arXiv:2305.03180v1 [cs.CG] 4 May 2023

关于范围摘要查询

Peyman Afshani ✉丹

麦奥胡斯大学

PinganCheng ✉奥胡斯

大学，丹麦

Aniket Basu Roy ✉

奥胡斯大学，丹麦

ZheweiWei !

中国人民大学，中国

摘要

我们研究近似重频项和分位数问题的查询版本。在前一个问题中，输入是一个参数 ε 和一个包含 n 个点的集合 P，这些点位于 $\mathbb{R}^d$ 中，且每个点都被赋予一个来自集合 C的颜色。目标是构建一种数据结构，使得对于给定的任意几何范围 γ，我们都能高效地找到 $\gamma \cap P$中的近似重频项列表，即在 $\gamma \cap P$中至少出现$\varepsilon|\gamma \cap P|$次的颜色，以及它们的频率，且加性误差为 $\varepsilon|\gamma \cap P|$。在后一个问题中，每个点都被赋予一个来自全序全集的权重，查询必须输出一个序列 S，其中包含 $1 + 1/\varepsilon$ 个权重，使得 S 中的第 i个权重具有近似秩 $i\varepsilon|\gamma \cap P|$，即秩 $i\varepsilon|\gamma \cap P|$ 的加性误差不超过 $\varepsilon|\gamma \cap P|$。此前，最优结果仅在一维 [35] 中已知，而在高维 [5, 8]中已有少数次优方法。

我们研究两类重要几何范围上的问题：三维半空间查询和三维支配查询。已知许多其他重要查询都可以归约为这两类问题，例如一维区间刺穿或区间包含、二维三边查询、二维圆形最近邻查询以及二维k最近邻查询。我们考虑实数 RAM 模型，其中还提供大小为 w 位、 $w = \Theta(\log n)$的整数寄存器。对于支配查询，我们为重频项和分位数问题都给出了最优解：使用线性空间，两类查询都能在$O(\log n + 1/\varepsilon)$时间内得到回答。注意，由于输出大小为 $\frac{1}{\varepsilon}$，在投入初始的 $O(\log n)$搜索时间后，我们的数据结构平均只需 $O(1)$时间就能找到一个重频项或分位数！对于更一般的半空间重频项查询，在三维（分别为二维）中，只需额外增加一个 $\log_w \frac{1}{\varepsilon}$（分别为 $\log\log_w \frac{1}{\varepsilon}$）因子，就能达到同样最优的查询时间，同时增加空间开销。通过在时间和空间上都增加额外的 $\log^{O(1)} \frac{1}{\varepsilon}$ 因子，我们还可以支持分位数查询。

我们备注道，除非在几何近似理论的数据结构方面取得重大进展，否则对于4维或更高维度，实现类似的查询界限是毫无希望的。

2012年 ACM 学科分类计算理论 → 随机性、几何和离散结构 →计算几何

关键词和短语 计算几何、范围搜索、数据结构与算法

数字对象标识符 10.4230/LIPIcs...

1 引言

范围搜索是计算几何中一个古老且基础的领域，研究如何将包含 n 个（可能带有权重的）点的输入集 $P \subset \mathbb{R}^d$ 存储在数据结构中，以便给定查询范围 γ 时，能够回答有关其中点子集的某些问题。范围搜索通常在一个通用框架下介绍，该框架能够回答各种各样的问题。例如，如果已为 P 中的点赋予整数或实数权重，那么就可以统计 γ 中的点（范围计数），计算 γ 中所有点的权重之和（加权范围计数），或者求出 γ 中的最大或最小权重（范围最大值或最小值查询）。



©Peyman Afshani、Pingan Cheng、Aniket Basu Roy、Zhewei Wei；根据知识共享署名 4.0 国际许可协议授权。莱布尼茨国际信息学会议录，达格施图尔城堡——莱布尼茨信息学中心，达格施图尔出版社，德国。

然而，有些重要问题无法在这一通用框架下得到解答。考虑下面这个引例：我们的数据包含一座城市中各栋房屋的位置及其估算价值。给定查询范围 γ，我们想了解 γ 内房屋价值的分布，例如，可能想知道房屋价值是否存在严重不平等。通过经典结果，我们可以找出最贵和最便宜的房屋（最大值和最小值查询），并计算房屋的平均价值（用价值的加权和除以 γ 中房屋的总数）。遗憾的是，这些信息无法充分反映 γ 内房屋价值的分布；例如，我们无法计算基尼指数，而它是衡量分布不平等程度的一种广泛使用的指标。理想情况下，若要了解 γ 内数值的精确分布，就必须掌握 γ 内的所有数值；在文献中，这被称为范围报告查询，它会报告查询范围 γ 内的所有点。然而，这可能是一项开销很大的操作，例如，当查询范围包含输入点中固定比例的点时，耗时可能达到 $\Omega(n)$。一种合理的替代方案是提出“摘要”查询，用以概括分布。事实上，流式处理文献中有许多重要的摘要概念，用于以近似但高精度的方式简洁表示大型数据流。计算 ε 分位数可视为对分布进行简洁近似的最重要概念之一，而且它还推广了许多熟悉的概念，例如 0 分位数、0。5 分位数和 1 分位数，它们也分别称为 S 的最小值、中位数和最大值。下面给出正式定义。

分位数摘要。给定一个值序列 $w_1 \leq \cdots \leq w_k$，对于 $0 \leq \delta \leq 1$，δ-分位数是指秩为$\lfloor \delta k \rfloor$的值。按照惯例，0-分位数和 1-分位数分别设为最小值和最大值，即 w_1 和 w_k。因此，$i = 0, \cdots \varepsilon^{-1}$时，$\varepsilon$-分位数摘要定义为由 $1 + \varepsilon^{-1}$ 个值组成的列表，其中第 i 个值是 $i\varepsilon$-分位数。正如我们稍后将要回顾的，计算精确分位数通常代价过高，因此我们转而关注近似值。我们将近似 ε-分位数摘要（近似分位数总结，AQS）定义为由 $1 + \varepsilon^{-1}$ 个值组成的序列，其中第 i 个值介于（$i - 1$）-分位数和（$i + 1$）-分位数之间，$i = 0, \cdots \varepsilon^{-1}$时皆如此。选择合理且较小的 ε，近似分位数摘要便能很好地近似分布，例如参见图 1。它还有一个好处：无论查询范围内有多少个点，查询只需输出 $O(\varepsilon^{-1})$个值。

为了获得相对精确的分布近似，ε 需要取足够小，因此我们将其视为一个额外参数（而非常数）。这也与流式处理领域的相关文献类似，在那里，对 ε 的依赖至关重要。

1.1 问题定义、先前工作与相关结果

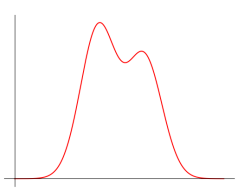
我们的一个主要问题是如何回答近似分位数摘要（AQS）查询，定义如下。

▶**问题 1**（近似分位数摘要）。考虑一个包含 n 个点的输入集 P，这些点位于 $\mathbb{R}^d$ 中，每个点 $p \in P$ 都被赋予一个来自全序全集的权重 w_p。给定一个值 ε，我们需要构建一种数据结构，使其能够针对给定的查询范围 γ，高效地返回 $P \cap \gamma$ 的近似分位数摘要（*AQS*）。

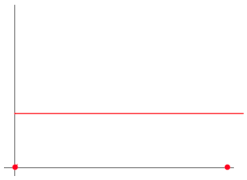
事实证明，另一类“范围摘要查询”对于构建 g

用于近似分位数摘要查询的数据结构。

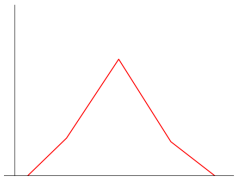
[1] 对于 $a \leq 0$（分别为 $a \geq k$），我们将 a-分位数定义为 0 分位数（分别为 k-分位数）。



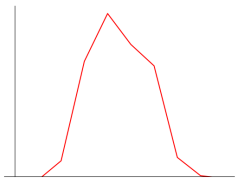
(a) 分布。



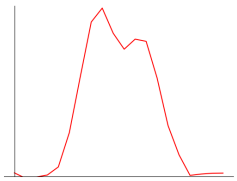
(b) 经典结果（最小值，最大值和平均值)



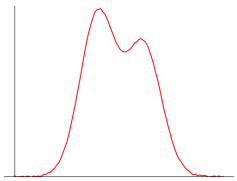
(c) 近似五分位数（$\varepsilon =$ 0。



(d) 近似十分位数 ($\varepsilon =$ 0.1)。



(e) 近似二十分位数 ($\varepsilon =$ 0.05)。



(f) 近似百分位数 ($\varepsilon = 0.01$)。

图 1 比较分布的各种近似。

重频项摘要。 考虑一个集合 P，其中包含 k 个点，集合 P 中的每个点都被赋予一个来自集合 $[n]$的颜色。设 f_i 为颜色 i 在 P中的频率，即颜色 i 在 P中的点里出现的次数。参数为 ε的重频项摘要 ***(HHS)*** 是一个列表，列出所有满足 $f_i \geq \varepsilon k$ 的颜色 i，以及对应的值 f_i。与之前一样，使用精确的 HHS 会导致数据结构效率极低，因此我们再次转向近似。参数为 ε 的近似重频项摘要 ***(AHHS)*** 是一个颜色列表 L，其中每个满足 $f_i \geq \varepsilon k$ 的颜色 i 都包含在 L 中；此外，每个颜色 $i \in L$还附有其频率的一个近似值 f_i'，使得 $f_i - \varepsilon k \leq f_i' \leq f_i + \varepsilon k$。

▶**问题 2**（近似重频项摘要）。 考虑一个输入集 P，其中包含 n 个点，且$\mathbb{R}^d$ 中的每个点都被赋予来自集合 $[n]$ 的一种颜色。给定参数 ε，我们需要构建一个数据结构，使其在给定查询 γ 时，能够返回集合 $P \cap \gamma$ 的 *AHHS*。

请注意，在这两个问题中，查询的输出大小在最坏情况下均为 $O(1/\varepsilon)$。我们的主要目标是获得具有最优最坏情况查询时间$O(\log n + \varepsilon^{-1})$的数据结构。注意，定义一个输出敏感变体是有意义的，其中查询时间为 $O(\log n + k)$，k 为输出大小。例如，对于 AHHS 查询，重频项的数量可能远少于 ε^{-1}。对于 AQS 查询而言，这样做的意义较小，因为除非查询范围 γ 内的权重分布几乎恒定，否则 AQS 将具有 $\Omega(\varepsilon^{-1})$ 个不同值。由于我们的主要关注点是 AQS，因此我们仅考虑查询时间为 $O(\log n + \varepsilon^{-1})$ 的最坏情况 AHHS 数据结构。

关于记号的说明。 为了简化查询时间和空间表达式中的繁杂形式，我们约定 $\log(\cdot)$ 函数的值至少为 1，例如，对任意正值 a, b，我们将 $\log_a b$ 定义为 $\max\{1, \ln b \ln a\}$。

1.1.1 已有结果

如前所述，经典的范围搜索解决方案主要关注较为简单的查询，这些查询可以返回给定范围内点的求和、加权和、最小值、最大值或完整列表

查询范围。这是一个经过广泛研究的领域，可供引用的结果众多，因此我们请读者参阅 Agarwal [7] 撰写的一篇优秀综述，其中涵盖了这类经典结果。

然而，经典的范围搜索数据结构无法提供有关查询区域内点集的详细统计信息，除非选择报告查询范围内的整个点子集；如果点集很大，这样做的代价可能会非常高。因此，人们尝试回答更多信息量更大的查询。例如，“范围中位数”查询受到了相当多的关注 [28, 12, 26]。请注意，中位数等同于 0.5 分位数，因此这可以视为最早尝试回答分位数查询的方法。可是，精确范围中位数查询的最优解（线性空间和对数查询时间）目前仅在一维情形下得到 [12]。对于高维情形，据我们所知，唯一已知的技术是将问题归约为若干范围计数实例 [12, 16], ，而在范围搜索领域，为精确范围计数寻找高效数据结构仍是一项重大开放问题。由于这一障碍，人们开始研究该问题的近似版本 [11] 。

数据摘要查询也受到了一定关注，尤其是在几何查询的背景下。Agarwal 等人 [8] 证明，重频项摘要（以及其他一些数据摘要）是“可合并的”，这为高维中的许多不同查询提供了一个基准解决方案，尽管直接应用他们的技术会导致对 ε 的依赖并非最优。具体而言，对于 $d = 2$ 和半空间（或单纯形）查询，该方法会得到一个线性空间的数据结构，其查询时间为 $O(\frac{\sqrt{n}}{\varepsilon})$。对于 $d = 3$，查询时间为 $O(n^{2/3}/\varepsilon)$。一般来说，在朴素实现中，查询时间为 $O(f(n)/\varepsilon)$，其中 $f(n)$ 是相应“基准”范围搜索查询的查询时间（详情见表 1）。 [25] 采用了一种更高效的摘要合并方法，他们在通信复杂度的框架下研究了这一问题。不过，似乎也可以将他们的方法与划分树的标准应用相结合，用于数据结构；具体而言，在构建最优划分树后，对于树中的任意节点 v，都将其视为通信问题中的一个参与方，并将 v 子树中的点集作为其输入。在查询时，确定覆盖查询范围的 $O(n^{2/3})$ 个子集后，目标是合并所有涉及的摘要。将 [25] 中的结果代入后，可以得到一个线性空间的数据结构，其查询时间为 $\tilde{O}(n^{2/3} + n^{1/6}\varepsilon^{-3/2})$。

