# Lower Bounds for Intersection Reporting Among Flat Objects

**Peyman Afshani** ✉
Aarhus University, Denmark

**Pingan Cheng** ✉
Aarhus University, Denmark

## Abstract

Recently, Ezra and Sharir [20] showed an $O(n^{3/2+\sigma})$ space and $O(n^{1/2+\sigma})$ query time data structure for ray shooting among triangles in $\mathbb{R}^3$. This improves the upper bound given by the classical $S(n)Q(n)^4 = O(n^{4+\sigma})$ space-time tradeoff for the first time in almost 25 years and in fact lies on the tradeoff curve of $S(n)Q(n)^3 = O(n^{3+\sigma})$. However, it seems difficult to apply their techniques beyond this specific space and time combination. This pheonomenon appears persistently in almost all recent advances of flat object intersection searching, e.g., line-tetrahedron intersection in $\mathbb{R}^4$ [19], triangle-triangle intersection in $\mathbb{R}^4$ [19], or even among flat semialgebraic objects [5].

We give a timely explanation to this phenomenon from a lower bound perspective. We prove that given a set $\mathcal{S}$ of $(d-1)$-dimensional simplicies in $\mathbb{R}^d$, any data structure that can report all intersections with a query line in small $(n^{o(1)})$ query time must use $\Omega(n^{2(d-1)-o(1)})$ space. This dashes the hope of any significant improvement to the tradeoff curves for small query time and almost matches the classical upper bound. We also obtain an almost matching space lower bound of $\Omega(n^{6-o(1)})$ for triangle-triangle intersection reporting in $\mathbb{R}^4$ when the query time is small. Along the way, we further develop the previous lower bound techniques by Afshani and Cheng [2, 3].

**2012 ACM Subject Classification** Theory of computation → Computational geometry

**Keywords and phrases** Computational Geometry, Intersection Searching, Data Structure Lower Bounds

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2023.3

**Related Version** *Full Version*: https://arxiv.org/abs/2302.11433

**Funding** Supported by DFF (Det Frie Forskningsråd) of Danish Council for Independent Research under grant ID DFF–7014–00404.

## 1 Introduction

Given a set $\mathcal{S}$ of triangles in $\mathbb{R}^3$, how to preprocess $\mathcal{S}$ such that given any query ray $\gamma$, we can efficiently determine the first triangle intersecting $\gamma$ or report no such triangle exists? This problem, known as ray shooting, is one of the most important problems in computational geometry with countless papers published over the last three decades [27, 8, 25, 26, 17, 10, 29, 30, 11, 18, 20, 5]. For a comprehensive overview of this problem, we refer the readers to an excellent recent survey [28].

Recently, there have been considerable and significant advances on ray shooting and a number of problems related to intersection searching on the upper bound side. We complement these attempts by giving lower bounds for a number of intersection searching problems; these also settle a recent open question asked by Ezra and Sharir [20].

## 1.1 Background and Previous Results

In geometric intersection searching, the input is a set $\mathcal{S}$ of geometric objects and the goal is to preprocess $\mathcal{S}$ into a data structure such that given a geometric object $\gamma$ at the query time, one can find all the objects in $\mathcal{S}$ that intersect $\gamma$. In the reporting variant of such a query, the output should be the list of all the intersecting objects in $\mathcal{S}$. Intersection searching is a generalization of range searching, a fundamental and core area of computational geometry [4]. This captures many natural classic problems e.g., simplex range reporting where the inputs are points (0-flats) and the queries are simplices (subsets of $d$-flats), ray shooting reporting among triangles in $\mathbb{R}^3$ where the inputs are triangles (subsets of 2-flats) and the queries are rays (subsets of 1-flats) and so on. See [4, 28] for more information.

Without going too much in-depth, it suffices to say that by now, the simplex range searching problem is more or less well-understood. There are classical solutions that offer the space and query time trade-off of $S(n)Q^d(n) = \tilde{O}(n^d)$ where $S(n)$ and $Q(n)$ are the space and query time of the data structure [15, 23, 12] and there are a number of almost matching lower bounds that show these are essentially tight [1, 13, 16].

However, intersection searching in higher dimensions is less well-understood. The classical technique is to lift the problem to the parametric space of the input or the query, reducing the problem to semialgebraic range searching, a generalized version of simplex range searching, where queries are semialgebraic sets of constant description complexity. In mid-1990s, semialgebraic range searching could only be solved efficiently in four and lower dimensions by classical tools developed for simplex range searching [7], resulting in a space-time trade-off bound of $S(n)Q(n)^4 = O(n^{4+\sigma})$ for line-triangle intersection searching in $\mathbb{R}^3$, where $\sigma > 0$ can be any small constant.

Recently, using polynomial techniques [22, 21], several major advances have been made on semialgebraic range searching. For example, near optimal small linear space and fast query data structure were developed [9, 24, 6]. These almost match the newly discovered lower bound bounds [2, 3]. However, these polynomial techniques also have led to significant advances in intersection searching. For ray-triangle intersection reporting in $\mathbb{R}^3$, Ezra and Sharir [20] showed that using algebraic techniques, it is possible to build a data structure of space $S(n) = O(n^{3/2+\sigma})$ and query time $Q(n) = O(n^{1/2+\sigma})$ for ray shooting among triangles. The significance of this result is that it improves the upper bound given by the trade-off curve of $S(n)Q(n)^4 = O(n^{4+\sigma})$ for the first time in almost 25 years and in fact it lies on the trade-off curve of $S(n)Q(n)^3 = O(n^{3+\sigma})$. This leads to the following very interesting question asked by Ezra and Sharir. To quote them directly: *"There are several open questions that our work raises. First, can we improve our trade-off for all values of storage, beyond the special values of $O(n^{3/2+\varepsilon})$ storage and $O(n^{1/2+\varepsilon})$ query time? Ideally, can we obtain query time of $O(n^{1+\varepsilon}/s^{1/3})$, with s storage, as in the case of ray shooting amid planes? Alternatively, can one establish a lower-bound argument that shows the limitations of our technique?"*

Inspired by [20], additional results for flat intersection searching were discovered during the last two years, e.g., triangle-triangle intersection searching in $\mathbb{R}^4$ [19], line-tetrahedron intersection searching in $\mathbb{R}^4$ [19], curve-disk intersection searching in $\mathbb{R}^3$ [5], and even more general semialgebraic flat intersection searching [5]. Similar to the result in [20], the improved results are only observed for a special space-time combination and the improvement to the entire trade-off curve is limited. This once again raises the question of whether it is possible to obtain the trade-off curve of $S(n)Q(n)^d = O(n^{d+\sigma})$ for intersection searching in $\mathbb{R}^d$.

## 1.2 Our Results

We give a negative answer to this question. We show that answering intersection searching queries in polylogarithmic time when the queries are lines in $\mathbb{R}^d$ and input objects are subsets of $(d-1)$-flats (that we call hyperslabs) requires $\mathring{\Omega}(n^{2(d-1)})$ space[1]. Our lower bound in fact applies to "thin" $(d-1)$-dimensional slabs (e.g., in 3D, that would be the intersection of the region between two parallel hyperplanes with another hyperplane). This almost matches the current upper bound for the problem and shows that the improvement in [20] cannot significantly improve the trade-off curve when the query time is small. To be specific, we obtain a lower bound of

$$S(n) = \mathring{\Omega}\left(\frac{n^{2(d-1)}}{Q(n)^{4(3d-1)(d-1)-1}}\right)$$

for line-hyperslab intersection reporting in $\mathbb{R}^d$ and a lower bound of

$$S(n) = \mathring{\Omega}\left(\frac{n^6}{Q(n)^{125}}\right)$$

for triangle-triangle intersection reporting in $\mathbb{R}^4$. Here, $S(n)$ and $Q(n)$ are the space and query time of the data structure. Similar to the other semialgebraic range reporting lower bounds [2, 3], these lower bounds have a much larger exponent on $Q(n)$ than on $n$ which does allow for substantial improvements when $Q(n)$ is no longer too small; we have not opted for optimizing the exponent of $Q(n)$ in our bounds and using tighter arguments, these exponents can be improved but they cannot match the exponent of $n$.

