithm 4 has t iterations. For $i \in [t]$, let V_i be the set V at the beginning of the i-th iteration and $C_i^* = \bigcap_{\vec{v}\in V_i} H_{\vec{v}}$ which is just the convex polygon C^* in the i-th iteration. Note that V_t is just the set V at the end of the algorithm, as the algorithm does not add new vectors to V in the last iteration. For convenience, we write $V_0 = \emptyset$. Denote by $\mathcal{A}(V_i)$ the set of angles between adjacent vectors in V_i. We first bound $|V_i \backslash V_{i-1}|$ for $i \in [t]$. The proofs for this subsection are included in the full version.

▶ **Lemma 23.** *$|V_i \backslash V_{i-1}| = O(q)$ for all $i \in [t]$.*

The above lemma directly implies $|V_t| = O(qt)$. Thus, it suffices to bound t. To this end, we establish the following properties for the angles in each $\mathcal{A}(V_i)$.

▶ **Lemma 24.** *For all $i \in [t]$ and $\alpha \in \mathcal{A}(V_i)$, we have $|\alpha| \geq \frac{1}{12q^2}$.*

▶ **Lemma 25.** *Let $i \in [t]$. We have $|\alpha| = \frac{\pi}{2^i}$ for all $\alpha \in \mathcal{A}(V_i) \backslash \mathcal{A}(V_{i+1})$.*

Lemma 24 and Lemma 25 together imply that $t = O(\log q)$. Indeed, $\mathcal{A}(V_i) \neq \mathcal{A}(V_{i+1})$ for all $i \in [t-1]$, for otherwise the algorithm terminates after the i-th iteration. Thus there exists $\alpha \in \mathcal{A}(V_{t-1}) \backslash \mathcal{A}(V_t)$. We have $\frac{\pi}{2^{t-1}} = |\alpha| \geq \frac{1}{q^2}$ by Lemma 24 and Lemma 25, which implies $t = O(\log q)$. It follows that $|V_t| = O(qt) = O(q \log q)$.

Bounding the error. Next, we show that the output $C^* \cap [0,1]^2$ of Algorithm 4 satisfies $\|(C^* \cap [0,1]^2) \oplus \mathcal{CH}(P)\| = O(\frac{1}{q^2} + \delta)$. Let $V = \{\vec{v}_1,\ldots,\vec{v}_n\}$ be the set of vectors at the end of Algorithm 4, where $\vec{v}_1,\ldots,\vec{v}_n$ are sorted in clockwise order. Set $\vec{v}_0 = \vec{v}_n$. As in the algorithm, for each $i \in [n]_0$, let $H_{\vec{v}_i} = \textsc{Query}(\vec{v}_i)$ and $\sigma_i = C^* \cap \partial H_{\vec{v}_i}$. Then $C^* = \bigcap_{i=1}^n H_{\vec{v}_i}$. We have $|\mathsf{ang}(\vec{v}_{i-1},\vec{v}_i)| \cdot (|\sigma_{i-1}| + |\sigma_i|) > \frac{1}{q^2}$ for all $i \in [n]_0$, for otherwise the repeat-until loop

in Algorithm 4 cannot terminate. Define $L_{\vec{v}_i}$ as the minimal halfplane with normal vector $\vec{v}_i$ that contains P and $C = \bigcap_{i=1}^{n} L_{\vec{v}_i}$. We have $C \subseteq C^*$, since $L_{\vec{v}_i} \subseteq H_{\vec{v}_i}$ for all $i \in [n]$. Also, we have $C \subseteq [0,1]^2$, since V consists of the four vectors $(1,0),(0,1),(-1,0),(0,-1)$, which guarantees that C is contained in the axis-parallel bounding box of P and thus contained in $[0,1]^2$. Now $\mathcal{CH}(P) \subseteq C \subseteq C^* \cap [0,1]^2$. By Lemma 16, $\|(C^* \cap [0,1]^2) \backslash C\| = O(\delta)$. So it suffices to show $\|C \backslash \mathcal{CH}(P)\| = O(\frac{1}{q^2} + \delta)$. Define $\mu_i = C \cap \partial L_{\vec{v}_i}$ for $i \in [n]$ and $\mu_0 = \mu_n$.