构建用于范围摘要查询的最优数据结构这一问题，Wei 和 Yi 仅在一维情形下进行了研究 [36]。他们构建了一种数据结构，可用于回答多种摘要查询，包括重频项查询，并证明可以获得一种空间复杂度为 $O(n)$、查询时间为 $O(\log n + 1/\varepsilon)$ 的最优数据结构。除此之外，目前只有次优解。近期，人们开始研究“范围采样查询”，其目标是从集合 $|P \cap \gamma|$ [3, 5, 24] 中提取 k 个随机样本。事实上，研究范围采样查询的主要动机之一，是获取有关查询 [3] 内点集分布的信息。具体而言，范围采样提供了一种获取“数据摘要”的通用方法；例如，可以借此解决重频项查询问题。然而，这种方法存在一些问题，尤其是需要从集合 $|P \cap \gamma|$ 中至少采样 $1/\varepsilon^2$ 个点；即便如此，它也只能提供一种蒙特卡洛类型的近似，这意味着若要提高概率保证，还需要采样更多点。例如，要获得高概率保证，需要 $\Omega(\varepsilon^{-2}\log n)$ 个样本。

类型2颜色计数。 这些查询由 Gupta 等人于 1995 年在“彩色范围计数”领域提出。对于这个问题，给定一组带颜色的点，我们希望报告出现在给定查询范围内的所有颜色的频率。这是

一个经过充分研究的问题，但研究大多集中于正交情形，例如参见 [14]。

AHHS 查询可以视为带有加性误差的近似二类颜色计数查询。考虑一个包含 k 个点的查询。如果我们允许二类计数存在 εk 的误差，那么可以忽略频率低于 εk 的颜色；但对于其他颜色，我们必须报告其频率，且误差不超过 εk。这等价于回答一个 AHHS 查询。

其他相关问题。 Karpinski 和 Nekrich [27]研究了在给定的（正交）查询范围内查找最高频颜色的问题。这个问题后来也受到学界的进一步关注 [10, 9, 21]。但引入近似后，问题的性质会发生根本变化。

计算模型。 我们的计算模型是实数 RAM：我们可以使用实数寄存器，在常数时间内对实数执行标准运算；同时，我们还可以使用长度为 $w = \Theta(\log n)$ 位的整数寄存器，对整数执行标准运算，以及一些可通过查表实现的额外非标准运算，因为我们只需要对少于 $\frac{1}{2}\log n$ 位的数执行二元运算。请注意，我们的数据结构适用于输入坐标为实数的情况；不过，在某个阶段，我们会利用计算模型的能力来操作其整数寄存器中的位。

1.2 我们的贡献

我们的主要结果以及与已知结果的比较见表 1。

总体而言，我们通过可合并性和独立范围采样 [8, 5]，为三维 AHHS 和 AQS 查询问题取得了一系列新结果，在几乎保持相同线性空间用量的同时，将查询时间最多提升了一个巨大的乘法 $n^{\Omega(1)}$ 倍数。这项改进颇具挑战性，需要创新地结合浅切割引理、划分定理、ε-近似等已知技术，以及一些新思路，例如针对非正交查询的位打包、利用 AHHS 实例求解 AQS 查询问题、保秩几何采样等。

对于支配查询，我们取得了首个最优结果。当 $\varepsilon^{-1} = O(\log n)$ 时，我们的半空间 AHHS 结果也达到了最优。请注意，对于较小的 ε 值，我们的半空间 AHHS 结果显著缩短了查询时间，优于以往方法。此外，我们还为二维情形下的上述问题给出了改进结果，并略微改进了精确二类单纯形颜色计数结果。

2 预备知识

在本节中，我们将介绍在研究结果中使用的主要工具。有关所用工具的全面介绍，请参见附录 A 和附录 B。

2.1 浅切割与近似范围计数

给定 $\mathbb{R}^3$ 中的 n 个超平面组成的集合 H，点 $q \in \mathbb{R}^3$ 的层级是 H 中位于 q 下方的超平面数量。我们将所有层级至多为 k 的点的轨迹称为 $(\leq k)$-层级，其轨迹的边界称为 k-层级。针对 H 的 $(\leq k)$-层级的浅切割 $\mathscr{C}$（简称 k-浅切割）是一组不相交单元（四面体），它们共同覆盖 H 的 $(\leq k)$-层级，并且具有以下性质：切割中的每个单元 $C \in \mathscr{C}$ 都与 H 中的一组 H_C 超平面相交，这组超平面称为 C 的冲突列表，其大小为 k 的 O。浅切割引理如下。

XX:6 关于范围汇总查询

表 1 我们的主要结果与基于可合并性的 [8] 方案和基于独立范围采样（IRS）的 [5] 方案的比较。基于 IRS 的方案是随机化的，在参数 $0 < \delta < 1$ 下，成功概率为 $1 - \delta$。 F 是输入中的颜色数。 $w = \Theta(\log n)$ 是机器的字长。† 表示最优方案。

摘要 查询类型	空间	查询时间	备注
类型2 **单纯形颜色计数**	$O(n)$	$O\left(n^{1-\frac{1}{d}} + \frac{n^{1-\frac{1}{d}} F^{\frac{1}{d}}}{w^{\alpha}}\right)$	New
三维 AHHS **半空间**	$O(n)$ $O(n)$ $O(n)$ $\boldsymbol{O(n \log_w \frac{1}{\varepsilon})}$	$O(\log n + \frac{1}{\varepsilon} n^{2/3})$ $\tilde{O}(n^{2/3} + \frac{1}{\varepsilon_{3/2}} n^{1/6})$ $O(\log n + \frac{1}{\varepsilon^2} \log^{-\frac{1}{\delta}})$ $\boldsymbol{O(\log n + \frac{1}{\varepsilon})}$	基于可合并性 [8] 蒙特卡洛 [25] 基于IRS [5] **New**
三维 AHHS **支配关系**	$O(n)$ $O(n)$ $\boldsymbol{O(n)}$	$O(\log n + \frac{1}{\varepsilon} \log^3 n)$ $O(\log n + \frac{1}{\varepsilon^2} \log^{-\frac{1}{\delta}})$ $\boldsymbol{O(\log n + \frac{1}{\varepsilon})}$	基于可合并性的 [8] 基于IRS [5] **New†**
三维AQS **半空间**	$O(n)$ $O(n)$ $\boldsymbol{O(n \log^2 \frac{1}{\varepsilon} \log_w \frac{1}{\varepsilon})}$	$O(\log n + \frac{1}{\varepsilon} n^{2/3} \log(\varepsilon n))$ $O(\log n + \frac{1}{\varepsilon^2} \log^{-\frac{1}{\delta}})$ $\boldsymbol{O(\log n + \frac{1}{\varepsilon} \log^2 \frac{1}{\varepsilon})}$	基于可合并性的 [8] 基于IRS [5] **New**
三维AQS **支配关系**	$O(n)$ $O(n)$ $\boldsymbol{O(n)}$	$O(\log n + \frac{1}{\varepsilon} \log^3 n \log(\varepsilon n))$ $O(\log n + \frac{1}{\varepsilon^2} \log^{-\frac{1}{\delta}})$ $\boldsymbol{O(\log n + \frac{1}{\varepsilon})}$	基于可合并性 [8] 基于IRS [5] **New†**

▶**引理 1**。对于 $\mathbb{R}^3$ 中的任意 n 个超平面以及参数 k，存在一个大小为 $O(n/k)$ 的 $O(k/n)$-浅切割，它覆盖 $\leq k(\leq k)$-层级。切割中的单元都是从下方无界的竖直棱柱（其顶点位于 $(0,0,-\infty)$）。

此外，对于任意 $a > 1$，我们可以在 $O(n \log n)$ 时间内，同时构造所有形如 a^i 的 k。给定任意点 $q \in \mathbb{R}^3$，我们可以在 $O(\log n)$ 时间内找到位于 q 上方的最小层级 k，以及包含 q 的单元。

上述内容同样适用于支配范围，其定义如下。给定 $\mathbb{R}^d$ 中的两个点 p 和 q，当且仅当 p 的每个坐标都大于或等于 q 的相应坐标时，p 支配 q。被 p 支配的 $\mathbb{R}^d$ 的子集称为 支配范围。当范围搜索问题中的查询范围是支配范围时，我们称其为 支配查询。正如 Chan 等人 [15]，所观察到的，支配查询可以通过半空间查询来模拟，因此引理 1 适用于支配查询。详细信息请参见附录 B。

我们利用浅切割得到范围计数结果的近似版本。

▶ **定理 2**（近似范围计数 [2]）。设 P 是 $\mathbb{R}^3$ 中的 n 个点的集合。对于半空间或支配范围，可以构建一个大小为 $O(n)$ 的数据结构，使得对于给定的查询范围 γ，可以在 O 时间内报告 $|\gamma \cap P|$，误差为 $\alpha|\gamma \cap P|$，其中 $\alpha > 0$ 为任意常数。

2.2 ε-近似

我们还将使用另一种工具：ε-近似，这是一种有用的抽样技术：

▶ **定义 3。** 设 (P,Γ) 是一个有限集合系统。给定任意 $0<\varepsilon<1$，集合 $A\subseteq P$ 称为 (P,Γ) 的 ε-近似，当且仅当对任意 $\gamma\in\Gamma$，$\left|\frac{|\gamma\cap A|}{|A|}-\frac{|\gamma\cap P|}{|P|}\right|\leq\varepsilon$.

上述集合 A 使我们能够通过精确计算 $|\gamma\cap A|$，以加性误差 $\varepsilon|P|$ 近似得到 $\gamma\cap P$ 中的点数；本质上，ε-近似将大集合 P 上的近似计数问题转化为小集合 A 上的精确计数问题。

已有研究表明，由点和半空间/支配范围构成的集合系统存在较小规模的 ε-近似：

▶ **定理 4** (ε-近似 [31, 32])。 对于半空间和支配范围，分别存在大小为 $\varepsilon(\varepsilon^{-\frac{2d}{d+1}})$ 和$O(\varepsilon^{-1}$对数$^{d+1/2}\,\varepsilon^{-1})$ 的近似。

3 近似重频项摘要查询

我们利用近似重频项摘要（AHHS）查询的改进结果来解决近似分位数摘要（AQS）查询。本节概述我们新提出的 AHHS 解决方案的主要思路，并请读者参阅附录 D 以了解详细内容。为使描述更加清晰，我们使用 ε_0 表示 AHHS 查询的目标误差，并将 ε 保留为通用误差参数。我们证明了以下结论。

▶ **定理 5. </style}对于，近似半空间重频项摘要查询可以使用 (**$n\log_w(1/\varepsilon_0)$) 空间回答，并且查询时间达到最优的 $O(\log n+1/\varepsilon_0)$。

▶ **定理 6. </style}对于，近似半空间重频项摘要查询可以使用 (**$n\log\log_w(1/\varepsilon_0)$) 空间回答，并且查询时间达到最优的 $O(\log n+1/\varepsilon_0)$。

▶ **定理 7. </style}对于 ,3**，近似支配重频项摘要查询可以使用最优的 $O(n)$ 空间回答，并且查询时间达到最优的 $O(\log n+1/\varepsilon_0)$。

3.1 基本解法

上述结果基于一种基本解法，该解法解决以下问题：

▶**问题 3.** *[粗粒度 AHHS 查询]* 设 P 是 $\mathbb{R}^d$中的一组点，每个点都关联一个颜色。问题是将 P 存储在某种结构中，使得对于给定的查询范围 q，能够高效地估计 $q\cap P$ 中各颜色的频率，误差不超过加性误差 $\varepsilon|P|$，其中参数 $0<\varepsilon<1$取某个值。

注意，这里我们允许更大的误差（因为误差是针对整个点集定义的）。要解决问题 3，我们需要的一个关键组件是适用于半空间的更优（精确）类型2颜色计数结构。我们以一种新颖的方式结合了几种已知技术与位打包，得到以下定理。详情见附录 C。

▶ **定理 8.** 给定一个整数参数 F，一个由 n 个 $\mathbb{R}^d$ 中的点组成的集合 P，其中每个点都被赋予集合 $[F]$中的一种颜色，可以构建一个线性规模的数据结构，使得给定一个查询单纯形 q，它能在总时间 $\max\{O(n^{(d-1)/d})\ O(n^{(d-1)/d}F^{1/d}/w^{\alpha})\}$内输出 $P\cap q$ 中每种颜色出现的次数，其中某个适当的常数为 α，字长为 w。

求得一个基础解的主要思路相对简单。我们根据颜色的频率对颜色进行分组，使每组中的颜色频率大致相同。不过，我们必须谨慎执行，而分析也有些棘手。对于

例如，如果将所有点都放入定理 8 的一个数据结构副本中，就会得到次优结果。不过，只要正确地对点进行分组，并严格进行分析，就可以得到以下结论。

▶ **定理 9. </style}对于，单纯形查询（个半空间的交集）中的问题 3 可使用 (**X) </style} 空间解决，其中 $\min\{|P|, \varepsilon^{-\frac{2d}{d+1}}\}$，查询时间为

$$O\left(\frac{|P|^{1-\frac{2}{d-1}}}{w^{\alpha}\varepsilon^{\frac{2}{d-1}}}\right)+O\left(X^{\frac{d-1}{d}}\right)$$

其中 w 是机器的字长，α 是某个正数常量。

主要的挑战在于，ε 点上 n 近似的大小有两种情况，因为它受 $\min\left\{n, O(\varepsilon^{-\frac{2d}{d+1}})\right\}$ 的界定；此外，定理 8 的查询时间也有两种情况。不过，核心思路是，由于总误差预算为 $\varepsilon|P|$，我们可以选取更大的误差参数 $\varepsilon_i = \frac{\varepsilon|P|}{|P_i|}$，其中 P_i 是颜色为 i 的点集。详细内容见附录 D。