We believe our results are timely as flat intersection searching is a hotly investigated field recently, and as mentioned, with many open questions that need to be answered from a lower bound point of view.

## 1.3 Technical Contributions

From a technical point of view, our results require going beyond the previous attempts [2, 3]. To elaborate, the previous general technique assumed a particular form for the polynomials involved in defining the query semialgebraic ranges, namely, of the form $X_1 = X_2^\Delta + P(X_1, \cdots, X_d)$ where the coefficients of $P$ had to be independent and thus could be set arbitrarily small. Unfortunately, the problems in intersection searching cannot fit this framework and there seems to be no easy fix for the following reason. The previous technique relies heavily on the fact that if the coefficients of $P$ is small enough, then one can approximate $X_1$ with $X_2^\Delta$ and for the technique to work both conditions must hold (i.e., small coefficients for $P$ and having degree $\Delta$ on $X_2$).

Generally speaking, the previous techniques do not say anything about problems in which the polynomials involved have a specific form; the only exception is the lower bound for annuli [2] where specific approaches had to be created that could only be applied to the specific algebraic form of circles.

The issue is very prominent in intersection searching where we are dealing with polynomials where the coefficients of the monomials are no longer independent and the polynomials involved have specific forms; for instance, the coefficient of $X_2^\Delta$ is zero. We introduce techniques that allows us circumvent these limitations and obtain lower bounds for some broader class of problems that involve polynomials with some specific forms.

[1] In this paper, $\mathring{\Omega}(\cdot), \mathring{\Theta}(\cdot), \mathring{O}(\cdot)$ hides $n^{o(1)}$ factors; $\tilde{\Omega}(\cdot), \tilde{\Theta}(\cdot), \tilde{O}(\cdot)$ hides $\log^{O(1)} n$ factors.

# 2 Preliminaries

## 2.1 The Geometric Range Reporting Lower Bound Framework in the Pointer Machine

We use the pointer machine lower bound framework that was also used in the latest proofs [3]. This is a streamlined version of the one originally proposed by Chazelle [14] and Chazelle and Rosenberg [16]. In the pointer machine model, the memory is represented as a directed graph where each node stores one point as well as two pointers pointing to two other nodes in the graph. Given a query, the algorithms starts from a special "root" node, and then explores a subgraph which contains all the input points to report. The size of the directed graph is then a lower bound for the space usage and then minimum subgraph needed to explore to answer any query is a lower bound for the query time.

Intuitively, to answer a range reporting query efficiently, we need to store the output points to the query close to each other. If the answer to any query contains many points and two queries share very few points in common, many points must be stored multiple times, leading to a big space usage.

The streamlined version of the framework is the following [3].

▶ **Theorem 1.** *Suppose a $d$-dimensional geometric range reporting problem admits an $S(n)$ space and $Q(n) + O(k)$ query time data structure, where $n$ is the input size and $k$ is the output size. Let $Vol(\cdot)$ denote the $d$-dimensional Lebesgue measure. Assume we can find $m = n^c$, for a positive constant $c$, ranges $\mathscr{R}_1, \mathscr{R}_2, \cdots, \mathscr{R}_m$ in a $d$-dimensional hyperrectangle $\boldsymbol{R}$ such that*

1. *$\forall i = 1, 2, \cdots, m$, $Vol(\mathscr{R}_i \cap \boldsymbol{R}) \geq 4c\, Vol(\boldsymbol{R})Q(n)/n$;*
2. *$Vol(\mathscr{R}_i \cap \mathscr{R}_j) = O(Vol(\boldsymbol{R})/(n2^{\sqrt{\log n}}))$ for all $i \neq j$ .*

*Then, we have $S(n) = \overset{o}{\Omega}(mQ(n))$.*

## 2.2 Notations and Definitions for Polynomials

In this paper, we only consider polynomials on the reals. Let $P(X_1, \cdots, X_d)$ be a polynomial on $d$ indeterminates of degree $\Delta$. Sometimes we will use the notation $X$ to denote the set of $d$ interminates $X_1, \cdots, X_d$ and so we can write $P$ as $P(X)$. We denote by $I_{d,\Delta}$ a set of $d$-tuples of non-negative integers $(i_1, \cdots, i_d)$ whose sum is at most $\Delta$. We might omit the subscripts $d$ and $\Delta$ if they are clear from the context. For an $\mathbf{i} \in I$, we use the notation $X^{\mathbf{i}}$ to represent the monomial $\Pi_{j=1}^{d} X_j^{i_j}$ where $\mathbf{i} = (i_1, \cdots, i_d)$. Thus, given real coefficients $A_{\mathbf{i}}$, for $\mathbf{i} \in I$, we can write $P$ as $\sum_{\mathbf{i} \in I} A_{\mathbf{i}} X^{\mathbf{i}}$.

## 2.3 Geometric Lemmas

We introduce and generalize some geometric lemmas about the intersection of polynomials used in [2]. We first generalize the core Lemma in [2] for univariate polynomials, using a proof similar to [3]. We refer the readers to the full version of the paper for the proof.

▶ **Lemma 2.** *Let $P(x) = \sum_{i=0}^{\Delta} a_i x^i$ and $Q(x) = \sum_{i=0}^{\Delta} b_i x^i$ be two univariate (constant) degree-$\Delta$ polynomials in $\mathbb{R}[x]$ and $|a_i - b_i| \geq \eta$ for some $0 \leq i \leq \Delta$.*

*Suppose there is an interval $\mathscr{I}$ of $x$ such that for every $x_0 \in \mathscr{I}$ we have $|P(x_0) - Q(x_0)| \leq w$, then the length of $\mathscr{I}$ is upper bounded by $O((w/\eta)^{1/\mathcal{U}})$, where $\mathcal{U} = \binom{\Delta+1}{2}$ and the $O(\cdot)$ notation hides constant factors that depend on $\Delta$.*

Using Lemma 2, we can show the following. The proof is given in the full version.

▶ **Lemma 3.** *Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_\mathbf{i} X^\mathbf{i}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_\mathbf{i} X^\mathbf{i}$ be two $d$-variate degree-$\Delta$ polynomials in $\mathbb{R}[X]$ and $|A_\mathbf{i} - B_\mathbf{i}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$.*

*Suppose for each assignment $X_d \in \mathscr{I}_d$ to $P_1, P_2$, where $\mathscr{I}_d$ is an interval for $X_d$, all the coefficients of the resulting $(d-1)$-variate polynomial $Q_1(X_1, \cdots X_{d-1})$ and $Q_2(X_1, \cdots X_{d-1})$ differ by at most $\eta_{d-1}$, then $|\mathscr{I}_d| = O((\eta_{d-1}/\eta_d)^{1/\mathcal{U}})$.*

We can use Lemma 3 $d-2$ times, and obtain the following corollary.