▶ **Lemma 26.** *$|\mu_i| \leq |\sigma_i| + O(q^2\delta)$ for all $i \in [n]$.*

Proof. Let a and a' be the endpoints of μ_i, where a (resp., a') is on the counterclockwise (resp., clockwise) side of μ_i with respect to C. We first consider the general case $\sigma_i \neq \emptyset$ and then discuss the special case $\sigma_i = \emptyset$. If $\sigma_i \neq \emptyset$, let b and b' be the endpoints of σ_i, where b (resp., b') is on the counterclockwise (resp., clockwise) side of σ_i with respect to C^*. As $\sigma_i \subseteq \partial H_{\vec{v}_i}$ and $\mu_i \subseteq \partial L_{\vec{v}_i}$, the 4-gon $aa'b'b$ is a trapezoid with parallel edges μ_i and σ_i. Therefore, we have the equation

$$|\mu_i| = |\sigma_i| + \frac{h}{\tan \angle baa'} + \frac{h}{\tan \angle b'a'a}.$$

where h is the height of this trapezoid. We have $h \leq \delta$ since $\mathsf{dist}(\partial H_{\vec{v}_i}, \partial L_{\vec{v}_i}) \leq \delta$. Furthermore, we observe that $|\angle baa'| \geq \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i)$ and $|\angle b'a'a| \geq \mathsf{ang}(\vec{v}_i, \vec{v}_{i+1})$. Indeed, $\angle baa' = \pi - \angle b'ba$. Since $a \in C^*$, the angle $\angle b'ba$ is smaller than or equal to the angle of C^* at the vertex b, where the latter is just $\pi - \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i)$. Thus,

$$|\angle baa'| = \pi - |\angle b'ba| \geq \pi - (\pi - \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i)) = \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i).$$

For the same reason, $|\angle b'a'a| \geq \mathsf{ang}(\vec{v}_i, \vec{v}_{i+1})$. By Lemma 24, it follows that $|\angle baa'| \geq \frac{1}{12q^2}$ and $|\angle b'a'a| \geq \frac{1}{12q^2}$. As such, $|\mu_i| = |\sigma_i| + O(q^2\delta)$.

If $\sigma_i = \emptyset$, then C^* is contained in the interior of $H_{\vec{v}_i}$. In this case, C^* has a unique vertex that is closest to $\partial H_{\vec{v}_i}$. Let both b and b' be this vertex. The same argument applies. ◀

The definition of each $L_{\vec{v}_i}$ guarantees that there exists a point $p_i \in P$ satisfying $p_i \in \partial L_{\vec{v}_i}$. As $p_i \in P \subseteq C$, we have $p_i \in \mu_i$. Let $P' = \{p_1, \ldots, p_n\}$. We have $P' \subseteq P$ and thus $\|C \backslash \mathcal{CH}(P')\| \geq \|C \backslash \mathcal{CH}(P)\|$. We shall show $\|C \backslash \mathcal{CH}(P')\| = O(\frac{1}{q^2} + \delta)$, which bounds $\|C \backslash \mathcal{CH}(P)\|$ as well. For each $i \in [n]$, let a_i be the intersection point of μ_{i-1} and μ_i, and τ_i be the segment with endpoints p_{i-1} and p_i. Note that $\|C \backslash \mathcal{CH}(P')\| = \sum_{i=1}^{n} \|\triangle p_{i-1}p_ia_i\|$.