3.2 求解 AHHS 查询

我们首先将问题转换到对偶空间中。因此，点集 P 变为一个由超平面组成的集合 H，而任意查询半空间则变为一个点 q。我们希望找到 H 下方超平面的近似重频项。在这里，我们指出，构建一个空间复杂度为 $O(n\log\frac{1}{\varepsilon_0})$ 的数据结构并不太困难：为 H的排列构建一个浅切割层次结构，覆盖 $i=0,1,\cdots,\log(\varepsilon_0 n)$ 层级 $2^i/\varepsilon_0$。对于每个浅切割单元 Δ，我们针对参数 $\varepsilon=\varepsilon_0/c$，为冲突列表 $\mathcal{S}_\Delta$ 构建前面介绍的基础数据结构，其中 c 是一个足够大的常数。然后注意，对于低于层级 ε_0^{-1} 的查询，我们可以花费 $O(\log n+\frac{1}{\varepsilon_0})$ 的时间，找到所有经过该查询下方的超平面，并显式回答 AHHS 查询。此外，对于高于层级 $\varepsilon_0^{-3/2}$ 的浅切割层级，基础解决方案所使用的总空间为 $O(n)$。因此，事实证明，主要困难在于处理介于 ε_0^{-1} 和 $\varepsilon_0^{-3/2}$ 之间的层级。

为了将空间缩减到 $O(\log_{w}\frac{1}{\varepsilon_0})$，回想一下，在基础数据结构的查询时间中，我们有两项：$O(1/(\varepsilon_0 w^{\alpha}))$ 和 $O(X^{2/3})$。注意，我们可以将 ε 设置为大约 ε_0/w^{α}，而第一项仍然是 $O(\varepsilon_0^{-1})$；因为我们处于低于 $\varepsilon_0^{-3/2}$ 的层级，所以有 $X<\varepsilon_0^{-3/2}$，因此第二项始终是 $O(\varepsilon_0^{-1})$！设置 $\varepsilon=\varepsilon_0/w^{\alpha}$ 的效果是，现在我们为一个单元构建的基础数据结构可以输出频率，精度提高 w^{α} 倍，这意味着它可用于多出 w^{α} 倍的层级。因此，我们只需要为基于浅切割构建的结构构建基础数据结构，且浅切割的因子为 w^{α}！这样便得到 $O(n\log_{w}\frac{1}{\varepsilon_0})$ 的空间上界。当然，此处的输出大小为 $O(\varepsilon^{-1})=O(w^{\alpha}\varepsilon_0^{-1})$，我们无法负担检查所有这些颜色。这里的最后一个关键点是，我们可以利用因子为 2 的浅切割，维护一个包含 $O(\varepsilon_0^{-1})$ 个候选颜色的列表。

我们备注，尽管这些工具都是标准的，但将这些工具结合起来并进行分析颇具难度。此外，当我们有 $\Theta(1/\varepsilon_0)$ 个重频项时，我们的查询时间是最优的。查询时间能否做到与输出敏感，是一个有趣的开放问题。

4 近似分位数摘要查询

在本节中，我们解决问题 1。首先，我们展示一种通用技术：利用 AHHS 查询的解，高效地解决近似分位数摘要查询。我们证明，

对于三维半空间和支配范围，我们可以将 AHHS 查询的解转换为近似分位数摘要查询的解，空间和时间开销增加 $O(\log^2 \frac{1}{\varepsilon})$。随后在第 4.2 节中，我们基于另一种思路给出支配范围的最优解。

首先，我们展示如何使用 AHHS 查询解法来处理近似分位数摘要查询。我们介绍针对半空间的数据结构，因为正如前文所述，同样的方法也适用于三维中的支配范围。我们结构的整体思路如下：首先将问题转换到对偶空间。由此得到的问题实例中有n 个带权超平面，给定查询点 q，我们希望提取穿过其下方的超平面的近似分位数摘要。为此，我们构建层次化浅切割。对于每个切割中的每个单元，我们收集其冲突列表中的超平面，然后按照权重递增的顺序将它们划分为 $O(\frac{1}{\varepsilon_0})$ 组。给定对偶空间中的一个查询点，我们首先找到包含该点的切割和单元，然后在查询点下方的子集中，确定每组的近似排名。为此，我们生成一个 AHHS 问题实例并应用定理 5。我们构造该实例，使得近似排名的误差足够小，从而可以遍历各组，并从相应组中任意选取一个超平面，组成近似 ε_0 分位数摘要。

4.1 数据结构与查询算法

我们对这组 P 个输入点 n 进行对偶变换，得到一组 $H = P$ 个超平面 n。然后，我们构建一个浅切割层次结构，其中第 i 个浅切割 $\mathscr{C}_i$ 是一个 k_i-浅切割，满足 $k_i = \frac{2^i}{\varepsilon_0}$，其中 $i = 0$, 1，2，$\cdots$，$\log(\varepsilon_0 n)$。考虑第 i 个浅切割中的一个单元 Δ 及其冲突列表 $\mathcal{S}_\Delta$。令 $\epsilon = \frac{\varepsilon_0}{c}$，其中常数 c 足够大。我们将 $\mathcal{S}_\Delta$ 划分为 $t = \frac{1}{\epsilon}$ 组 G_1，G_2，$\cdots$ G_t，并按权重排序，也就是说，对于 $j = 1$, 2，$\cdots$ $t-1$，G_j 中任意超平面的权重都不大于 G_{j+1} 中任意超平面的权重。

对于每个组 G_j，我们将其中超平面的最小权重作为该组的代表。为简化描述，我们作出简化假设，认为 t 是 2 的幂（否则，可以添加一些虚拟组）。我们将各组 G_j 作为平衡二叉树 $\mathcal{T}$ 的叶节点，并令 $V(\mathcal{T})$ 表示 $\mathcal{T}$ 的顶点集合。接下来，我们构建以下与 Δ 相关联的彩色超平面集合 A_Δ：令 $\varepsilon' = \epsilon \log^2 t$。对于每个顶点 $v \in V(\mathcal{T})$，令 G_v 为包含在 v 的子树中的所有超平面构成的集合；我们将 G_v 的一个 ε'-近似 E_v，以颜色 v 加入 A_Δ。利用定理 5，我们将与 A_Δ 中超平面对应的对偶点存储在数据结构 Ψ_Δ 中，以支持误差参数为 ε' 的 AHHS 查询。至此，我们的数据结构描述完毕。

查询算法。给定的查询 q 按如下方式回答。我们快速回顾一下标准步骤：我们在对偶空间中考虑该查询，因此 q 被视为一个点。令 k 为经过 q下方的超平面数。注意，根据定理 2，对于任意常数 α，我们可以使用消耗线性空间的数据结构，在 $O(\log n)$ 时间内，找到 k 的 $(1+\alpha)$ 因子近似 k^*。这使我们能够找到首个 $k_{i-1} < k \le k_i$为 $\mathscr{C}_i$ 的 k_i-浅切割。使用标准的点定位数据结构（例如，参见 [1]），也可以在 $O(\log n)$ 时间内找到包含 q 的单元 $\Delta \in \mathscr{C}_i$。

查询中有趣的部分，是找到单元 Δ后如何处理该查询。令H_q 为 H 中位于 q下方的子集。回想一下，$\mathcal{S}_\Delta$ 是 H 中与 Δ相交的子集。 Δ 的重要性质是， $H_q \subset \mathcal{S}_\Delta$，并且 $|\mathcal{S}_\Delta| = O(|H_q|) = O(\{v21)$。

我们查询为 Δ 构建的数据结构 Ψ_Δ，以获得颜色列表及其近似计数，其中近似的加性误差至多为 $\varepsilon'|A_\Delta|$。为了继续描述查询算法，我们用记号 g_j 表示 G_j 中位于 q 下方的子集，并令 $g = \cup_{j=1}^{t} g_j$，于是 $|g| = k$。

XX:10 关于范围汇总查询

请注意，虽然查询算法无法直接访问 g 或 k，但我们声称，利用数据结构 Ψ_Δ 的输出，可以计算 g_i 中各元素在 g 中的近似秩，且加性误差至多为 $\varepsilon_0 k$。同样，我们可以用树 $\mathcal{T}$ 来直观展示这一过程。回想一下，在 Ψ_Δ 中，每个顶点 $v \in V(\mathcal{T})$ 都表示数据结构 Ψ_Δ 中的一种独特颜色，而该数据结构会返回一个误差参数为 ε' 的 AHHS 摘要。这使我们能够估计 E_v 中经过 q 下方的元素数量，误差为 $\varepsilon'|A_\Delta|$；又由于 E_v 是 G_v 的 ε'-近似值，因此我们能够估计 G_v 中经过 q 下方的元素数量，误差至多为 $2\varepsilon'|A_\Delta|$。考虑表示 $g_j \subset G_j$ 的叶节点，以及将它连接到 $\mathcal{T}$ 根节点的路径 π。g_j 的近似秩 r_j 按如下方式计算。考虑一个根节点为 u、悬挂在路径 π 左侧的子树（如图 2 所示）。如果颜色 u 未出现在 AHHS 查询的输出中，那么我们可以断定，其超平面中至多有 $2\varepsilon'|A_\Delta|$ 个经过 q 下方；此时我们不作任何处理。如果该颜色出现在 AHHS 查询的输出中，那么我们就知道其子树中经过 q 下方的超平面数量，误差为 $2\varepsilon'|A_\Delta|$；此时，我们将这一估计值加到 r_j 上。在这两种情况下，我们的结果都存在 $2\varepsilon'|A_\Delta|$ 的加性误差。我们对悬挂在 π 左侧的每个子树重复这一操作。此类子树的数量至多为 $\log t$，因此总误差至多为 $2\varepsilon'|A_\Delta|\log t$。现在注意到

$$2\varepsilon'|A_\Delta|\log t = 2\frac{\epsilon}{\log^2 t}\cdot \log t|\mathcal{S}_\Delta|\cdot\log t = O(\varepsilon k) = O\left(\frac{\varepsilon_0 k}{c}\right) \le \varepsilon_0 k$$

只需将 c 设置得足够大，并注意到 $|A_\Delta| \le \log t|\mathcal{S}_\Delta|$，因为 $\mathcal{S}_\Delta$ 中的每个超平面都重复了 $\log t$ 次。

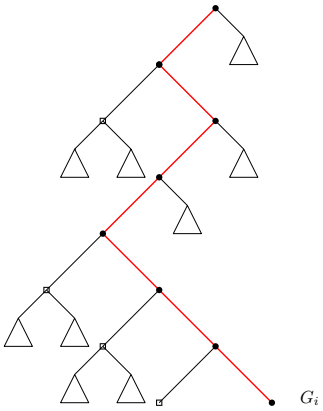


图 2 计算一组元素的近似秩：G_i 的近似秩等于所有方形节点的近似计数之和。

现在我们几乎完成了。我们刚刚证明，在每个 g_i中，我们都知道其元素在 g 中的秩，且误差不超过 $\varepsilon_0 k$的加性误差。这意味着，从每个 G_i中选取一个元素，就能得到一个近似分位数总结的超集；在查询算法的最后阶段，我们只需按如下方式剪除不必要的元素：从左到右扫描$\mathcal{T}$中的所有叶子，即依次考虑从 $j=1$ 到 t 的组 G_j，并以直接的方式计算分位数摘要。具体来说，我们初始化一个变量 $j'=0$，然后依次考虑 G_j，其中 $j=1$ 从 t开始。当 r_j 首次超过一个分位数边界，即 $r_j \ge j'\varepsilon_0 k$*时，我们将 G_j 中权重最低的超平面加入近似的 ε_0分位数摘要，然后将 j'递增。

4.1.1 分析

根据上一段，正确性已经得到证明。因此，接下来只需分析空间复杂度和查询复杂度。我们先分析前者。

空间使用量。 考虑为单元 Δ 从 k_i-浅切割 $\mathscr{C}_i$ 构建的结构 Ψ_Δ。注意，由于在求和中每个超平面都会被计算 $\log t$ 次，因此 $\sum_{v\in V(\mathcal{T})} |G_v| = |\mathcal{S}_\Delta| \log t$。$E_v$ 是 G_v 的 ε'-近似，因此

$$|E_v| \leq \min\left\{\varepsilon'^{-3/2}, G_v\right\} \tag{1}$$

这意味着

$$|A_\Delta| = \sum_{v\in V(\mathcal{T})} |E_v| \leq \min\left\{\varepsilon'^{-3/2} 2t, |\mathcal{S}_\Delta| \log t\right\} \tag{2}$$

其中，第一部分成立是因为 $\mathcal{T}$ 中至多有 $2t$ 个顶点，第二部分则由 (1) 得出。我们在集合 A_Δ 上构造定理 5 的一个实例；根据定理 5，其空间使用量为 $O(|A_\Delta| \log_w \frac{1}{\varepsilon'})$。假设 Δ 属于一个 k_i-浅切割 $\mathscr{C}_i$，则有 $|\mathcal{S}_\Delta| = O(k_i)$，并且 $\mathscr{C}_i$ 中有 $O(n/k_i)$ 个单元。注意

$$\sum_{\Delta\in\mathscr{C}_i} |A_\Delta| = \sum_{\Delta\in\mathscr{C}_i} \min\left\{\varepsilon'^{-3/2} 2t, |\mathcal{S}_\Delta| \log t\right\} = \sum_{\Delta\in\mathscr{C}_i} O\left(\min\left\{\varepsilon'^{-3/2} t, k_i \log t\right\}\right) = O\left(\min\left\{\frac{n}{k_i}\varepsilon_0^{-3}, n\log\frac{1}{\varepsilon_0}\right\}\right). \tag{3}$$

因此，$\mathscr{C}_i$ 所使用的总空间为

$$O\left(\min\left\{\frac{n}{k_i}\varepsilon_0^{-3}\log_w\frac{1}{\varepsilon_0}, n\log_w\frac{1}{\varepsilon_0}\log\frac{1}{\varepsilon_0}\right\}\right).$$

最后注意，第二项占主导的层级最多有 $O(\log \frac{1}{\varepsilon_0})$ 个；具体而言，至少当 k_i 超过 ε_0^{-4} 时，第一项占主导，而这些层级所使用的总空间为 $O(n)$，因为 k_i'构成一个几何级数。因此，我们的数据结构的总空间使用量为 $O(n\log^2\frac{1}{\varepsilon_0}\log_w\frac{1}{\varepsilon_0})$。