▶ **Corollary 4.** *Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_\mathbf{i} X^\mathbf{i}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_\mathbf{i} X^\mathbf{i}$ be two $d$-variate degree-$\Delta$ polynomials in $\mathbb{R}[X]$ and $|A_\mathbf{i} - B_\mathbf{i}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$ for $d \geq 3$.*

*Suppose for each assignment $X_i \in \mathscr{I}_i$ to $P_1, P_2$, where $\mathscr{I}_i$ is an interval for $X_i$, for $i = 3, 4, \cdots, d$, all the coefficients of the resulting bivariate polynomial $Q_1(X_1, X_2)$ and $Q_2(X_1, X_2)$ differ by at most $\eta_2$, then $|\mathscr{I}_i| = O((\eta_{i-1}/\eta_i)^{1/\mathcal{U}})$ for all $i = 3, 4, \cdots, d$.*

To get the final corollary, we would like the set each $\eta_i$ such that the length of all each interval $\mathscr{I}_i$ is bounded by some parameter $\vartheta$ for $i = 3, \cdots, d$. We thus set $\eta_{d-i} = \eta_{d-i+1}\vartheta^\mathcal{U}$.

▶ **Corollary 5.** *Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_\mathbf{i} X^\mathbf{i}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_\mathbf{i} X^\mathbf{i}$ be two $d$-variate degree-$\Delta$ polynomials in $\mathbb{R}[X]$ and $|A_\mathbf{i} - B_\mathbf{i}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$ for $d \geq 3$.*

*Suppose for each assignment $X_i \in \mathscr{I}_i$ to $P_1, P_2$, where $\mathscr{I}_i$ is an interval for $X_i$, for $i = 3, 4, \cdots, d$, all the coefficients of the resulting bivariate polynomial $Q_1(X_1, X_2)$ and $Q_2(X_1, X_2)$ differ by at most $\eta_d \vartheta^{\mathcal{U}(d-2)}$, then $|\mathscr{I}_i| = O(\vartheta)$ for all $i = 3, 4, \cdots, d$.*

## 2.4 Algebra Preliminaries

In this section, we review some tools from algebra. The first tool we will use is the linearity of determinants from linear algebra.

▶ **Theorem 6** (Linearity of Determinants). *Let $A = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_n\end{bmatrix}$ be an $n \times n$ matrix where each $\mathbf{a}_i \in \mathbb{R}^n$ is a vector. Suppose $\mathbf{a}_j = r \cdot \mathbf{w} + \mathbf{v}$ for some $r \in \mathbb{R}$ and $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$, then the determinant of $A$, denoted by $\det(A)$, is*

$$\begin{aligned}&\det(A)\\&= \det(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{a}_j & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix})\\&= r \cdot \det(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{w} & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix}) + \det(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{v} & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix}).\end{aligned}$$

We will use two types of special matrices in the paper. The first is Vandermonde matrices.

▶ **Definition 7** (Vandermonde Matrices). *An $n \times n$ Vandermonde matrix is defined by $n$ values $x_1, \cdots, x_n$ such that each entry $e_{ij} = x_i^{j-1}$ for $1 \leq i, j \leq n$.*

We can compute the determinant of Vandermonde matrices easily.

▶ **Theorem 8** (Determinant of Vandermonde Matrices). *Let $V$ be a Vandermonde matrix defined by parameters $x_1, \cdots, x_n$. Then $\det(V) = \prod_{1 \leq i < j \leq n}(x_j - x_i)$.*

We also need Sylvester matrices.

▶ **Definition 9** (Sylvester Matrices). *Let $P = \sum_{i=0}^{\Delta_1} a_i x^i$ and $Q = \sum_{i=0}^{\Delta_2} b_i x^i$ be two univariate polynomials over $\mathbb{R}[x]$ of degrees $\Delta_1, \Delta_2$ respectively . Then the Sylvester matrix of $P$ and $Q$, denoted by Syl$(P, Q)$, is a $(\Delta_1 + \Delta_2) \times (\Delta_1 + \Delta_2)$ matrix of the following form*

$$\begin{bmatrix} a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_0 & 0 & \cdots & 0 & 0 \\ 0 & a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_1 & a_0 \\ b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_0 & 0 & \cdots & 0 & 0 \\ 0 & b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_1 & b_0 \end{bmatrix}.$$

The Sylvester matrix has $\Delta_2$ rows with entries from $P$ and $\Delta_1$ rows with entries from $Q$. For example, the Sylvester matrx of two polynomials $P = p_1x + p_2$ and $Q = q_1x + q_2$ is

$$\text{Syl}(P, Q) = \begin{bmatrix} p_1 & p_2 \\ q_1 & q_2. \end{bmatrix}$$

One application of Sylvester matrices is to compute the resultant, which is one of the important tools in algebraic geometry. One significance of the resultant is that it equals zero if and only if $P$ and $Q$ have a common factor.

▶ **Definition 10.** *Let $P, Q$ be two univariate polynomials over $\mathbb{R}$. The resultant of $P$ and $Q$, denoted by $Res(P, Q)$, is defined to be the determinant of the Sylvester matrix of $P$ and $Q$, i.e., $Res(P, Q) = \det(Syl(P, Q))$.*

## 3 An Algebraic Geometry Lemma

In this section, we prove an important algebraic geometry lemma that will later be used in our lower bound proof.

▶ **Lemma 11.** *Let $F$ and $G$ be two univariate polynomials on $x$ of degree $\Delta_F$ and $\Delta_G$ respectively and the leading coefficient of $G$ is 1. Let $P(x, y) \equiv yG(x) - F(x)$.*

*Let $L$ be a set of $\ell = \Delta_1 + \Delta_G + 1$ points $(x_k, y_k)$ where $\Delta_1 \geq \Delta_F - 1$ and each $x_k = \Theta(1)$ such that $|P(x_k, y_k)| \leq \varepsilon < 1$ for a parameter $\varepsilon$, and $G(x_k) = \Theta(1)$.*

*Let $V$ be a vector of $\ell$ monomials consisting of monomials $x^i$ for $0 \leq i \leq \Delta_1$ and monomials $yx^i$ for $0 \leq i \leq \Delta_G - 1$.*

*If $A$ is an $\ell \times \ell$ matrix where the $k$-th row of $A$ is the evaluation of the vector $V$ on point $(x_k, y_k)$, then $|\det(A)| \geq \Omega(Res(G, F)\lambda^{\ell^2}) - O(\varepsilon)$ where $\lambda = \min_{1 \leq k_1 < k_2 \leq \ell} |x_{k_1} - x_{k_2}|$.*

**Proof.** Note that if $\text{Res}(G, F) = 0$, then there is nothing to prove and thus we can assume this is not the case. Now observe that since $G(x_k) = \Theta(1)$, we can write $y_k = \frac{F(x_k)}{G(x_k)} + \gamma_k$ where $|\gamma_k| = O(\varepsilon)$.

Now consider the matrix $A$ and plug in this value of $y_k$. An entry of $A$ is in the form of a monomial $yx^i$ being evaluated on a point $(x_k, y_k)$ and thus we have:

$$y_k x_k^i = \left( \frac{F(x_k)}{G(x_k)} + \gamma_k \right) x_k^i = \frac{F(x_k)}{G(x_k)} x_k^i + \gamma_{i,k} \tag{1}$$

where $|\gamma_{i,k}| = O(\varepsilon)$. We use the linearity of determinants (see Theorem 6) in a similar fashion that was also used in [3]. In particular, consider a column of the matrix $A$; it consists of the evaluations of a monomial $yx^i$ on all the points $(x_1, y_1), \cdots, (x_\ell, y_\ell)$. Using Eq. (1), we can write this column as the addition of a column $C_i$ that consists of the evaluation of the rational function $\frac{F(x)}{G(x)}x^i$ on the points $x_1, \cdots, x_\ell$ and a column $\Gamma_i$ that consists of all the values $\gamma_{i,k}$ for $1 \le k \le \ell$. By the linearity of determinants, we can write the determinant of $A$ as the sum of determinants of two matrices where one matrix includes the column $C_i$ and the other has $\Gamma_i$; observe that the magnitude of the determinant of the latter matrix can be upper bounded by $O(\varepsilon)$, with hidden constants that depend on $\Delta$. By performing this operation on all the columns, we can separate all the entries involving $\gamma_{i,k}$ into separate matrices and the magnitude of sum of the determinants can be bounded by $O(\varepsilon)$.