▶ **Lemma 27.** *$\|\triangle p_{i-1}p_ia_i\| = O(|\tau_i| \cdot (\frac{1}{q^2} + q^2\delta))$ for all $i \in [n]$.*

Proof. By construction, the distance between p_{i-1} and p_i is just $|\tau_i|$. So we only need to show that the height h of $\triangle p_{i-1}p_ia_i$ with respect to the edge $p_{i-1}p_i$ is bounded by $O(\frac{1}{q^2} + q^2\delta)$. Let $\theta = \angle a_ip_ip_{i-1}$. As the distance between a_i and p_i is at most $|\mu_i|$, we have

$$h \leq |\mu_i| \cdot \sin|\theta| \leq |\mu_i| \cdot \theta \leq \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i) \cdot |\mu_i|.$$

By Lemma 26, this implies $h = \mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i) \cdot |\sigma_i| + O(q^2\delta)$. Since $\mathsf{ang}(\vec{v}_{i-1}, \vec{v}_i) \cdot |\sigma_i| \leq \frac{1}{q^2}$, we have $h = O(\frac{1}{q^2} + q^2\delta)$ and thus $\|\triangle p_{i-1}p_ia_i\| = O(|\tau_i| \cdot (\frac{1}{q^2} + q^2\delta))$. ◀

The above lemma directly implies

$$\|C \backslash \mathcal{CH}(P')\| = \sum_{i=1}^{n} \|\triangle p_{i-1}p_ia_i\| = O((\sum_{i=1}^{n} |\tau_i|) \cdot (\frac{1}{q^2} + q^2\delta)).$$

As $\tau_1, \ldots, \tau_n$ are just the edge of the convex polygon $\mathcal{CH}(P')$, we have $\sum_{i=1}^{n} |\tau_i| = O(1)$ and thus $\|C \backslash \mathcal{CH}(P')\| = O(\frac{1}{q^2} + q^2\delta)$. As Lemma 16 implies $\|(C^* \cap [0,1]^2) \backslash C\| = O(\delta)$, we have $\|(C^* \cap [0,1]^2) \oplus \mathcal{CH}(P)\| \leq \|(C^* \cap [0,1]^2) \backslash \mathcal{CH}(P')\| = O(\frac{1}{q^2} + q^2\delta)$. Setting $\delta = \frac{1}{q^4}$, we see that one can approximate $\mathcal{CH}(P)$ with error $O(\frac{1}{q^2})$ via $O(q \log q)$ queries to a $\frac{1}{q^4}$-accurate extreme halfplane oracle for P.

Lemma 15 shows that a query to a $\frac{1}{q^4}$-accurate extreme halfplane oracle can be simulated with $O(\log q)$ adaptive halfplane emptiness queries. Therefore, one can approximate $\mathcal{CH}(P)$ with error $O(\frac{1}{q^2})$ via $O(q \log^2 q)$ adaptive halfplane emptiness queries. Equivalently, if we are allowed to use q queries, then the error achieved can be bounded by $O(\frac{\log^2 q}{q^2})$, i.e., $\widetilde{O}(\frac{1}{q^2})$.

▶ **Theorem 28.** *There exists an algorithm for* Approximate Convex Hull *in* $\mathbb{R}^2$ *that performs* q *adaptive halfplane emptiness queries and has error* $\widetilde{O}(1/q^2)$.

We remark that replacing the threshold $\frac{1}{q^2}$ in Algorithm 4 with an even smaller number cannot result in an improvement for Theorem 28: while it can decrease the error of the output, it also increases the number of queries substantially. In fact, the bound $\widetilde{O}(1/q^2)$ is already tight up to logarithmic factors. The details are included in the full version.

▶ **Theorem 29.** *Any deterministic algorithm for* Approximate Convex Hull *in* $\mathbb{R}^2$ *that performs* q *adaptive halfplane emptiness queries has error* $\Omega(1/q^2)$.