查询时间。 根据引理 1，我们可以在 $O(\log n)$ 时间内找到所需的切割单元。接下来，我们查询数据结构 Ψ_Δ，根据定理 5，它使用 $O(\log n + \varepsilon'^{-1}) = O(\log n + \epsilon\log^2 t) = O(\log n + \frac{1}{\varepsilon_0}\log^2\frac{1}{\varepsilon_0})$ 查询时间。扫描各组并对数据结构 Ψ_Δ 的输出进行剪枝所需的时间渐近地更短，因此可将其吸收到上述表达式中。因此，我们得到以下结果。

▶**定理 10。** 给定一个包含误差参数 ε_0 和 $\mathbb{R}^3$ 中 n 个点的集合 P 的输入，其中每个点 $p\in P$ 都关联有一个来自全序全集的权重 w_p，可以构建一个占用 $O(n\log^2\frac{1}{\varepsilon_0}\log_w\frac{1}{\varepsilon_0})$ 空间的数据结构，使其能够针对任意查询半空间 h，在 $O(\log n + \frac{1}{\varepsilon_0}\log^2\frac{1}{\varepsilon_0})$ 时间内回答参数为 ε_0 的 *AQS* 查询。

对于二维的情况，只需将 Ψ_Δ 替换为定理 6 中的结构，便可立即得到以下结果：

▶**定理 11。** 给定一个由误差参数 ε_0 和一个包含 n 个点的集合 P 构成的输入，这些点位于 $\mathbb{R}^2$ 中，其中每个点 $p\in P$ 都关联一个来自全序全集的权重 w_p，可以构建一个占用 $O(n\log^2\frac{1}{\varepsilon_0}\log\log_w\frac{1}{\varepsilon_0})$ 空间的数据结构，使其能够针对任意给定的查询半空间 h，以 $O(\log n + \frac{1}{\varepsilon_0}\log^2\frac{1}{\varepsilon_0})$ 时间，回答参数为 ε_0 的 *AQS* 查询。

4.2 支配近似分位数总结查询

现在我们将注意力转向支配范围。我们将展示一种类似于半空间查询的结构。主要区别在于，我们现在使用精确的二型颜色计数作为辅助结构，以估计每个组的秩。这使我们在空间和查询时间方面都大约节省了 $\log^2 \frac{1}{\varepsilon_0}$ 个因子，因此我们可以在最优的$O(\log\ n + \frac{1}{\varepsilon_0})$ 时间内回答分位数查询。为了将空间降至线性，我们还需要更多想法。我们首先介绍一种次优但更简单的结构，以展示我们的主要思路。然后，我们修改该结构以获得所需的最优结构。我们在原始空间中使用浅切割。

4.2.1 一种次优的 $O(n \log\log \frac{1}{\varepsilon_0})$ 空间解决方案

我们首先介绍一种数据结构，它以 $O(n\log\log \frac{1}{\varepsilon_0})$ 空间解决支配近似分位数总结问题，并达到最优的 $O(\log\ n + \frac{1}{\varepsilon_0})$ 查询时间。

加权点的秩保持近似。 设 S 是一个加权点集，其中每个点都被赋予一个来自全序全集的权重。设 $r_S(p)$ 为集合 S 中点 p 的秩。考虑一个几何集合系统 $(P, \mathscr{D})$，其中 P 是 $\mathbb{R}^3$ 中的加权点集，而 $\mathscr{D}$ 是由三维支配范围诱导的 P 的子集族。我们介绍一种为 P 构造样本 A 和参数 ϵ 的方法，使得

$$\left|\frac{r_{P\cap\mathcal{D}}(p)}{|P|} - \frac{r_{A\cap\mathcal{D}}(p)}{|A|}\right| \le \epsilon \tag{4}$$

对于任意点 $p \in P$和任意范围 $\mathcal{D} \in \mathscr{D}$。首先注意，对 P取 ϵ-近似**并不**可行，因为这没有考虑 P 的权重。我们的一个简单但重要的观察是，可以将点 P 提升到四维空间，并将其对应的权重作为第四个坐标。我们将这个新的点集称为 P'，并令 $(P', \mathscr{D}')$ 表示由四维支配范围在四维空间中诱导的集合系统。考虑 P' 的一个 ϵ-近似 A'，并令 A 表示将 A' 投影到前三维后得到的集合（即再次移除权重）。 A 将作为 P 的样本，为了将其与无权近似区分开来，我们称之为保秩 ϵ-近似。实际上，对于任意权重为 w_p的点 $p \in P$ 以及任意 $\mathcal{D} \in \mathscr{D}$， $r_{P\cap\mathcal{D}}(p)$（分别为 $r_{A\cap\mathcal{D}}(p)$）等于 P'（分别为 A'）中包含在四维支配范围 $\mathcal{D} \times (-\infty, w_p)$ 内的点数。根据 ϵ-近似的定义，性质 (4) 成立。

现在我们将注意力转向用于三维支配查询的近似分位数总结。

数据结构与查询算法。 与我们为半空间查询提出的结构类似，我们为 $i = 0$、1、$\cdots$、$\log(\varepsilon_0 n)$ 构建 $\frac{2^i}{\varepsilon_0}$-浅切割。设 $\kappa = O(1)$ 为一个常数，使得 $O(\frac{1}{\varepsilon_0}\log^\kappa \frac{1}{\varepsilon_0})$ 是四维支配范围近似的规模。考虑一个 k-浅切割 $\mathscr{C}$。我们分两种情况讨论：

- 如果 $k \le \frac{1}{\varepsilon_0}\log^\kappa \frac{1}{\varepsilon_0}$，对于切割 $\mathscr{C}$ 中的每个单元 Δ，我们收集其冲突列表 $\mathcal{S}_\Delta$ 中的点，并根据权重将它们分成 $t = \frac{1}{\epsilon}$ 组 G_1、G_2、$\cdots$ G_t（也就是说， G_i 中的权重不大于组 G_{i+1} 中的权重），其中 $\epsilon = \frac{\varepsilon_0}{c}$ 取足够大的常数 c，方法与处理半空间查询时相同。
- 对于 $k > \frac{1}{\varepsilon_0}\log^\kappa \frac{1}{\varepsilon_0}$，我们首先对 $\mathcal{S}_\Delta$ 取一个保持秩的 ϵ-近似，然后像上述情况一样将该近似分成 $t = \frac{1}{\epsilon}$ 组。同样，对于每一组，我们存储其中各点的最小权重。

我们为每个单元 Δ 构建以下结构。

令 N 表示我们为一个单元 Δ生成的所有 t 组中的点数。对于每个 $i = 0, 1, \cdots$ $t/\alpha - 1$，其中$\alpha = (\log\log \frac{1}{\varepsilon_0})^3$，我们将组 $G_{i\cdot\alpha+1}, G_{i\cdot\alpha+2}, \cdots$ $G_{(i+1)\cdot\alpha}$ 收集到一个簇 $\mathcal{C}_i$ 中。对于簇$\mathcal{C}_i$ 中 $j = 1, 2, \cdots$ α对应的每个组，我们将其中的点着为

该组的颜色 j。然后，我们为$\mathcal{C}_i$构建以下类型2颜色计数结构 Ψ_i。令 N_i 表示$\mathcal{C}_i$中的点总数：

- 首先，我们为每个坐标各存储一个前驱搜索数据结构，共三个。这使我们能够将输入坐标和查询坐标映射到秩空间。接下来，我们构建一个大小为 $\sqrt[3]{N_i} \times \sqrt[3]{N_i} \times \sqrt[3]{N_i}$ 的网格，使得每个切片包含 $\sqrt[3]{N_i^2}$ 个点。对于每个网格点，我们使用紧凑表示，将其支配的点存储在一个频率向量中。最后，我们对每个网格层递归（即沿每个维度各递归一次）。
- 当子问题中的点数小于某个足够小的常数 η 的 $N_* = N_i^{\eta}$ 时，递归停止。对于这些“叶节点”子问题，注意，查询的不同答案总数至多为 $O(N_i^{3\eta})$。我们构建一个查找表，记录这些答案对应的频率向量。注意，由于每一步都会进行秩空间缩减，将查询坐标缩减到秩空间后，只需 $O(1)$ 时间即可完成查找。

查询算法。给定查询 q，我们首先定位包含 q C 的网格单元，这将得到三个秩。利用 x 和 y 的秩，我们获得一个条目；再利用 z 的秩，找到存储在 C 下方角落中的对应单词和对应频率向量。通过递归处理三个子问题，我们再获得三个频率向量。将这三个频率向量合并，即可生成最终答案。至此，我们完成了对为每个族 $\mathcal{C}_i$ 构建的数据结构的描述。

要回答查询 q，我们首先找到位于 q 上方的第一个浅切割层级及其对应的切割单元 Δ。然后，我们查询上述数据结构，获取在 t 个组中分别被 q 支配的点数。接着，通过维护一个运行计数器，从左到右扫描这 t 个组，以构造近似的 ε_0-分位数摘要。

空间使用量。对于空间使用量，注意每个递归层级中有 N_i 个网格点，递归深度为 $O(1)$。共有 α 种颜色，且一种颜色的频率不超过 N_i。因此，存储频率向量所需的单词总数为$O(N_i{}^{\alpha}\log_w{}^{N_i})$。当问题规模小于 N_i^{η} 时，对于每个子问题，我们使用 $O(N_i^{3\eta\alpha}\log_w{}^{N_i})$ 个单词存储一个查找表。因此，底层所使用的单词总数为 $O(\binom{N_i}{N_i^{\eta}}) \cdot O(N_i^{3\eta\alpha}\log_w{}^{N_i}) = O(N_i{}^{N_i^{2\eta}\alpha}\log_w{}^{N_i})$。注意，根据我们的构造以及 $\varepsilon_0 \geq \frac{1}{n}$，$N = O(\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0})$，$\alpha = (\log\log\frac{1}{\varepsilon_0})^3 \leq (\log\log n)^3$ 且 $N_i = O(\alpha_{1/\varepsilon_0}^{N}) = O(\alpha\log^{\kappa}\frac{1}{\varepsilon_0}) = O(\alpha\log^{\kappa} n)$。由于根据假设，$w = \Omega(\log n)$，通过选取 η 为 $N_* = N_i^{\eta}$ 中足够小的常数，频率向量的空间使用量满足

$$f(N_i) = \begin{cases} 3\sqrt[3]{N_i} f(\sqrt[3]{N_i^2}) + O(\frac{N_i}{w^{1-o(1)}}), \text{for } N_i \geq N_* \\ O(\frac{N_i}{w^{1-\beta}}), \text{otherwise} \end{cases},$$

对于某个常数 $0 < \beta < 1$，其解为 $O(N_i{}^{(\log_{w^{1-\beta}} N_i{}^{3})}) = O(w^{N_i^{1-\tau}})$，其中 $0 < \tau < 1$ 为某个常数。由于递归深度为 $O(1)$，所有前驱搜索结构的空间使用量为 $O(N_i)$。因此，Ψ_i 的空间使用量为 $O(N)$。所以，每个浅切割单元 Δ 的总空间使用量上界为 $\frac{N}{N_i} \cdot O(N_i) = O(N)$。

对于 $k_i \geq \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$，$N = O(\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0})$。因此，它们的总空间使用量上界为

$$\sum_{i=\kappa\log\log\frac{1}{\varepsilon_0}}^{\varepsilon_0 n} O\left(\frac{n}{k_i}\right) \cdot O(N) = \sum_{i=\kappa\log\log\frac{1}{\varepsilon_0}}^{\varepsilon_0 n} O\left(\frac{n\varepsilon_0}{2^i}\right) \cdot O\left(\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}\right) = O(n).$$

对于 $k_i < \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}, N = k_i$，因此我们得到空间上界

$$\sum_{i=0}^{\kappa\log\log\frac{1}{\varepsilon_0}} O\left(\frac{n}{k_i}\right)\cdot O(N) = \sum_{i=0}^{\kappa\log\log\frac{1}{\varepsilon_0}} O\left(\frac{n}{k_i}\right)\cdot O(k_i) = O\left(n\log\log\frac{1}{\varepsilon_0}\right).$$

至此，我们完成了空间上界的证明。

查询时间。对于查询时间，我们首先花费 $O(\log n)$ 时间，通过浅切割的性质找到合适的浅切割层级及对应的单元。然后我们查询 Ψ_i 以估算单元中每个组的计数，依次为 $i = 0, 1, \cdots t/\alpha - 1$。对于每个 Ψ_i，注意每次前驱搜索都需要 $O(\log N_i)$ 时间。此外，每个频率向量都能装入一个字，因此合并两个频率向量所需时间为 $O(1)$。由此得到查询时间的以下递推关系。

$$g(N_i) = \begin{cases} 3g(\sqrt[3]{N_i^2}) + O(\log N_i), \text{for } N_i \geq N_* \\ O(\log N_i), \text{otherwise} \end{cases},$$

解得 $O((\log N_i)^3) = O((\log\log\frac{1}{\varepsilon_0})^3) = O(\alpha)$。由于我们需要查询 t/α 个这样的数据结构，以获取所有组的计数，因此计数估计的总查询时间为$O(t) = O(1/\varepsilon_0)$。然后我们遍历各组并报告近似分位数，这又需要 $O(1/\varepsilon_0)$ 时间。因此，总查询时间为 $O(\log n + \frac{1}{\varepsilon_0})$。

正确性。给定查询 q，设 k 为被 q支配的点的实际数量。根据浅切割的性质，我们可以在其上方的浅切割层级 k_i中找到一个包含 q 的单元 Δ，使得 $k \leq k_i \leq 2k$。当 $k_i < \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$时，在估算每个组中的计数后，由于估算是精确的，且每组的大小为 $|\mathcal{S}_\Delta| = \varepsilon_0|\mathcal{S}_c|$，因此我们输出的每个分位数的误差至多为 $\varepsilon_0|\mathcal{S}_c|$。对于 $k_i \geq \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$，我们在$\epsilon$-近似中引入误差 $\epsilon|\mathcal{S}_\Delta|$，但由于每个组都采用精确计数，将各组的秩相加时总误差不会增加。因此，总误差至多为 $2\varepsilon_0|\mathcal{S}_c|$。在这两种情况下，只要 c足够大，总误差至多为 $\varepsilon_0 k$。