Let $B$ be the matrix that remains after removing all the $\gamma_{i,k}$ terms. We bound $|\det(B)|$. Note that $B$ consists of row vectors

$$U = (1 \quad x \quad \cdots \quad x^{\Delta_1} \quad y \quad yx \quad \cdots \quad yx^{\Delta_G - 1}).$$

evaluated at some value $x = x_k$ and $y = \frac{F(x_k)}{G(x_k)}$ at its $k$-th row. This is equivalent to the evaluation of the following vector:

$$(1 \quad x \quad \cdots \quad x^{\Delta_1} \quad \tfrac{F}{G} \quad \tfrac{F}{G}x \quad \cdots \quad \tfrac{F}{G}x^{\Delta_G - 1}).$$

Observe that row $k$ of matrix $B$ will be evaluating $U$ on the point $x_k$. Since $G(x_k) = \Theta(1) \neq 0$, we can multiply row $k$ by $G(x_k)$ and this will only change the determinant by a constant factor. With a slight abuse of the notation, let $B$ denote the matrix after this multiplication step. Thus, the columns of $B$ now correspond to the evaluation of the following vector.

$$(G \quad Gx \quad \cdots \quad Gx^{\Delta_1} \quad F \quad Fx \quad \cdots \quad Fx^{\Delta_G - 1}).$$

Note that we can exchange columns and it will only flip the signs of the determinant of a matrix. We will focus on bounding the determinant of

$$(Gx^{\Delta_1} \quad Gx^{\Delta_1 - 2} \quad \cdots \quad G \quad Fx^{\Delta_G - 1} \quad Fx^{\Delta_G - 2} \quad \cdots \quad F).$$

The key observation is that there is a strong connection between the Sylvester matrix of $G, F$ and matrix $B$. Recall that the Sylvester matrix of $G$ and $F$ is of the form

$$\mathrm{Syl}(G, F) = \begin{bmatrix} G_{\Delta_G} & G_{\Delta_G - 1} & \cdots & G_0 & 0 & \cdots & 0 & 0 \\ 0 & G_{\Delta_G} & G_{\Delta_G - 1} & \cdots & G_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & G_{\Delta_G} & G_{\Delta_G - 1} & \cdots & G_1 & G_0 \\ F_{\Delta_F} & F_{\Delta_F - 1} & \cdots & F_0 & 0 & \cdots & 0 & 0 \\ 0 & F_{\Delta_F} & F_{\Delta_F - 1} & \cdots & F_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & F_{\Delta_F} & F_{\Delta_F - 1} & \cdots & F_1 & F_0 \end{bmatrix},$$

where $G_i$ (resp. $F_i$) is the coefficient of $x^i$ in $G$ (resp. $F$). Observe that

$$\begin{aligned}(Gx^{\Delta_F - 1} \quad Gx^{\Delta_F - 2} \quad \cdots \quad G \quad Fx^{\Delta_G - 1} \quad Fx^{\Delta_G - 2} \quad \cdots \quad F) = \\ \mathrm{Syl}(G, F) \cdot (x^{\Delta_F + \Delta_G - 1} \quad x^{\Delta_F + \Delta_G - 2} \quad \cdots \quad x \quad 1)^T,\end{aligned}$$

which means that by the linear transformation described by $\mathrm{Syl}(G, F)^{-1}$, which exists as $\mathrm{Res}(G, F) = \det(\mathrm{Syl}(G, F)) \neq 0$, we can turn the last $\Delta_F + \Delta_G$ columns in $B$ to

$$(x^{\Delta_F + \Delta_G - 1} \quad x^{\Delta_F + \Delta_G - 2} \quad \cdots \quad x \quad 1).$$

Since the remaining columns are all polynomials in $x$ and the highest degree in column $i$ is $\Delta_G + \Delta_1 - i$ for $i = 0, 1, \cdots, \Delta_F$, by using column operations, we can eliminate all lower degree terms for each column and the only term left for column $i$ is $G_{\Delta_G} x^{\Delta_G + \Delta_1 - i}$. Note that column operations do not change the determinant.

By assumption, the leading coefficients of $G$ is 1, i.e., $G_{\Delta_G} = 1$. Thus, this transforms $B$ into a Vandermonde matrix $V_B$ of size $\ell \times \ell$. By Theorem 8, $|\det(V_B)| = \Omega(\lambda^{\ell^2})$. Since multiplying the inverse of $\mathrm{Syl}(G, F)$ scales $\det(B)$ by a factor of $\Theta(|\det(\mathrm{Syl}(G, F)^{-1})|) = \Theta(|\mathrm{Res}(G, F)^{-1}|)$, we bound $|\det(B)| = |\det(V_B)|/(1/|\mathrm{Res}(G, F)|) = \Omega(|\mathrm{Res}(G, F)|\lambda^{\ell^2})$. The claim then follows from this. ◀

## 4 Lower Bounds for Flat Intersection Reporting

We are now ready to show lower bounds for flat intersection reporting. We first establish a reduction from special polynomial slab reporting problems to flat intersection reporting.

### 4.1 A Reduction from Polynomial Slab Range Reporting to Flat-hyperslab Intersection Reporting

We study the following flat intersection reporting problem.

▶ **Definition 12** (Flat-hyperslab Intersection Reporting). *In the $t$-flat-hyperslab intersection reporting problem, we are given a set $\mathcal{S}$ of $n$ $(d-t)$-dimensional hyperslabs in $\mathbb{R}^d$, i.e., regions created by a linear translation of $(d-t-1)$-flats, where $0 \le t < d$, as the input, and the goal is to preprocess $\mathcal{S}$ into a data structure such that given any query $t$-flat $\gamma$, we can output $\mathcal{S} \cap \gamma$, i.e., the set of $(d-t)$-hyperslabs intersecting the query $t$-flat, efficiently.*

First, observe that any $t$-flat that is not parallel to any of the axes can be formulated as

$$\begin{bmatrix} a_{0,1} & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ a_{1,1} & a_{1,2} & \cdots & a_{1,t} & a_{1,t+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{d-t,1} & a_{d-t,2} & \cdots & a_{d-1,t} & a_{d-t,t+1} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix},$$

where $a_{i,j}$'s are the parameters defining the $t$-flat, and $\tau_1, \cdots, \tau_t$ are the free variables that generate points in the $t$-flat. Note that we only need $(d-t)(t+1)$ independent $a_{i,j}$'s to define a $t$-flat.

On the other hand, we consider $(d-t)$-hyperslabs of form

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & b_{1,1} & b_{1,2} & \cdots & b_{1,d-t} \\ 0 & 1 & \cdots & 0 & b_{2,1} & b_{2,2} & \cdots & b_{2,d-t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{t,1} & b_{t,2} & \cdots & b_{t,d-t} \\ 0 & 0 & \cdots & 0 & b_{t+1,1} & b_{t+1,2} & \cdots & b_{t+1,d-t} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1+w \end{bmatrix},$$

where $b_{i,j}$'s are the parameters defining a $(d-t-1)$-flat, and parameter $w \in [0, w_0]$ adds one extra dimension to the flat to make it $(d-t)$-dimensional; in essence, we will be considering all the $(d-t-1)$-flats for all $w \in [0, w_0]$ which will turn it into a $(d-t)$-hyperslab.