References

1. Peyman Afshani and Timothy M Chan. On approximate range counting and depth. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pages 337–343, 2007. doi:10.1145/1247069.1247129.
2. Peyman Afshani and Timothy M Chan. Optimal halfspace range reporting in three dimensions. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 180–186. SIAM, 2009. doi:10.1137/1.9781611973068.21.
3. Pankaj K Agarwal. Range searching. In *Handbook of discrete and computational geometry*, pages 1057–1092. Chapman and Hall/CRC, 2017.
4. Pankaj K Agarwal, Jeff Erickson, et al. Geometric range searching and its relatives. *Contemporary Mathematics*, 223(1):56, 1999.
5. Pankaj K. Agarwal and Jirí Matousek. On range searching with semialgebraic sets. *Discrete & Computational Geometry*, 11(4):393–418, 1994. doi:10.1007/BF02574015.
6. Pankaj K Agarwal and Jiří Matoušek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13(4):325–345, 1995. doi:10.1007/BF01293483.
7. David Avis and David Bremner. How good are convex hull algorithms? In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 20–28, 1995. doi:10.1145/220279.220282.
8. C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996. doi:10.1145/235815.235821.
9. Jon Louis Bentley and Hermann A. Maurer. Efficient worst-case data structures for range searching. *Acta Informatica*, 13(2):155–168, 1980. doi:10.1007/BF00263991.
10. Kevin Q Brown. Voronoi diagrams from convex hulls. *Information processing letters*, 9(5):223–228, 1979. doi:10.1016/0020-0190(79)90074-7.
11. Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & computational geometry*, 16(4):361–368, 1996. doi:10.1007/BF02712873.
12. Timothy M Chan, Kasper Green Larsen, and Mihai Pătraşcu. Orthogonal range searching on the ram, revisited. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 1–10, 2011.

13 Bernard Chazelle. Lower bounds for orthogonal range searching: I. the reporting case. *Journal of the ACM (JACM)*, 37(2):200–212, 1990. doi:10.1145/77600.77614.

14 Bernard Chazelle, Ding Liu, and Avner Magen. Sublinear geometric algorithms. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 531–540, 2003. doi:10.1145/780542.780620.

15 Artur Czumaj, Funda Ergün, Lance Fortnow, Avner Magen, Ilan Newman, Ronitt Rubinfeld, and Christian Sohler. Approximating the weight of the euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing*, 35(1):91–109, 2005. doi:10.1137/S0097539703435297.

16 Artur Czumaj and Christian Sohler. Property testing with geometric queries. In *European Symposium on Algorithms*, pages 266–277. Springer, 2001. doi:10.1007/3-540-44676-1_22.

17 Anne Driemel, Morteza Monemizadeh, Eunjin Oh, Frank Staals, and David P. Woodruff. Range Counting Oracles for Geometric Problems. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry (SoCG 2025)*, volume 332 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:16, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2025.42.

18 Herbert Edelsbrunner and Emo Welzl. Halfplanar range search in linear space and o (n0. 695) query time. *Information processing letters*, 23(5):289–293, 1986. doi:10.1016/0020-0190(86)90088-8.

19 Jeff Erickson. Space-time tradeoffs for emptiness queries. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 304–313, 1997.

20 David Haussler and Emo Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the second annual symposium on Computational geometry*, pages 61–71, 1986. doi:10.1145/10515.10522.

21 Jiří Matoušek. Geometric range searching. *ACM Computing Surveys (CSUR)*, 26(4):422–461, 1994. doi:10.1145/197405.197408.

22 Morteza Monemizadeh. Facility Location in the Sublinear Geometric Model. In Nicole Megow and Adam Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:24, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.6.

23 Edgar A Ramos. On range reporting, ray shooting and k-level construction. In *Proceedings of the fifteenth annual symposium on Computational Geometry*, pages 390–399, 1999. doi:10.1145/304893.304993.

24 Raimund Seidel. *A convex hull algorithm optimal for point sets in even dimensions*. PhD thesis, University of British Columbia, 1981.

25 Raimund Seidel. Convex hull computations. In *Handbook of discrete and computational geometry*, pages 687–703. Chapman and Hall/CRC, 2017.

26 Micha Sharir and Hayim Shaul. Semialgebraic range reporting and emptiness searching with applications. *SIAM Journal on Computing*, 40(4):1045–1074, 2011. doi:10.1137/090765092.

27 Andrew Chi-Chih Yao. A lower bound to finding convex hulls. *Journal of the ACM (JACM)*, 28(4):780–787, 1981. doi:10.1145/322276.322289.