4.2.2 三维支配 AQS 的最优解

在本节中，我们修改上一节中的数据结构，将空间使用量降至线性级别。从空间分析可以看出，瓶颈在于具有 $k_i \leq \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$ 的浅切割。对于为这些层级构建的数据结构，前驱搜索结构在每个层级都占用线性空间，导致总体空间使用量超线性。为了解决这一问题，我们在构建 Ψ_i 之前，对这些层级的单元中的点进行秩空间缩减，从而可以利用整数寄存器为前驱搜索结构使用次线性空间。

秩空间缩减结构。我们考虑 $\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$-浅切割中的单元。令 $A = \log^{\kappa+1}\frac{1}{\varepsilon_0}$。对于浅切割单元 $\Delta \in \mathscr{C}$ 的冲突列表 $\mathcal{S}_\Delta$ 中的点，我们构建一个大小为 $A \times A \times A$ 的网格，使得网格的每个切片都包含 $O(1/(\varepsilon_0\log\frac{1}{\varepsilon_0}))$ 个点。每个网格点的坐标由其三个坐标在相应维度上的秩组成。对于 $\mathcal{S}_\Delta$中的每个 $O(\varepsilon_0 A/\log(1/\varepsilon_0))$ 个点，我们将其向下取整到不被它支配的最近网格点。这样可将点的坐标缩减到 $O(\log\log\frac{1}{\varepsilon_0})$ 位；现在我们可以应用上一小节中的次优解，从而得到一个空间复杂度为 $O(n)$ 的解。更具体地说，我们为 $\mathcal{S}_\Delta$ 中的超平面局部构建 $k_i \leq \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$ 的分层浅切割，并取足够大的常数 c，以值 $\varepsilon' = \varepsilon/c$ 应用前述解。

查询算法和查询时间。查询算法与前一个次优解法类似。唯一的区别是，当查询 q 所在的浅切割层级小于 $\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$ 时，我们使用秩空间缩减数据结构将 q 缩减到秩空间。设 q' 为将 q 缩减到秩空间后得到的网格点。注意，被 q 支配的点集可以表示为被 q' 支配的点集与被 q 支配的点的子集之并；后者位于包含 q 的 A 个网格条带中。我们使用基于网格点实现的数据结构，为前一个点集取得一个 ε'-分位数。关键观察是，包含 q 的条带中有 $O(\varepsilon_0^{-1}/\log\frac{1}{\varepsilon_0})$ 个点，因此我们可以在 $O(\frac{1}{\varepsilon_0})$ 时间内为这些点构建一个近似的 ε'-分位数摘要。随后，我们可以合并这两个分位数，并将答案作为结果返回。通过在 ε' 的定义中将 c 设得足够小，我们可以确保结果是一个有效的 ε_0 分位数摘要。这也给出了 $O(\frac{1}{\varepsilon_0})$ 的查询时间（在浅切割中定位到正确的单元 Δ 之后）。

正确性。由于我们在每个单元内构建了 $\frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$-浅切割中的 $k_i \le \frac{1}{\varepsilon_0}\log^{\kappa}\frac{1}{\varepsilon_0}$ 浅切割，因此变换后的坐标是一致的。如上所述，这会给计数 Ψ_i的输出带来误差，但由于我们随后会明确地纠正误差，得到的计数仍然是精确的。其余部分与次优解相同，因此我们的结构能够正确地找到 ε_0-分位数。

空间使用量。对于秩空间缩减结构，我们需要为查询存储一个前驱搜索结构，其空间与切片数量呈线性关系，即$O(A)$。我们为 A/ε_0-浅切割层级中的每个单元构建此结构，总共有$O(n\varepsilon_0/A)$个单元，因此空间使用量为 $O(n\varepsilon_0)$。根据浅切割的性质，在每个单元内构建浅切割只会使空间增加一个常数因子。

对于每个 Ψ_i，根据我们对次优解的分析，频率向量将占用 $O(\frac{N}{w^{1-\tau}})$ 空间。现在，由于点和查询的坐标都是大小至多为 A 的整数，编码一个坐标需要 $O(\log A) = O(\log\log\frac{1}{\varepsilon_0})$ 位。由于字长为 $w = \Omega(\log n)$，我们只需 $O(N_i \log_w A)$ 空间，即可为 Ψ_i 构建前驱搜索结构。总的来说，对于小于 A/ε_0的每个浅切割层级，我们需要 $O(\frac{N}{w^{1-o(1)}})$ 空间。因此，总空间使用量为 $O(n)$。我们以如下定理结束本节。

▶ **定理 12.** 给定由参数 $\varepsilon_0 > 0$ 和一组 P 个 n 点组成的输入，其中 $\mathbb{R}^3$ 中的每个点 $p \in P$ 都关联一个来自全序全集的权重 w_p，可以构建一个使用最优 $O(n)$ 空间的数据结构，使得对于任意支配查询 γ，该数据结构都能以最优查询时间 $O(\log n + \frac{1}{\varepsilon_0})$回答参数为 ε_0 的 *AQS* 查询。

5 开放问题

我们的结果带来了许多有趣的开放问题。首先，对于类型 2 颜色计数问题，我们为单纯形查询给出了一种线性规模的数据结构。目前尚不清楚增加空间是否能够缩短查询时间。确定该问题正确的时空权衡是一个引人入胜的开放问题。请注意，定理 8 中的查询时间取决于颜色总数。目前尚不清楚能否使查询时间达到输出敏感。这似乎很困难，而且遗憾的是，似乎没有合适的下界技术来解决这个问题。此外，鉴于改进精确单纯形范围计数的结果已经极具挑战性，考虑该问题允许乘法误差的近似版本是合理的。

其次，对于重击者查询，存在两个开放问题。在我们的解决方案中，

空间使用量在（$\frac{1}{\varepsilon}$）中仅额外增加了某个多对数因子，因此是最优的。一个有趣且富有挑战性的开放问题是，能否将空间使用量降至线性。另一方面，我们的查询时间并非输出敏感。严格来说，重频项的数量可能少于 $1/\varepsilon$，在这种情况下，如果能以（接近）线性空间，为 k 个输出重频项实现 $O(\log n + k)$查询时间，将会很有意义[2]。

第三，对于近似分位数摘要查询，我们针对半空间范围的数据结构并非最优。主要原因是，我们需要将类型二范围计数解决方案作为子程序。对于半空间范围，精确的类型二解决方案代价过高，因此我们不得不改用近似版本。这会引入一定误差，因而我们需要使用更小的误差参数，从而导致时间和空间方面都增加额外的多对数因子。相比之下，通过精确的类型二计数，我们为支配型近似分位数摘要查询获得了最优解决方案。目前看来，改进半空间查询的精确类型二结果颇具挑战，可能需要一些不同的思路来改进我们的结果。

最后，研究更广泛类别的几何范围上的近似分位数摘要、重频项摘要（或流式处理文献中使用的其他数据摘要或数据草图）也很有意义。在本文中，我们关注的是速度非常快的数据结构，最好具有最优的 $O(\log\ n + \frac{1}{\varepsilon})$ 查询时间，但我们知道，对于许多重要的几何范围，这类数据结构并不存在。例如，在使用线性空间时，单纯形查询需要 $O(n^{(d-1)/d})$ 时间，并且存在一些相匹配的下界。尽管如此，一个有趣的开放问题是：是否可以在线性或近线性空间内，用 $O(n^{(d-1)/d} + \frac{1}{\varepsilon})$ 时间为单纯形查询构建近似分位数摘要或重频项摘要；正如我们在引言中所述，一般方法得到的查询时间并非最优，即为 $O(n^{(d-1)/d} \cdot \frac{1}{\varepsilon})$ 或 $O(n^{(d-1)/d} + \frac{1}{\varepsilon^2})$。

参考文献

1 Peyman Afshani 和 Timothy M. Chan。三维空间中最优半空间范围报告。载于 *ACM-SIAM*离散算法年会（*SODA*）论文集，第 180–186 页，2009年。2 Peyman Afshani、Chris Hamilton 和 Norbert Zeh。缓存无关范围报告与近似范围计数的一般方法。《计算几何：理论与应用》，43:700–712，2010年。初步版本发表于 SoCG'09。3 Peyman Afshani 和 Jeff M. Phillips。独立范围采样，再探。载于计算几何研讨会（*SoCG*），第 4:1–4:13 页，2019年。4 Peyman Afshani 和 Konstantinos Tsakalidis。三维支配范围的最优确定性浅切割。*Algorithmica*，80(11):3192–3206，2018年。5 Peyman Afshani 和 Zhewei Wei。独立范围采样，再探。载于欧洲算法研讨会（*ESA*）论文集，第 87 卷，第 3:1–3:14 页，2017年。6 Pankaj K. Agarwal。几何划分及其应用。载于 离散与计算几何（新泽西州新不伦瑞克，*1989/1990*），*DIMACS* 离散数学、理论计算机科学丛书第 6 卷，第 1–37 页。美国数学学会，普罗维登斯，罗得岛州，1991年。7 Pankaj K. Agarwal。范围搜索。载于 J. E. Goodman、J. O'Rourke 和 C. Toth 编，《离散与计算几何手册》。CRC Press, Inc.，2016年。8 Pankaj K. Agarwal、Graham Cormode、Zengfeng Huang、Jeff M. Phillips、Zhewei Wei 和 Ke Yi。可合并摘要。载于 *ACM* 数据库系统原理研讨会（*PODS*）论文集，第 23–34 页，2012年。9 Djamal Belazzougui、Travis Gagie、J. Ian Munro、Gonzalo Navarro 和 Yakov Nekrich。数组中的范围多数元素与少数元素。*Algorithmica*，83(6):1707–1733，2021年。

[2] 我们感谢一位匿名审稿人提出"输出敏感"版本的建议。

10 Djamal Belazzougui、Travis Gagie 和 Gonzalo Navarro。参数化范围多数查询和少数查询的更优空间上界。载于《算法与数据结构》，计算机科学讲义第 8037 卷，第 121–132 页。Springer，海德堡，2013 年。11 Prosenjit Bose、Evangelos Kranakis、Pat Morin 和 Yihui Tang。近似范围众数和范围中位数查询。载于《计算机科学理论方面年度研讨会（STACS）论文集》，2005 年。12 Gerth Stølting Brodal、Beat Gfeller、Allan Grønlund Jørgensen 和 Peter Sanders。迈向最优范围中位数。理论计算机科学，412(24):2588–2601，2011 年。13 Timothy M. Chan。最优划分树。载于《计算几何研讨会（SoCG）》，第 1–10 页。ACM，2010 年。14 Timothy M. Chan、Qizheng He 和 Yakov Nekrich。彩色范围搜索的进一步结果。载于《计算几何研讨会（SoCG）》，第 164 卷，第 28:1–28:15 页，2020 年。15 Timothy M. Chan、Kasper Green Larsen 和 Mihai Pătraşcu。重新审视 RAM 上的正交范围搜索。载于《计算几何研讨会（SoCG）》，第 1–10 页，2011 年。16 Timothy M. Chan 和 Gelin Zhou。多维范围选择。载于《算法与计算》，计算机科学讲义第 9472 卷，第 83–92 页。Springer，海德堡，2015 年。17 B. Chazelle 和 J. Friedman。随机采样的确定性视角及其在几何学中的应用。组合数学，10:229–249，1990 年。18 Bernard Chazelle。用于分治的超平面切割。离散与计算几何，9(2):145–158，1993 年。19 K. L. Clarkson。随机采样在计算几何中的新应用。离散与计算几何，2:195–222，1987 年。20 Mark de Berg、Otfried Cheong、Marc J. van Kreveld 和 Mark H. Overmars。《计算几何：算法与应用》，第 3 版。Springer，2008 年。21 Stephane Durocher、Meng He、J. Ian Munro、Patrick K. Nicholson 和 Matthew Skala。常数时间和线性空间内的范围多数查询。信息与计算，222:169–179，2013 年。22 P. Gupta、R. Janardan 和 M. Smid。广义交集搜索问题的进一步结果：计数、报告和动态化。算法杂志，19(2):282–317，1995 年。23 David Haussler 和 Emo Welzl。ε-网与单纯形范围查询。离散与计算几何，2:127–151，1987 年。24 Xiaocheng Hu、Miao Qiao 和 Yufei Tao。独立范围采样。载于 Richard Hull 和 Martin Grohe 编，《第 33 届 ACM SIGMOD-SIGACT-SIGART数据库系统原理研讨会论文集》，PODS'14，美国犹他州斯诺伯德，2014 年 6 月 22–27 日，第 246–255 页。ACM，2014 年。 doi:10.1145/2594538.2594545。25 Zengfeng Huang 和 Ke Yi。分布式 ε-近似的通信复杂度。SIAM计算杂志，46(4):1370–1394，2017 年。26 Allan Grønlund Jørgensen 和 Kasper Green Larsen。范围选择与中位数：紧确的单元探测下界和自适应数据结构。载于《第 22 届 ACM-SIAM离散算法年会（SODA）论文集》，第 805–813 页，2011 年。27 Marek Karpinski 和 Yakov Nekrich。矩形中的高频颜色搜索。载于《第 20 届加拿大计算几何会议论文集》，加拿大蒙特利尔，2008 年 8 月 13–15 日，2008 年。28 Danny Krizanc、Pat Morin 和 Michiel Smid。列表与树上的范围众数和范围中位数查询。北欧计算机学报，12(1):1–17，2005 年。29 Jiří Matoušek。高效划分树。离散与计算几何，8(3):315–334，1992 年。30 Jiří Matoušek。切割超平面排列。离散与计算几何，6(5):385–406，1991 年。