Therefore, the intersection of a $t$-flat and a $(d-t)$-hyperslab must be a solution to

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & b_{1,1} & b_{1,2} & \cdots & b_{1,d-t} \\ 0 & 1 & \cdots & 0 & b_{2,1} & b_{2,2} & \cdots & b_{2,d-t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{t,1} & b_{t,2} & \cdots & b_{t,d-t} \\ 0 & 0 & \cdots & 0 & b_{t+1,1} & b_{t+1,2} & \cdots & b_{t+1,d-t} \end{bmatrix} \cdot \begin{bmatrix} a_{0,1} & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ a_{1,1} & a_{1,2} & \cdots & a_{1,t} & a_{1,t+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{d-t,1} & a_{d-t,2} & \cdots & a_{d-1,t} & a_{d-t,t+1} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1+w \end{bmatrix}.$$

Multiplying the two matrices, we obtain the following system

$$\begin{bmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1+w \end{bmatrix}.$$

We denote this linear system by $A\boldsymbol{\tau} = \mathbf{s}$ and assume

$$\det(A) \neq 0 \tag{2}$$

which is the case when the $t$-flat and the $(d-t)$-hyperslab properly intersect, and this system has a solution iff the last entry of the solution vector is 1. So by Cramer's rule, we have

$$1 = \frac{\begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & 0 \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & -1+w \end{vmatrix}}{\begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} \end{vmatrix}}.$$

By the linearity of determinants, we have

$$0 = \begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & 1 + \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} - w \end{vmatrix}. \tag{3}$$

Consider the value of the above determinant using Leibniz formula for determinants, which is the sum of $(t+1)!$ terms. Consider the terms that have at most 1 factor of $b_{i,j}$; these can only come from the diagonals. Thus, any $t$-flat parameterized by $\mathbf{a} = (a_{i,j})$ intersects a query $(d-t)$-hyperslab parameterized by $\mathbf{b} = (b_{i,j})$ if and only if

$$0 = a_{0,1} + a_{0,1} \sum_{j=2}^{t+1} \sum_{i=1}^{d-1} a_{i,j}b_{j,i} + \sum_{i=1}^{d-1} a_{i,1}b_{1,i} + E(\mathbf{a},\mathbf{b}) + f(\mathbf{a},\mathbf{b},w) \quad = P(\mathbf{a},\mathbf{b}) + f(\mathbf{a},\mathbf{b},w),$$

where $E(\mathbf{a},\mathbf{b})$ contains the sum of products of at least two distinct $a_{i_1,i_2}b_{i_3,i_1}$ and $f(\mathbf{a},\mathbf{b},w)$ is a polynomial with factor $w$.

Note that after fixing $\mathbf{a}, \mathbf{b}$, $f(\mathbf{a}, \mathbf{b}, w)$ is a polynomial in $w$ and we assume that

$$\frac{\partial f(\mathbf{a},\mathbf{b},w)}{\partial w} = -\begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t,i} & \cdots & 1 + \sum_{i=1}^{d-t} a_{i,t}b_{t,i} \end{vmatrix} < 0. \tag{4}$$

This implies the following lemma.

▶ **Lemma 13.** *Assuming* $\mathbf{a}, \mathbf{b}$ *satisfying Assumptions (2) and (4), for any fixed* $\mathbf{a}$*, there is a* $\mathbf{b}$ *such that* $0 \le P(\mathbf{a}, \mathbf{b}) \le -f(\mathbf{a}, \mathbf{b}, w_0)$ *if and only if there is some* $w \in [0, w_0]$ *such that* $P(\mathbf{a}, \mathbf{b}) + f(\mathbf{a}, \mathbf{b}, w) = 0$*.*

**Proof.** Since $f(\mathbf{a}, \mathbf{b}, w)$ is a polynomial in $w$ and $\frac{\partial f}{\partial w} < 0$, $f(\mathbf{a}, \mathbf{b}, w)$ is continuous and decreasing in $[0, w_0]$. Furthermore $f(\mathbf{a}, \mathbf{b}, 0) = 0$ as $w$ is a factor of $f$. The lemma follows. ◀

Fixing $\mathbf{a}$ in $P(\mathbf{a}, \mathbf{b})$, we obtain a polynomial in $\mathbf{b}$. Let $(P(\mathbf{a}, \mathbf{b}), f(\mathbf{a}, \mathbf{b}, w_0)) = \{\mathbf{b} : 0 \le P(\mathbf{a}, \mathbf{b}) \le -f(\mathbf{a}, \mathbf{b}, w_0)\}$ be a polynomial slab. This essentially establishes a reduction between polynomial slab reporting and flat intersection reporting.

▶ **Corollary 14.** *Assuming* $\mathbf{a}, \mathbf{b}$ *satisfying Assumptions (2) and (4), for any fixed* $\mathbf{a}$*, there is a* $\mathbf{b}$ *such that* $\mathbf{b} \in (P(\mathbf{a}, \mathbf{b}), f(\mathbf{a}, \mathbf{b}, w_0))$ *if and only if a* $t$*-flat parameterized by* $\mathbf{a}$ *intersects a* $(d-t)$*-hyperslab of width* $w_0$ *parameterized by* $\mathbf{b}$*.*

## 4.2 Lower Bounds for Flat-hyperslab Intersection Reporting

We are now ready to prove the lower bounds. We show lower bounds for 1-flat-hyperslab intersection reporting in $\mathbb{R}^d$ and 2-flat-hyperslab intersection reporting in $\mathbb{R}^4$.

First observe that by setting $t = 1$ in Eq. (3) and using Corollary 14 a polynomial slab reporting problem with polynomial

$$\begin{aligned} P_1(\mathbf{a}, \mathbf{b}) &= a_{0,1} + a_{0,1}\sum_{i=1}^{d-1} a_{i,2}b_{2,i} + \sum_{i=1}^{d-1} a_{i,1}b_{1,i} + \sum_{i,j=1 \wedge i \ne j}^{d-1} (a_{i,1}a_{j,2} - a_{j,1}a_{i,2})b_{1,i}b_{2,j} \\ &= b_{1,1}G_1(b_{2,2}) + F_1(b_{2,2}), \end{aligned} \tag{5}$$

reduces to a line-hyperslab intersection reporting problem, where to get $G_1$, we have collected all the monomials that have $b_{1,1}$ in them and then we have factored $b_{1,1}$ out and we are considering it as a polynomial of $b_{2,2}$ (all the other variables are considered "constant"). $F_1$ is defined similarly by considering the remaining terms as a function of $b_{2,2}$. Observe that the polynomial does not have any term with degree 3. Let $G_1 = g_{1,1}b_{2,2} + g_{1,0}$ and $F_1 = f_{1,1}b_{2,2} + f_{1,0}$.

Similarly, polynomial slab reporting with

$$\begin{aligned} P_2(\mathbf{a}, \mathbf{b}) &= a_{0,1} + a_{0,1}\sum_{j=1}^{2}\sum_{i=2}^{3} a_{j,i}b_{i,j} + \sum_{j=1}^{2} a_{j,1}b_{1,j} \\ &\quad + a_{0,1}\sum_{j,l=1 \wedge j \ne l}^{2} (a_{j,2}a_{l,3} - a_{j,3}a_{l,2})b_{2,j}b_{3,l} + \sum_{j,l=1 \wedge j \ne l}^{2}\sum_{k=2}^{3}(a_{j,1}a_{l,k} - a_{j,k}a_{l,1})b_{1,j}b_{k,l} \\ &= b_{1,1}G_2(b_{2,2}) + F_2(b_{2,2}) \end{aligned} \tag{6}$$

reduces to 2-flat-hyperslab intersection reporting in $\mathbb{R}^4$ where $G_2, F_2$ are defined similarly as $G_1, F_1$.

For the moment, we focus on the case of line-hyperslab intersection reporting but the same applies also to 2-flat-hyperslab intersection reporting in $\mathbb{R}^4$ since the polynomials $F_2$ and $G_2$ involved in the definition of Eq. (6) are quite similar to Eq. (5).

Here, we will use our techniques from Section 3. The general idea is that we will use Corollary 5, to reduce the $2(d-1)$-variate polynomials $P_1$ and $P_2$ into bivariate polynomials on $b_{1,1}$ and $b_{2,2}$. Then, the variable $b_{1,1}$ will be our $y$ variable and $b_{2,2}$ will be the $x$ variable in Section 3, and $G_1$ and $F_1$ here will play the same role as in that section. We will set

$$a_{1,1} = \frac{1 + a_{1,2}a_{2,1}}{a_{2,2}} \tag{7}$$

which will ensure that the leading coefficient of $G_1$ is 1. This is our normalization step, since we can divide the equations defining the intersection (and thus polynomials $P_1$ and $P_2$) by any constant. Eventually, the resultant of the polynomials $F_1$ and $G_1$ will play an important role. Observe that the resultant is

$$\mathrm{Res}(G_1, F_1) = \begin{vmatrix} 1 & g_0 \\ f_1 & f_0 \end{vmatrix} = f_0 - g_0 f_1. \tag{8}$$

## 4.3 Construction of Input Points and Queries

Now we are ready to describe our input and query construction. Assume we have a data structure that uses $S(n)$ space and has the query time $Q(n) + O(k)$ where $k$ is the output size; for brevity we use $Q = Q(n)$.

We will start with a fixed line and a fixed hyperslab and then build the queries and inputs very close to these two fixed objects. However, we require a certain "general position" property with respect to these two fixed objects.

Recall that Eq. (5) refers to the condition of whether a (query) line described by $\mathbf{a}$ variables intersects a $(d-2)$-dimensional flat described by the $\mathbf{b}$ variables (which corresponds to setting the variable $w$ to zero). Consider a fixed flat and a fixed line. To avoid future confusion, let $\mathbf{A}$ and $\mathbf{B}$ refer to this fixed line and flat. We require the following.

- $\mathbf{A}$ and $\mathbf{B}$ must intersect properly (i.e., the line is not contained in the flat). Observe that it implies that when we consider $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in $\mathbf{b}$ variables, $\mathbf{B}$ does not belong to the zero set of $P_1(\mathbf{A}, \mathbf{b})$. Note that this satisfies Assumption (2).
- The polynomial $P_1(\mathbf{A}, \mathbf{b})$ (as a polynomial in $\mathbf{b}$) is irreducible. This is true as long as $\mathbf{A}$ is chosen so that no coefficient in $P_1$ is zero. To see this, note that $P_1$ is a polynomial in $\mathbf{b}$ and any variable $b_{i,j}$ has degree 1. Suppose for the sake of contradiction that $P_1$ is reducible, then the factorization must be of the form

  $$P_1(\mathbf{A}, \mathbf{b}) = \left(c_{10} + \sum_{i=1}^{d-1} c_{1i} b_{1i}\right) \cdot \left(c_{20} + \sum_{i=1}^{d-1} c_{2i} b_{2i}\right),$$

  for nonzero coefficients $c_{10}, c_{20}, c_{1i}, c_{2i}$. Then by Eq. (5),
  1. $a_{0,1} = c_{10}c_{20}$,
  2. $\forall i = 1, \cdots, d-1 : a_{0,1}a_{i,2} = c_{10}c_{2i}$,
  3. $\forall i = 1, \cdots, d-1 : a_{i,1} = c_{1i}c_{20}$,
  4. $\forall i, j = 1, 2, \cdots, d-1 : a_{i,1}a_{j,2} - a_{j,1}a_{i,2} = c_{1i}c_{2i}$.

  However, for these conditions to hold, all coefficients of $P_1$ must be zero, a contradiction.
- Observe that the irreducibility of $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in $\mathbf{b}$ implies that it has only finitely many points where the tangent hyperplane at those points is parallel to some axis. We assume $\mathbf{B}$ is not one of those points.

- The irreducibility of $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in $\mathbf{b}$ can be used to satisfy Assumption (4) since the corresponding polynomial of the determinant involved in Assumption (4) can only have $\Theta(1)$ many common roots with $P_1(\mathbf{A}, \mathbf{b})$.
- Finally, since the polynomial $P_1(\mathbf{A}, \mathbf{b})$ is irreducible and since $\mathrm{Res}(G_1, F_1)$ is also of degree 2 in $\mathbf{b}$ variables, it follows that $\mathrm{Res}(G_1, F_1)$ is algebraically independent of $P_1(\mathbf{A}, \mathbf{b})$. This means that there are only finitely many places where both polynomials are zero, meaning, we can additionally assume that Eq. (8) is non-zero (when evaluated at $\mathbf{B}$).

Consider two parameters $\varepsilon_p$ and $\varepsilon_q = \varepsilon_p/C$ where $C$ is a large enough constant and $\varepsilon_p$ is a parameter to be set later. Consider the parametric space of the input objects, where the variable $\mathbf{b}$ defines a single point. In such a space, $\mathbf{B}$ defines a single point. Place an axis-aligned cube $\boldsymbol{R}$ of side-length $\varepsilon_p$ centered around $\mathbf{B}$. The input slabs are defined by placing a set of $n$ random points inside $\boldsymbol{R}$. Each point in $\boldsymbol{R}$ defines a $(d-2)$-dimensional flat. We set $w = \Theta(\frac{Q}{n})$ which in turn defines a "narrow $(d-1)$-hyperslab".

We now define the set of queries. Notice that $P_1$ has exactly $2(d-1)$ algebraically independent coefficients; these are the coefficients of linear terms involved plus $a_{0,1}$; recall that by Eq. (7), $a_{1,1}$ was fixed as a function of $a_{1,2}a_{2,1}$ and $a_{2,2}$ but we still have $a_{0,1}$ as a free parameter. These $2(d-1)$ coefficients define another parametric space, where $\mathbf{A}$ denotes a single point. Place a $2(d-1)$-dimensional hypercube of side length $\varepsilon_q$ and then subdivide it into a grid where the side-length of every cell is $\tau$. Every grid point now defines a different query. Let $\mathcal{Q}$ be the set of all the queries we have constructed.

Notice that a query defined by a point $\mathbf{a} \in \mathcal{Q}$ defines a line in the primal space, but when considered in the parametric space $\boldsymbol{R}$, it corresponds to a manifold (zeroes of a degree two multilinear polynomial) that includes the set of points that correspond to $(d-2)$-dimensional flats that pass through the line in the primal space. The variable $w$ allows us to turn it to a range reporting problem where we need to output any $(d-2)$-dimensional flat that passes within $w$ vertical distance of the query line. The following observations and lemmas are the important geometric properties that we require out of our construction.

▶ **Observation 15.** *For two different queries $\mathbf{a}_1$ and $\mathbf{a}_2$, the polynomials $P_1(\mathbf{a}_1, \mathbf{b})$ and $P_1(\mathbf{a}_2, \mathbf{b})$ differ by at least $\tau$ in at least one of their coefficients.*

▶ **Observation 16.** *Consider a line $f$ parallel to an axis. For small enough $\varepsilon_p$, and any $\mathbf{a} \in \mathcal{Q}$, the function $P_1(\mathbf{a}, \mathbf{b})$ evaluated on the line $f$ is such that the magnitude of its derivative is bounded by $\Omega(1)$.*

**Proof.** Recall that $\mathbf{B}$ was chosen such that the manifold corresponding to $\mathbf{A}$ does not have a tangent parallel to any of the axes at point $\mathbf{B}$ and thus the derivate of the function $P_1(\mathbf{A}, \mathbf{B})$ is non-zero at $\mathbf{B}$. The lemma then follows since $\varepsilon_p$ and $\varepsilon_q$ are small enough and $P_1(\mathbf{A}, \mathbf{B})$ is a continuous function w.r.t any of its variables. ◀

Let $\mathrm{Vol}'(\boldsymbol{R})$ be the $(d-1)$-dimensional volume of $\boldsymbol{R}$, i.e., the volume of the projection of $\boldsymbol{R}$ to any of its $(d-1)$-dimensional subspace.