31 Jiří Matoušek. 《几何差异：图解指南》. 算法与组合学. Springer Berlin Heidelberg，2009。32 Jeff M. Phillips. 核心集与草图。 *CoRR*，abs/1601.00617，2016。网址： http://arxiv.org/abs/1601.00617， arXiv:1601.00617。33 Rajeev Raman、Venkatesh Raman 和 Satti Srinivasa Rao。用于编码 k叉树、前缀和与多重集的简洁可索引字典。 *ACM* 算法汇刊（*TALG*），3(4)，2007年。34 Jiří Matoušek。利用高效的分层切割进行范围搜索。 离散与计算几何，10(2):157–182，1993年。35 Zhewei Wei 和 Ke Yi。超越简单聚合：面向摘要查询的索引编制。载于*ACM* 数据库系统原理研讨会（*PODS*）论文集，第117–128页。ACM，2011年。36 Ke Yi、Lu Wang 和 Zhewei Wei。面向摘要查询的索引编制：理论与实践。 *ACM* 数据库系统汇刊（*TODS*），39(1)，2014年1月。

附录

A 划分定理、切割与精确范围计数

划分定理是单纯形范围搜索中的一种标准工具。该定理由 Matoušek [29]于 20 世纪 90 年代初首次以如下形式证明。

▶ **定理 13** (Matoušek 的划分定理 [29])。 设 P 是 $\mathbb{R}^d(d \geq 2)$中的一个大小为 n 的点集，s和 $2 \leq s \leq n$是整数参数，并令 $t = \frac{n}{2s}$。可以将 P划分为 t个子集 P_1、⋯、 P_t，使得对于每个 $1 \leq i \leq s$， $s \leq |P_i| \leq 2s$都被包含在一个单纯形 Δ_i 中，并且 $\mathbb{R}^d$ 中的任意超平面的穿越数为 $O(t^{1-1/d})$，也就是说，它与 $O(t^{1-1/d})$ 个单纯形相交。

直接应用该定理，我们可以在近乎最优的时间内（相差对数$^{O(1)}$ n 因子）解决单纯形范围搜索问题。主要原因是，若用该定理构建具有超常数层级的划分树，穿越数中的常数将会急剧增大。为了达到最优查询时间，Chan [13] 对划分定理进行了改进和完善，并证明了如下版本。

▶ **定理 14** （Chan 的划分细化定理 [13]）。 设 P 是一个大小为 n 的$\mathbb{R}^d$ （$d \geq 2$）中的点集，H 是 $\mathbb{R}^d$中的一组 m 个超平面。假设给定了覆盖 P的 τ 个不相交单元，并且每个单元至多包含 P 中的 $2n/\tau$ 个点，且H 中的每个超平面至多穿过 ℓ 个单元。那么对于任意 b，我们都可以将每个单元细分为 O（b）个互不相交的子单元，总计至多 $b\tau$ 个子单元，使得每个子单元至多包含 P中的 $2n/$（$b\tau$）个点，并且 H 中的每个超平面穿过的单元总数至多为以下数量：

$$O((b\tau)^{1-1/d} + b^{1-1/(d-1)}\ell + b\log\tau\log m).$$

注意，Chan 版本中的穿越数包含一些额外项（$b^{1-1/(d-1)}l$ 和$b\log\ \tau\log\ m$）。当 $\tau = \log^{\Omega(1)} n$ 时，这些项被第一项（$(b\tau)^{1-1/d}$）控制；但在其他情况下，Chan 的界中的穿越数界可能比 Matoušek 的更差。对于划分定理的大多数应用，我们将构建一个层级为 $\Omega(\log\log\ n)$ 的划分树，在这种情况下，这些额外项可以安全地忽略。具体而言，对于单纯形范围计数问题，通过应用 Chan 方法，我们得到了线性空间结构的以下最优结果。

▶ **定理 15**（小空间精确单纯形范围计数 [34]）。设 P 是 $\mathbb{R}^d$ 中的一个包含 n 个点的集合。可以为单纯形范围构建一个大小为 $O(n)$ 的数据结构，使得给定一个查询范围 γ，可以在 $O(n^{(d-1)/d})$ 时间内报告 $|\gamma \cap P|$。

给定两个点 p，$q \in \mathbb{R}^d$，如果 p 的所有坐标都不小于 q的对应坐标，我们就称 p 支配 q，记作 $q \geq p$。支配范围 γ 由一个点p_γ 指定，定义为 $\{q \in \mathbb{R}^d : q \leq p_\gamma\}$。在上下文明确时，我们会交替使用 p_γ 和 γ。本文将关注半空间和支配范围的范围计数。不过，由于这两类范围都是单纯形范围的特殊情况，因此定理 15 适用。

如果我们希望以多对数时间更快地回答查询，一种方法是增加空间开销。这可以通过使用点线对偶 [20]并应用切割技术来实现。

给定一个单纯形 $\Delta \subset \mathbb{R}^d$ 和一组 H 超平面，我们用 H 表示与 Δ 内部相交的超平面集合$\mathcal{S}_\Delta$，称为 Δ的冲突列表。H 的 $(1/r)$-切割定义为由 $\mathscr{C}$ 个两两内部不相交的 d维闭单纯形组成的集合，它们共同覆盖 $\mathbb{R}^d$，并满足对所有 $\Delta \in \mathscr{C}$都有 $|\mathcal{S}_\Delta| \leq n/r$。切割$\mathscr{C}$的大小记作 $|\mathscr{C}|$，定义为其中单纯形的数量。切割的最优大小界及构造算法经过计算几何领域先驱们的一系列研究 [19, 23, 30, 6, 17] 后得以发展，并最终由 Chazelle 的工作 [18]完成。

▶ **引理 16**（切割引理 [18]）。给定 $\mathbb{R}^d$中的一组 H 个 n 超平面，对于任意 $r \geq 1$，存在一个大小为 $O(r^d)$ 的$1/r$）切割，其参数为 H （。切割及其冲突列表均可在确定性 $O(nr^{d-1})$ 时间内构造。

利用定理 [18], ，我们可以得到以下关于单纯形范围计数的快速查询时间结果。

▶ **定理 17**（快速查询精确单纯形范围计数 [34]）。设 P 为二维（分别为三维）中的一组 n 个点。可以构建一个大小为 $O(n^d)$ 的数据结构，用于处理单纯形范围查询；给定查询范围 γ，可以在 $O(\log n)$ 时间内报告 $|\gamma \cap P|$。

B 浅切割与近似范围计数

为了更形式化地定义浅切割，我们借用了文献 [2]中针对一般代数曲面的浅切割定义。设颜色数 F为一组连续且处处有定义的代数函数 $f : \mathbb{R}^d \to \mathbb{R}$，其维数 d 和次数 Δ均为常数。每个函数 $f \in \mathcal{F}$都定义了 $\mathbb{R}^{d+1}$中的一个连续曲面。给定任意点 $p = (p_1, p_2, \cdots\ p_{d+1}) \in \mathbb{R}^{d+1}$，若 $f(p_1, p_2, \cdots\ p_d) \leq p_{d+1}$，则称 f从 p 下方经过。

所有位于 颜色数 中的曲面的集合将 $\mathbb{R}^{d+1}$ 划分为由不相交单元组成的细分，这些单元的维度为 0、1、$\cdots$ $d+1$，并且共同覆盖整个 $\mathbb{R}^{d+1}$。我们称该细分为 颜色数的排列。给定任意点 $p \in \mathbb{R}^{d+1}$，其层级定义为低于该点的函数 $f \in$颜色数 的数量。我们将 颜色数 的 $(\leq k)$-层级定义为 $\mathbb{R}^{d+1}$ 中所有层级不超过 k的点的闭包。

颜色数 的（$\leq k$）层级的浅切割 $\mathscr{C}$（简称 k-浅切割）是由一组不相交单元组成的集合，它们共同覆盖颜色数 的（$\leq k$）层级，并具有以下性质：切割中的每个单元 $C \in \mathscr{C}$ 都与颜色数$_C$ 中的 O 个（k）函数相交。我们将颜色数$_C$ 称为 C 的冲突列表。

对于一个集合 颜色数，如果它具有以下性质，则称其为可浅切割的：**1.** 对于每个 $k \in \mathbb{N}$，都存在一个 $(\leq k)$-浅切割，适用于 颜色数，且该浅切割所具有的单元数量有上界 $O(|$颜色数$|/k)$。

2. 浅切割可以在时间 $O(|\text{颜色数}|\log|\text{颜色数}|)$内构建。3. 给定一个 α^i-浅切割的层次结构，存在一个线性规模的数据结构，使得对于任意给定的查询点 q，我们都可以在时间 α^i内找到包含 $\mathscr{C}$ 的 $C \in \mathscr{C}$-浅切割，以及其中具有最小 q 的单元 i，所需时间为 $O(\log|\text{颜色数}|)$。

对于半空间和支配范围搜索问题，在将问题转换到对偶空间后，通常会使用浅切割；在对偶空间中，输入和输出的角色会“互换”。已有研究表明，二维和三维中的半空间范围与支配范围均可浅切割 [2]。更一般地，对于可浅层切割范围，我们可以在线性空间内以对数时间近似计算点的数量。

<escape>▶ </escape>**定理 2** （近似范围计数 <escape>[2]</escape>）。设 <escape>P </escape>是 <escape>$\mathbb{R}^3$</escape> 中的一个由 <escape>n </escape>个点组成的集合。对于半空间或支配范围，可以构建一个大小为 <escape>O</escape>（<escape>n</escape>）的数据结构，使得给定一个查询范围 <escape>γ</escape>，可以在 <escape>$|\gamma \cap P|$ </escape>（log <escape>n</escape>）时间内报告结果，误差为 <escape>$\alpha|\gamma \cap P|$ </escape>，其中任意常数 <escape>$\alpha > 0$</escape> 均适用。

我们提到，浅切割的概念也可以在原始空间中定义，即针对一组 P 点。例如，对于支配范围，给定任意点 $p \in P$，我们可以将其层级定义为 P 中被 p支配的点的数量。对于 P，一个（原始） k-浅切割是 $\mathbb{R}^d$ 中的一组 $\mathscr{C}$ 点，满足： P 中层级至多为 k的每个点都被 $\mathscr{C}$ 中的某个点支配，并且 $\mathscr{C}$ 中的每个点都支配 P中的 O（k）个点。研究表明，这样的浅切割可以高效地构造，并且我们可以像在对偶空间 [4]中的浅切割层次结构中一样，高效地找到一个浅切割层级和浅切割层次结构中的一个单元。

C 类型二彩色单纯形范围计数

本节研究针对 $\mathbb{R}^d$ 中单纯形查询的“类型2”彩色范围计数问题。该问题的输入是 $\mathbb{R}^d$ 中的一个 n 点集，且每个点都被赋予 F 颜色集合中的一种颜色。我们希望构建一种数据结构，使得给定一个查询单纯形 q 后，能够报告 q 中每种颜色对应的点数。换言之，我们希望输出一个颜色的“频率向量”。形式化地，我们将频率向量定义如下。

▶**定义 18** (频率向量)。给定点集 P，其中每个点都被赋予 F 种颜色中的一种，P 的频率向量定义为 $\mu(P) = (\mu_1, \ldots, \mu_F)$，其中 μ_i 是 P 中第 i 种颜色的频率。

在给出完整细节之前，我们先介绍数据结构的总体思路。根据 F/w 的不同取值，我们将构建不同的结构。当 $F/w > {}^{n}\log^4 n$ 时，我们使用 Matoušek 定理 13 构建划分树 T。否则，我们使用 Chan 定理 14 构建划分树 T。对于 T 的每个顶点 v，我们将对应于 v 的划分单纯形记为 $v(\mathcal{R})$。我们将叶节点 T 上的彩色点显式存储。对于每个内部顶点 v，我们仅存储 $v(\mathcal{R})$ 中点的频率向量。频率向量将以一种紧凑的方式表示，稍后我们会对此进行说明。

查询算法如下。给定查询 q，我们查询相应的划分树 T。在此过程中，我们维护一个累积频率向量，即到目前为止见到的所有不同颜色的总数（采用紧凑表示）。我们从 T 的根节点 r 开始。如果 q 完全包含 v（$\mathcal{R}$），那么我们将 v（$\mathcal{R}$）的频率向量加到累积频率向量中。如果 q 不包含 v（$\mathcal{R}$），但与其有非空交集，那么我们遍历其子节点。如果 q 与 v（$\mathcal{R}$）的交集为空，则不做任何操作。如果 v 是 T 的叶节点，我们就明确计数 v（$\mathcal{R}$）中每种颜色的频率，并更新频率向量。

查询总时间包括访问内部顶点的频率向量的紧凑表示，以及对遍历到的叶节点进行显式计数。下面我们介绍具体细节。首先从紧凑表示的概念说起。

紧凑表示。考虑一个频率向量 $\mu(\mu_1, \ldots\ldots, \mu_F)$，并令 $s = \sum_{i=1}^{F} \mu_i$。我们希望使用 $O(F \log(2(s+2F)/F))$ 位来存储 μ，同时支持以下两种操作：其一是“加法操作”，将两个此类向量的表示相加，以获得其和的表示；其二是“提取”操作，对于给定的 i，在常数时间内检索第 i 个频率 μ_i。

我们首先介绍一种使用第三个字符 #（除 0 和 1 之外）的编码方式：将 μ_i 以二进制编码，并在它们之间插入字符 #。只需将字符 0、1 和 # 重新编码为二进制，即可轻松将该表示转换为二进制，这会使编码大小增加一倍。令 $L = 2\sum_{i=1}^{F}(\log(\mu_i + 2) + 1)$，即该表示的位长度。根据算术平均数与几何平均数不等式以及 $\sum_{i=1}^{F} \mu_i = s$，

$$L = 2\sum_{i=1}^{F} \log(\mu_i+2)+1 = 2F+2\log\prod_{i=1}^{F}(\mu_i+2) \le 2F+2F\log\frac{s+2F}{F} = 2F\log\frac{2(s+2F)}{F}.$$