▶ **Observation 17.** *The intersection volume of the range defined by a query $\mathbf{a}$ and $\boldsymbol{R}$ is $\Theta(w\, Vol'(\boldsymbol{R}))$ if $C$ in the definition of $\varepsilon_q$ is large enough, for $w \le \varepsilon_p$.*

**Proof.** Observe that the query manifold defined by $\mathbf{A}$ passes through the center, $\mathbf{B}$, of $\boldsymbol{R}$ by construction. Since each coordinate of $\mathbf{a}$ differs from $\mathbf{A}$ by at most $\varepsilon_q$, it thus follows that by setting $C$ large enough, we can ensure that the distance between $\mathbf{B}$ and $\mathbf{a}$ is less than $\varepsilon_p/2$. Also observe that the width of the range along any axis will be $\Theta(w)$. The claim now follows by integrating the volume over vertical lines using Observation 16. ◀

▶ **Lemma 18.** *Consider a query* $\mathbf{a} \in \mathcal{Q}$ *and let* $\mathsf{r}$ *be the range that represents* $\mathbf{a}$ *in the parametric space defined by* $\boldsymbol{R}$*. Consider an interval* $\mathscr{I}$ *on the* $i$*-th side of* $\boldsymbol{R}$*, for some* $i$*. Let* $\mathsf{r}_{\mathscr{I}}$ *be the subset of* $\mathsf{r}$ *whose projection on the* $i$*-th side of* $\boldsymbol{R}$ *falls inside* $\mathscr{I}$*. Then, the volume of* $\mathsf{r}_{\mathscr{I}}$ *is* $O(\mathit{Vol}'(\boldsymbol{R})w|\mathscr{I}|/\varepsilon_p)$*.*

**Proof.** Both claims follow through Observation 16 by integrating the corresponding volumes over lines parallel to axes. ◀

## 4.4 Using the Framework

Observe that by the above Observation 17, setting $w = \Theta(\frac{Q}{n}\varepsilon_p)$ satisfies Condition 1 of the lower bound framework in Theorem 1.

Satisfying Condition 2 requires a bit more work however. To do that, consider two queries defined by points $\mathbf{a}_1$ and $\mathbf{a}_2$. Let $\mathsf{r}_1$ and $\mathsf{r}_2$ be the two corresponding ranges in the parametric space of $\boldsymbol{R}$.

To satisfy Condition 2, assume for contradiction that the volume of $\mathsf{r}_1 \cap \mathsf{r}_2$ is large, i.e., $\omega(\mathrm{Vol}(\boldsymbol{R})/(n\psi))$ where $\psi = 2^{\sqrt{\log n}}$. We now combine Observation 15, and Corollary 5 with parameter $\vartheta$ set to $\varepsilon_0 \frac{\varepsilon_p}{Q\psi}$ where $\varepsilon_0$ is a small enough constant and where $X_1$ represents $b_{1,1}$, $X_2$ represents $b_{2,2}$ and the remaining indeterminates represent the rest of variables in $\mathbf{b}$; note that the value of $d$ in Corollary 5 is $\beta = 2(d-1)$ and $\mathcal{U} = \binom{2+1}{2} = 3$. Observe that each interval $\mathscr{I}_i$ determined by Corollary 5 defines a slab parallel to the $i$-th axis in $\boldsymbol{R}$; let $\boldsymbol{R}_{\mathrm{bad}}$ be the union of these slabs. By Lemma 18, and choice of small enough $\varepsilon_0$, a positive fraction of the intersection volume of $\mathsf{r}_1$ and $\mathsf{r}_2$ must lie outside $\boldsymbol{R}_{\mathrm{bad}}$. In addition, Corollary 5 allows us to pick some fixed values for all variables in $\mathbf{b}$, except for $b_{1,1}$ and $b_{2,2}$ with the property the final polynomials $H_1$ and $H_2$ (on indeterminates $b_{1,1}$ and $b_{2,2}$) that we obtain have the property that they have at least one coefficient which differs by

$$\Omega\left(\tau\left(\varepsilon_0\frac{\varepsilon_p}{Q\psi}\right)^{3(\beta-2)}\right) \tag{9}$$

between them; we call this operation of plugging values for all $\mathbf{b}$ except for $b_{1,1}$ and $b_{2,2}$ *slicing*. After slicing, we are reduced to the bivariate case; consider the set of points on which both $H_1$ and $H_2$ have value $O(w)$. If the 1D interval length of such points is $O(\varepsilon_p/(Q\psi)$, we call this a *good* slice, otherwise a *bad* slice. By Lemma 18, there must be bad slices since if all the slices are good, by integration of the intersection area of $\mathsf{r}_1$ and $\mathsf{r}_2$ over all the remaining variables in $\mathbf{b}$, $\mathsf{r}_1$ and $\mathsf{r}_2$ intersect with volume $O(\mathrm{Vol}(\boldsymbol{R})/(n\psi))$, a contradiction.

We now show that we can arrive at a contradiction, assuming the existence of a bad slice. Given a bad slice, and any constant $\ell$, we can find $\ell$ points $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ such that $|x_{k_1} - x_{k_2}| = \omega(\varepsilon_p/(Q\psi))$ for all $1 \le k_1 < k_2 \le \ell$ and that $H_1(x_k, y_k), H_2(x_k, y_k) = O(w)$ for all $k \in \{1, 2 \cdots, \ell\}$. Observe that $H_i(x, y)$ has only monomials $y$, $x$, $xy$ and a constant term. The critical observation here is that the coefficient of the monomial $xy$ is always 1 since the coefficient of the monomial $b_{1,1}b_{2,2}$ was 1 and there was no monomial of degree three in $P_1$, meaning, after slicing this coefficient will not change. We pick $\ell = 3$ and thus we tweak all the three other coefficients of $H_1$. Tweaking $H_1$ such that $\tilde{H}_1(x_k, y_k) = H_2(x_k, y_k)$ corresponds to solving a linear system of equations that come from evalutions of monomials $X$, $Y$, and a constant term at points $(x_k, y_k)$. We can thus use Lemma 11 with $\Delta_1 = \Delta_F = 1$, $\lambda = \omega(\varepsilon_p/(Q\psi))$. Observe that $\mathrm{Res}(G, F)$ here is a constant by the properties of our construction. Also observe that by Lemma 11, the magnitude of the determinant of matrix $A$ defined in Lemma 11 is

$$\omega\left((\varepsilon_p/(Q\psi)))^9\right).$$

By the same argument in [3], this means that the tweaking operation can be done such that each coefficient of $H_1$ is changed by

$$o\left(\left(\varepsilon_p/(Q\psi)\right)^{-9} w\right). \tag{10}$$

We observe that after tweaking, $\tilde{H}_1$ and $H_2$ must coincide since by Lemma 11, the determinant of the relevant monomials is non-zero and thus there's a unique polynomial that passes through points $(x_1, y_1), \cdots, (x_\ell, y_\ell)$. Finally, to get a contradiction, we simply need to ensure that Eq. (10) is asymptotically smaller than Eq. (9). This yields a bound for the value of $\tau$,

$$\tau = \Theta\left(w(Q\psi)^{3(\beta-2)+9}\right) = \Theta\left(w(Q\psi)^{3\beta+3}\right) \tag{11}$$

where we have assumed that $\varepsilon_p$, and $\varepsilon_0$ are small enough constants that have been absorbed in the $\Theta(\cdot)$ notation. Thus, this choice of $\tau$ will make sure that Condition 2 of the framework is also satisfied. It remains to calculate the number of queries that have been generated. Observe that $\tau$ was the side-length of a small enough grid around the point $\mathbf{A}$ in a $\beta$-dimensional space. Thus, the number of queries we generated is

$$m = \overset{o}{\Omega}\left(\left(\frac{1}{\tau}\right)^{\beta}\right) = \overset{o}{\Omega}\left(\frac{n^{\beta}}{Q^{\beta(3\beta+4)}}\right). \tag{12}$$