我们将该表示打包为 $O(\lceil L/w \rceil)$ 个字长的字，并对每个打包字存储其中包含的 μ_i 数量。令 c_j 表示编码在第 j 个字中的频率数量。注意，$c_j = O(w)$，因此我们可以将它们存储在一个支持常数查询时间的前缀和数据结构中。至此，打包表示的描述完毕。

注意，可以使用非标准字操作或查表法[3]，以一种直接的方式将两个表示相加。

现在考虑索引为 i 的提取操作。我们需要找到最小的索引 j，使得截至 j 的前缀和超过 i。然而，由于每个 c_j 至多为 w，因此可以采用一元编码，而提取问题就等价于位向量上的 `select` 操作；在建立一个耗时线性时间 [33] 的索引后，该操作可在 $O(1)$ 时间内完成。总而言之，我们证明了如下结论。

▶**引理19**。对于任意一个至多包含 s 个点的集合，其中每个点被赋予 F 种颜色中的一种，我们可以使用 $O(1 + \frac{F}{w}\log^{2}_{(s+2F)\,F})$ 个字长的紧凑表示对其频率向量进行编码，其中 w 为字长。此外，我们可以在 $O(1 + \frac{F}{w}\log^{2}_{(s+2F)\,F})$ 时间内将两个紧凑表示相加，并在 $O(1)$ 时间内提取第 i 种颜色的频率，条件是 $1 \le i \le F$。

数据结构。现在我们描述数据结构的具体细节。如前所述，我们基于 Matoušek 的早期版本（定理 13）以及 Chan 的较新版本（经过改进且更为简洁）（定理 14）构建划分树。我们需要这两种结构的主要原因是，尽管 Chan 的版本在许多方面更为优越，但它无法处理划分树非常浅的情况（这在我们的应用中可能发生）。幸运的是，Matoušek 的早期版本能够在这种情况下给出最优界。因此，我们将针对不同情况使用不同的划分树。

▶ **定理 8.** 给定一个整数参数 F，以及一个包含 n 个点的集合 P（这些点位于 $\mathbb{R}^d$ 中，每个点都被赋予集合 $[F]$ 中的一种颜色），可以构建一个线性规模的数据结构，使得给定

[3] 例如，将其打包成大小为 $\log_2 n$ 的字，然后构建一个大小为 n 的表，以支持在常数时间内执行加法操作。

一个查询单纯形 q，它能够在总时间最大$\{O(n^{(d-1)/d})\ O(n^{(d-1)/d}F^{1/d}/w^{\alpha})\}$内输出每种颜色在 $P\cap q$ 中出现的次数，其中 α 为某个适当的常数，字长为 w。

证明。 我们根据 $\frac{F}{w}$ 的取值构建两棵划分树。两种情况的基本情形相同：当子问题的规模小于 F/w 时，我们停止并通过暴力算法检查所有点。

当 $\frac{F}{w}\geq \frac{n}{\log^4 n}$ 时，我们构建一棵采用 Matoušek 版本（定理 13）的划分树 T，其扇出为 $t=w^{\delta}$，其中 $\delta>0$ 足够小。对于 T 中的每个节点，我们以紧凑表示存储以该节点为根的点的频率向量。查询时间 $Q(n)$ 满足以下递推关系

$$Q(n)=O(t^{\frac{d-1}{d}})Q\left(\frac{n}{t}\right)+O(t)\cdot O\left(1+\frac{F}{w}\log\frac{2(n/t+2F)}{F}\right).$$

运行 k 步后，我们有，

$$Q(n)\leq c_0^k t^{k\frac{d-1}{d}}Q\left(\frac{n}{t^k}\right)+\sum_{i=1}^{k}c_1 t^{\frac{1}{d}}t^{i(\frac{d-1}{d})}\cdot\left(1+\frac{F}{w}\log\frac{2(n/t^i+2F)}{F}\right),\tag{5}$$

对于某些常数 c_0，c_1。

注意，当 $\frac{n}{t^i}\geq 2F$ 时，$\log\frac{2(n/t^i+2F)}{F}\leq\log\frac{4n/t^i}{F}$，因此不等式 5 中的求和有如下上界

$$\sum_{i=1}^{k}c_1 t^{\frac{1}{d}}t^{i(\frac{d-1}{d})}\cdot\left(1+\frac{F}{w}\log\frac{4n}{Ft^i}\right)=c_1\sum_{i=1}^{k}(t^{\frac{1}{d}}t^{i(\frac{d-1}{d})})+t^{\frac{1}{d}}t^{i(\frac{d-1}{d})}\cdot\left(\frac{F}{w}\log\frac{4n}{Ft^i}\right).$$

令 $f(i)=t^{i(\frac{d-1}{d})}\log\frac{4n}{Ft^i}$，则有

$$\frac{f(i+1)}{f(i)}=t^{\frac{d-1}{d}}\left(1+\frac{\log\frac{1}{t}}{\log\frac{4n}{Ft^i}}\right)=t^{\frac{d-1}{d}}\left(1+\frac{\log t}{\log\frac{Ft^i}{4n}}\right)\geq t^{\frac{d-1}{d}}\left(1+\frac{\log t}{\log\frac{w}{4}}\right)\geq w^{\delta(\frac{d-1}{d})}=\omega(1),$$

其中，第一个不等式由 $\frac{n}{t^i}\geq\frac{F}{w}$ 得出，最后一个不等式则由 $t=w^{\delta}$ 得出，其中某个常数为 δ。当 $\frac{n}{t^i}<2F$ 时，不等式 5 中的求和上界为

$$\sum_{i=1}^{k}c_1 t^{\frac{1}{d}}t^{i(\frac{d-1}{d})}\cdot\left(1+\frac{F}{w}\log\frac{2(2F+2F)}{F}\right)\leq\sum_{i=1}^{k}c_1 t^{\frac{1}{d}}t^{i(\frac{d-1}{d})}\cdot\left(1+\frac{F}{w}\log 8\right).$$

在这两种情况下，求和都是几何级数，并且由最后一项主导，即当 i 取得最大值时。

由于 $\frac{F}{w}\geq\frac{n}{\log^4 n}$，我们注意到，应用 Matoušek 方案若干层级后，就会到达基本情形，因为当子问题规模小于 $\frac{F}{w}$ 时，我们便停止。因此有 $\frac{F}{w^{1+\delta}}\leq\frac{n}{t^k}\leq\frac{F}{w}$，这意味着 $t^k\leq\frac{nw^{1+\delta}}{F}$ 和 $t^k\geq\frac{nw}{F}$。将这些不等式代入公式 5，并利用 $Q(n/t^k)=O(\frac{F}{w})$，可得到以下界

$$Q(n)=O\left(\frac{n^{\frac{d-1}{d}}F^{\frac{1}{d}}}{w^{\frac{1}{d}-\delta}}\right).\tag{6}$$

当 $\frac{F}{w}\leq\frac{n}{\log^4 n}$ 时，我们使用 Chan 的版本（定理 14）来构建划分树。这里的观察是，我们可以将点集划分为至少 $\tau=\log^4 n$ 个子集。现在，穿越数至少为 $\log^2 n$，因此主项 $O((b\tau)^{1-1/d})$ 占主导地位。

我们使用参数为 $b = \Theta(1)$ 的 Chan 方法（定理 14）构建划分树。同样，我们将相应的频率向量附加到内部节点。在这种情况下，查询时间$Q(n)$ 的上界为

$$Q(n) = O(n^{1-1/d}) + O\left(\sum_{i=0}^{\beta-1} bl(b^i) \cdot \left(1 + \frac{F}{w} \log \frac{2(\frac{n}{b^i} + 2F)}{F}\right)\right),$$

其中，第一项是 Chan 版本中遍历划分树的标准开销，第二项是检查所有完全包含在 q 中的不相交单元的开销，而参数 $\beta = \log_b \frac{n}{F/w} = \Omega(\log \log n)$ 和 $l(u)$ 是 u 个单元的穿越数。注意，根据我们之前类似的分析，求和由最后一项主导，即，

$$\begin{aligned}\sum_{i=0}^{\beta-1} bl(b^i) \cdot \left(1 + \frac{F}{w} \log \frac{2(\frac{n}{b^i} + 2F)}{F}\right) &= O\left(b\left(\frac{nw}{bF}\right)^{\frac{d-1}{d}} \cdot \left(1 + \frac{F}{w} \log \frac{2(\frac{F}{w} + 2F)}{F}\right)\right) \\ &= O\left(\frac{n^{\frac{d-1}{d}} F^{\frac{1}{d}}}{w^{\frac{1}{d}}}\right).\end{aligned}$$

因此，在这种情况下，$Q(n) = O\left(\frac{n^{\frac{d-1}{d}} F^{\frac{1}{d}}}{w^{\frac{1}{d}}}\right)$。注意，公式 6 中的界略大，因此我们的查询时间由公式 6 界定。

◀

D 近似重频摘要查询

本节中，我们证明以下关于半空间和支配范围的近似重频项摘要查询结果。

▶ **定理 5. </style}对于，近似半空间重频项摘要查询可以使用 (**$n\log_w(1/\varepsilon_0)$) 空间，并具有最优的 $O(\log n + 1/\varepsilon_0)$ 查询时间。

▶ **定理 6. </style}对于，可以使用 (**$n \log\log_w(1/\varepsilon_0)$) </style空间来回答近似半空间重频项摘要查询，查询时间则达到最优的 (log $n + 1/\varepsilon_0$) *</style*。

▶ **定理 7.** 对于 $d = 2, 3$，可以使用最优的 $O(n)$ 空间来回答近似支配重频摘要查询，查询时间则达到最优的 $O(\log n + 1/\varepsilon_0)$ *</style*。

我们先介绍一种适用于更一般单纯形范围的基础数据结构，它将作为最终 AHHS 解决方案的构建模块。

D.1 AHHS 查询的基础解决方案

我们的基础数据结构针对以下问题：

▶**问题 3.** [粗粒度 *AHHS* 查询] 设 P 是 $\mathbb{R}^d$ 中的一个点集，每个点都关联一种颜色。问题是将 P 存储在一个结构中，使得给定查询范围 q 时，能够高效估算 $q \cap P$ 中各颜色的频次，且加性误差不超过 $\varepsilon|P|$，其中 $0 < \varepsilon < 1$ 是某个参数。

请注意，与原始的 AHH 查询问题 2 相比，这是一个更粗略的问题，因为误差固定为 $\varepsilon|P|$，且独立于查询范围。我们针对这个基本问题证明以下定理：

▶**定理 9**。 对于 $d \geq 3$，单纯形查询的问题 *3*（ $d+1$ 个半空间的交集）可以使用 $O(X)$ 空间来求解，其中空间复杂度为 $X = \min\{|P|, \varepsilon^{-\frac{2d}{d+1}}\}$，查询时间为

$$O\left(\frac{|P|^{1-\frac{2}{d-1}}}{w^{\alpha}\varepsilon^{\frac{2}{d-1}}}\right) + O\left(X^{\frac{d-1}{d}}\right)$$

其中 w 是机器的字长， α 是某个正常数。

我们将以不同方式处理频率不同的颜色。为简化说明，我们作如下定义：

▶**定义 20.** 设 P 是一个带有颜色的点集。设 P_i 是颜色为 i的点集。若颜色 i 满足 $|P_i| \geq (\varepsilon|P|)^{\frac{2d}{d-1}}$，则称其为大颜色；否则称其为小颜色。对于小颜色 i，若它满足 $\delta(\varepsilon|P|)^{\frac{2d}{d-1}} \leq |P_i| \leq 2\delta(\varepsilon|P|)^{\frac{2d}{d-1}}$，则称其为 δ-小。

注意，简单计算表明，若 $|P| > \varepsilon^{-\frac{2d}{d+1}}$，则不存在大颜色，因为大颜色定义中的界限超过了 $|P|$。

数据结构。 我们的思路是对大颜色进行精确计数，并通过 ε-近似对小颜色进行近似计数。我们将大颜色对应的点收集到集合 P_{Big}中，并利用定理 8 在 P_{Big} 上构建数据结构 Ψ_{Big}。对于每个小颜色 i，我们首先计算 P_i 的一个 ε_i-近似 P_i'，其中 $\varepsilon_i = \varepsilon|P|/|P_i|$。给定一个值 $\delta \leq 1$，令 P_δ为颜色属于 δ-小的点集，并令 P_δ' 为在 δ-小颜色上构建的近似的并集。对于每个 $\delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots, \frac{1}{2}(\varepsilon|P|)^{-\frac{d+1}{d-1}}$，我们都构建 P_δ'。我们可以将属于 δ-小的颜色数量界定为 $F_\delta = \Theta(|P_\delta|/(\delta(\varepsilon|P|)^{\frac{2d}{d-1}}))$。我们还使用定理 8，将$P_\delta'$ 存储在数据结构 Ψ_δ 中。

查询算法。 查询过程如下。给定一个查询单纯形范围q，我们查询数据结构 Ψ_{Big} 以及所有数据结构 Ψ_δ，对每个 $\delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots, \frac{1}{2}(\varepsilon|P|)^{-\frac{d+1}{d-1}}$。我们检查每个输出的颜色频率向量，并在其频率大于 $\varepsilon|P|$ 时输出该颜色。

正确性显而易见，因为我们要么精确计数颜色，要么使用 ε_i-近似。在后一种情况下，误差至多为 $\varepsilon_i|P_i| = \varepsilon|P|$。由于我们可以忽略频率不超过 $\varepsilon|P|$ 的颜色，因此对于$\delta \geq \frac{1}{2}(\varepsilon|P|)^{-\frac{d+1}{d-1}}$，只需考虑 δ-小颜色。