Applying Theorem 1 yields a space lower bound of

$$S(n) = \overset{o}{\Omega}(mQ) = \overset{o}{\Omega}\left(\frac{n^{2(d-1)}}{Q^{4(3d-1)(d-1)-1}}\right) \tag{13}$$

for line-hyperslab intersection reporting since $\beta = 2(d-1)$. One can verify that the same argument works for triangle-triangle intersection reporting in $\mathbb{R}^4$, since $P_2$ is also a multilinear polynomial of degree two. In this case, $\beta = 6$ which yields a space lower bound of

$$S(n) = \overset{o}{\Omega}\left(\frac{n^{6}}{Q^{125}}\right). \tag{14}$$

To sum up, we obtain the following results:

▶ **Theorem 19.** *Any data structure that solves line-hyperslab intersection reporting in* $\mathbb{R}^d$ *must satisfy a space-time tradeoff of* $S(n) = \overset{o}{\Omega}\left(\frac{n^{2(d-1)}}{Q(n)^{(4(3d-1)(d-1)-1}}\right)$.

▶ **Theorem 20.** *Any data structure that solves triangle-triangle intersection reporting in* $\mathbb{R}^4$ *must satisfy a space-time tradeoff of* $S(n) = \overset{o}{\Omega}\left(\frac{n^{6}}{Q(n)^{125}}\right)$.

## 5 Conclusion and Open Problems

We study line-hyperslab intersecting reporting in $\mathbb{R}^d$ and triangle-triangle intersecting reporting in $\mathbb{R}^4$. We show that any data structure with $n^{o(1)} + O(k)$ query time must use space $\overset{o}{\Omega}(n^{2(d-1)})$ and $\overset{o}{\Omega}(n^6)$ for the two problems respectively. This matches the classical upper bounds for the small $n^{o(1)}$ query time case for the two problems and answer an open problem for lower bounds asked by Ezra and Sharir [20]. Along the way, we generalize and develop the lower bound technique used in [2, 3].

The major open problem is how to show a lower bound for general intersection reporting between objects of $t$ and $(d-t)$ dimensions or for flat semialgebraic objects as studied recently in [5]. Many of our techniques work, however, one big challenge is that after applying Corollary 5, the leading coefficient changes and thus we can no longer guarantee big gaps between coefficients.

## References


1 Peyman Afshani. Improved pointer machine and I/O lower bounds for simplex range reporting and related problems. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, SoCG '12, pages 339–346, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2261250.2261301.

2 Peyman Afshani and Pingan Cheng. Lower bounds for semialgebraic range searching and stabbing problems. In *37th International Symposium on Computational Geometry*, volume 189 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 8, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021.

3 Peyman Afshani and Pingan Cheng. On semialgebraic range reporting. In *38th International Symposium on Computational Geometry*, volume 224 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Paper No. 3, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. doi:10.4230/lipics.socg.2022.3.

4 Pankaj K. Agarwal. Simplex range searching and its variants: a review. In *A journey through discrete mathematics*, pages 1–30. Springer, Cham, 2017.

5 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. In *38th International Symposium on Computational Geometry*, volume 224 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Paper No. 4, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. doi:10.4230/lipics.socg.2022.4.

6 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. Efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50(2):760–787, 2021. doi:10.1137/19M1268550.

7 Pankaj K. Agarwal and Jirí Matousek. On range searching with semialgebraic sets. *Discret. Comput. Geom.*, 11:393–418, 1994. doi:10.1007/BF02574015.

8 Pankaj K. Agarwal and Jiří Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22(4):794–806, 1993. doi:10.1137/0222051.

9 Pankaj K. Agarwal, Jiří Matoušek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM J. Comput.*, 42(6):2039–2062, 2013. doi:10.1137/120890855.

10 Pankaj K. Agarwal and Micha Sharir. Ray shooting amidst convex polyhedra and polyhedral terrains in three dimensions. *SIAM J. Comput.*, 25(1):100–116, 1996. doi:10.1137/S0097539793244368.

11 Boris Aronov, Mark de Berg, and Chris Gray. Ray shooting and intersection searching amidst fat convex polyhedra in 3-space. *Comput. Geom.*, 41(1-2):68–76, 2008. doi:10.1016/j.comgeo.2007.10.006.

12 Timothy M. Chan. Optimal partition trees. *Discrete Comput. Geom.*, 47(4):661–690, 2012. doi:10.1007/s00454-012-9410-z.

13 Bernard Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2(4):637–666, 1989. doi:10.2307/1990891.

14 Bernard Chazelle. Lower bounds for orthogonal range searching. I. The reporting case. *J. Assoc. Comput. Mach.*, 37(2):200–212, 1990. doi:10.1145/77600.77614.

15 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, December 1993. doi:10.1007/BF02189314.

16 Bernard Chazelle and Burton Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom.*, 5(5):237–247, 1996. doi:10.1016/0925-7721(95)00002-X.

17 M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. *Algorithmica*, 12(1):30–53, 1994. doi:10.1007/BF01377182.

18 Mark de Berg and Chris Gray. Vertical ray shooting and computing depth orders for fat objects. *SIAM J. Comput.*, 38(1):257–275, 2008. doi:10.1137/060672261.

19 Esther Ezra and Micha Sharir. Intersection searching amid tetrahedra in four dimensions. *CoRR*, abs/2208.06703, 2022. doi:10.48550/arXiv.2208.06703.

20 Esther Ezra and Micha Sharir. On ray shooting for triangles in 3-space and related problems. *SIAM J. Comput.*, 51(4):1065–1095, 2022. doi:10.1137/21M1408245.

21 Larry Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Cambridge Philos. Soc.*, 159(3):459–469, 2015. doi:10.1017/S0305004115000468.

22 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Ann. of Math. (2)*, 181(1):155–190, 2015. doi:10.4007/annals.2015.181.1.2.

23 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993. doi:10.1007/BF02573972.

24 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. doi:10.1007/s00454-015-9701-2.

25 Jiří Matoušek and Otfried Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10(2):215–232, 1993. doi:10.1007/BF02573975.

26 M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9(5):471–494, 1993. doi:10.1007/BF01187036.

27 Marco Pellegrini. Stabbing and ray shooting in 3 dimensional space. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 177–186. ACM, 1990. doi:10.1145/98524.98563.

28 Marco Pellegrini. Ray shooting and lines in space. In *Handbook of discrete and computational geometry (3rd Edition)*, CRC Press Ser. Discrete Math. Appl., pages 1093–1112. CRC, Boca Raton, FL, 2017.

29 Edgar A. Ramos. On range reporting, ray shooting and $k$-level construction. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry (Miami Beach, FL, 1999)*, pages 390–399. ACM, New York, 1999. doi:10.1145/304893.304993.

30 Micha Sharir and Hayim Shaul. Ray shooting and stone throwing with near-linear storage. *Comput. Geom.*, 30(3):239–252, 2005. doi:10.1016/j.comgeo.2004.10.001.