空间上界。 设 $Y = |P_{\text{Big}} \cup \bigcup_{i \in small} P_i'|$。我们的数据结构的空间使用量显然有界于 $O(Y)$。现在我们证明 $Y = O(X)$，从而证得空间上界的主张。

首先，我们界定每个小颜色 i 的 ε_i-近似大小。考虑一个 δ-小颜色 i 以及集合 P_i。根据构造，我们使用的是一个 $\varepsilon_i = \varepsilon|P|/|P_i|$ 近似，其大小为

$$\begin{aligned} O\left(\left(\frac{1}{\varepsilon_i}\right)^{\frac{2d}{d+1}}\right) &= O\left(\left(\frac{|P_i|}{\varepsilon|P|}\right)^{\frac{2d}{d+1}}\right) = O\left(\left(\frac{2\delta(\varepsilon|P|)^{\frac{2d}{d-1}}}{\varepsilon|P|}\right)^{\frac{2d}{d+1}}\right) \\ &= O\left(\left(\delta(\varepsilon|P|)^{\frac{d+1}{d-1}}\right)^{\frac{2d}{d+1}}\right) = O\left(\delta^{\frac{d-1}{d+1}}|P_i|\right) = O(|P_i|), \end{aligned} \tag{7}$$

其中倒数第二个等式由 $\delta(\varepsilon|P|)^{\frac{2d}{d-1}} \le |P_i|$ 得出。因此，每个 ε_i-近似的大小至多为 $O(|P_i|)$，因为 $\delta \le 1/2$，所以 $Y \le |P|$。

当 $|P| > \varepsilon^{-\frac{2d}{d+1}}$，正如我们所论证的，不存在大颜色，所有颜色都是小颜色。在这种情况下，我们可以界定

$$Y \le \left| \bigcup_{i \in small} P_i' \right| = \sum_i O\left(\left(\frac{1}{\varepsilon_i}\right)^{\frac{2d}{d+1}}\right) = O\left(\sum_i \left(\frac{|P_i|}{\varepsilon|P|}\right)^{\frac{2d}{d+1}}\right) = O\left(\left(\frac{1}{\varepsilon}\right)^{\frac{2d}{d+1}}\right), \tag{8}$$

其中，最后一个等式由函数 $x^{\frac{2d}{d+1}}$ 的凸性得出，因此当所有 $|P_i|$ 的值都为 0，且其中一个等于 $|P|$时，求和达到最大值。因此我们证明了 $Y \le X = \min\{|P|, \varepsilon^{-\frac{2d}{d+1}}\}$，所以 $O(X)$ 是空间上界。

查询时间上界。还需分析查询时间。对于 P_{Big}，注意颜色数不超过 $F_{Big} = |P|/(\varepsilon|P|)^{\frac{2d}{d-1}}$。根据定理 8，我们为 P_{Big} 构建的数据结构的查询时间为

$$\begin{aligned} O(|P_{Big}|^{(d-1)/d} + |P_{Big}|^{(d-1)/d} F_{Big}^{1/d}/w^\alpha) &= O\left(|P|^{(d-1)/d}\right) + O\left(\frac{|P|^{\frac{d-1}{d}}|P|^{\frac{1}{d}}}{w^\alpha(\varepsilon|P|)^{\frac{2}{d-1}}}\right) \\ &= O\left(X^{(d-1)/d}\right) + O\left(\frac{|P|^{1-\frac{2}{d-1}}}{w^\alpha \varepsilon^{\frac{2}{d-1}}}\right), \end{aligned}$$

对于某个常数 α，其中第一个等式由 $P_{Big} \subset P$ 得出，最后一个等式则由大颜色存在时的 $|P| \le \varepsilon^{-\frac{2d}{d+1}}$ 得出，此时根据定义有 $X = |P|$。现在我们来界定小颜色的查询时间。根据公式7，为 δ个小颜色存储的 ε-近似的总大小为 $n_\delta = |P_\delta'| = O(\delta^{\frac{d-1}{d+1}}|P_\delta|)$。根据定理 8，在 P_δ' 上构建的数据结构的查询时间为

$$O\left(\max\left\{n_\delta^{(d-1)/d}, n_\delta^{(d-1)/d} F_\delta^{1/d}/w^\alpha\right\}\right) = O(n_\delta^{(d-1)/d}) + O(n_\delta^{(d-1)/d} F_\delta^{1/d}/w^\alpha). \tag{9}$$

小颜色的总查询时间是对所有 δ 的公式9中表达式求和。我们可以分别对这两个项进行界定。

首先注意到，

$$\sum_\delta \left(\delta^{\frac{d-1}{d+1}} \left(\frac{|P_\delta|}{|P|}\right)\right)^{(d-1)/d} \le \sum_\delta \delta^{\frac{(d-1)^2}{d(d+1)}} = O(1),$$

因为根据定义，求和是针对 δ 和 $|P_\delta| \le |P|$ 的几何递减值。当 $X = \min\{|P|, \varepsilon^{-\frac{2d}{d+1}}\} = |P|$ 时，这意味着

$$\sum_\delta n_\delta^{\frac{d-1}{d}} = \sum_\delta (\delta^{\frac{d-1}{d+1}}|P_\delta|)^{(d-1)/d} \le |P|^{\frac{d-1}{d}} \le X^{\frac{d-1}{d}}.$$

当 $X = \min\{|P|, \varepsilon^{-\frac{2d}{d+1}}\} = \varepsilon^{-\frac{2d}{d+1}}$ 时，根据定义，

$$\begin{aligned} \sum_\delta n_\delta^{\frac{d-1}{d}} = \sum_\delta \left(\sum_{i:\delta\text{-small}} \left(\frac{1}{\varepsilon_i}\right)^{\frac{2d}{d+1}}\right)^{\frac{d-1}{d}} &= \sum_\delta \left(\sum_{i:\delta\text{-small}} \left(\frac{|P_i|}{\varepsilon|P|}\right)^{\frac{2d}{d+1}}\right)^{\frac{d-1}{d}} \le \sum_\delta \left(\frac{|P_\delta|}{\varepsilon|P|}\right)^{\frac{2(d-1)}{d+1}} \\ &\le \left(\frac{|P|}{\varepsilon|P|}\right)^{\frac{2(d-1)}{d+1}} = \left(\left(\frac{1}{\varepsilon}\right)^{\frac{2d}{d+1}}\right)^{\frac{d-1}{d}} = X^{\frac{d-1}{d}}, \end{aligned}$$

其中，第一个不等式由 $\frac{2d}{d+1} \geq 1$ for $d \geq 1$ 得出，第二个不等式则由 $\frac{2}{(d-1)}\frac{}{d+1} \geq 1$ when $d \geq 3$ 得出。

对于第二项，由于 $F_\delta = O(|P_\delta|/(\delta(\varepsilon|P|)^{\frac{2d}{d-1}}))$，

$$\sum_\delta O\left(\frac{n_\delta^{\frac{d-1}{d}} F_\delta^{\frac{1}{d}}}{w^\alpha}\right) = \sum_\delta O\left(\frac{(\delta^{\frac{d-1}{d+1}}|P_\delta|)^{\frac{d-1}{d}}}{w^\alpha} \cdot \left(\frac{|P_\delta|}{\delta(\varepsilon|P|)^{\frac{2d}{d-1}}}\right)^{\frac{1}{d}}\right) = \sum_\delta O\left(\frac{\delta^{\frac{d-3}{d+1}}|P_\delta|}{w^\alpha(\varepsilon|P|)^{\frac{2}{d-1}}}\right)$$
$$= O\left(\frac{(\sum_\delta \delta^{\frac{d-3}{d+1}})|P|}{w^\alpha(\varepsilon|P|)^{\frac{2}{d-1}}}\right) \leq O\left(\frac{|P|^{1-\frac{2}{d-1}}}{w^\alpha \varepsilon^{\frac{2}{d-1}}}\right),$$

其中，第三个等号是根据以下观察得出的：所有 n_δ 的和至多为 $|P|$ 最后一个等号则是因为 $d \geq 3$ 和 $\delta \leq \frac{1}{2}$ 构成一个递减几何级数。

D.2 三维 AHHS 查询

现在，我们将说明如何回答针对半空间和支配范围的三维重频项查询。我们给出以下两个结果。

▶ **定理 5**。对于 $d = 3$，可以使用 $O(n\log_w(1/\varepsilon_0))$ 空间回答近似半空间重频项摘要查询，且查询时间达到最优的 $O(\log n + 1/\varepsilon_0)$。

▶ **定理 7**。对于 $d = 2,3$，可以使用最优的 $O(n)$ 空间回答近似支配重频项摘要查询，且查询时间达到最优的 $O(\log n + 1/\varepsilon_0)$。

D.2.1 三维 AHHS 查询的数据结构

令 h 为查询半空间（或支配范围），并令 k 为 h中的点数（即 $k = |h \cap P|$）。我们考虑对偶空间。令 H 为与 P对偶的超平面集合，并令 q 为 h的对偶。利用定理 2，我们可以在 $O(\log n)$ 时间内，使用线性空间，求得 k 的一个 $(1+\alpha)$ 因子近似 k^*。在我们的数据结构中，将使用参数 β, λ构造一系列浅切割。根据处理的是半空间范围还是支配范围，这些参数的取值会有所不同。尽管如此，由于已知 k^* 的值，我们可以选择正确的层级进行查询。为进一步说明，我们考虑以下切割。

- 基础切割：我们为 ε_0^{-1}层构建一个基础浅切割。低层切割：为 ε_0^{-1}到 $\varepsilon_0^{-1}w^{\beta\lambda}$之间的层级构建，其中 $w = \Theta(\log n)$ 为字长，而 β 和 λ 是稍后设定的参数。具体来说，我们为 $i = 1, \cdots \lambda$ 构建 $\varepsilon_0^{-1}w^{\beta i}$-浅切割。高层切割：我们为 $i = 1, \cdots, \log(n\varepsilon_0/w^{\beta\lambda})$ 构建 $\varepsilon_0^{-1}w^{\beta\lambda}2^i$-浅切割。

该数据结构根据浅切割存储不同的辅助数据结构。因此，由于该数据结构知道 k^*，它便能选择正确的层级进行查询。例如，如果 $k^* < \varepsilon_0^{-1}$，则使用为基础切割构建的辅助数据结构回答查询。类似地，当 $\varepsilon_0^{-1} \leq k^* \leq \varepsilon_0^{-1}w^{\beta\lambda}$ 时，该数据结构使用低层切割；否则，它使用高层切割。

我们依次介绍它们。首先介绍高层切割。

D.2.1.1 高层切割。

这些切割的数据结构很简单。对于高层切割中的每个单元 Δ，我们为其冲突列表 $\mathcal{S}_\Delta$ 构建一个 $c_1\varepsilon_0$-近似 E_Δ，并将其存储在参数为 $c_1\varepsilon_0$ 的基础结构中。

查询算法。回想一下，当 k^*超过 $\varepsilon_0^{-1}w^{\beta\lambda}$ 时，查询会探查高层切割。在这种情况下，它可以找到满足 $|\mathcal{S}_\Delta| = O(k^*)$ 的单元 $\Delta \in \mathscr{C}_i$。我们查询基于 $\mathcal{S}_\Delta$ 构建的基础数据结构，并将结果作为 AHHS 返回。此处的查询时间为 $O(\varepsilon_0^{-1})$，这是由基础结构直接得出的。

正确性。考虑一种颜色 c，它在从 q下方经过的超平面中出现的频率为 $j\varepsilon_0 k$，其中 $j \geq 1$。由于 E_Δ 是 $\mathcal{S}_\Delta$的一个 $c_1\varepsilon_0$-近似，因此在从 q下方经过的 E_Δ 子集中，c 出现 $j\varepsilon_0|E_\Delta| \pm c_1\varepsilon_0|E_\Delta|$ 次。如果将 c_1 选为足够小的常数（例如 $c_1 < 0.5$），那么 c 在 E 中出现的频率足够高，以至于基础数据结构也会报告 c 及其频率，误差不超过 $c_1\varepsilon_0|\mathcal{S}_\Delta|$。因此，我们可以将 c 在 q 下方的超平面中的频率近似到加性误差 $2c_1\varepsilon_0 k$以内。

空间使用量。这里，我们需要区分半空间和支配范围。对于半空间，E_Δ 的大小上界为 $O(\varepsilon_0^{-3/2})$；而对于支配范围，根据定理 4，其大小上界为 $O(\varepsilon_0^{-1}\log^4 \frac{1}{\varepsilon_0})$。因此，对于半空间，我们有如下情况：对于一个单元 $\Delta \in \mathscr{C}_i$，其中 $\mathscr{C}_i$ 是 $\varepsilon_0^{-1}w^{\beta\lambda}2^i$-浅切割，根据定理 9，为 E_Δ 构建的基础数据结构的空间上界为 $O(\varepsilon_0^{-3/2})$，无论其他因素如何，总空间消耗因此有如下上界：

$$\sum_{i=1}^{\infty} O\left(\varepsilon_0^{-3/2}\frac{n}{\varepsilon_0^{-1}w^{\beta\lambda}2^i}\right) = O(n)$$

这可通过选取 λ 来得出，使得对于足够小的 β，$w^{\beta\lambda} = \varepsilon_0^{-1/2}$ 成立。

对于支配范围，只需选取 λ，使得 $\log^4 \frac{1}{\varepsilon_0} \leq w^{\beta\lambda} \leq \varepsilon_0^{-1/2}$。我们需要将 β 选为足够小的常数。不过，这意味着在支配情形下，λ 也可以选为常数。这里的总空间消耗上界为

$$\sum_{i=1}^{\infty} O\left(\varepsilon_0^{-1}\log^4\frac{1}{\varepsilon_0}\cdot\frac{n}{\varepsilon_0^{-1}\log^4\frac{1}{\varepsilon_0}\cdot 2^i}\right) = O(n).$$

D.2.1.2 较低层级切割。

对于低层切割中的每个单元 Δ，我们存储冲突列表 高频 颜色的列表，这些颜色在 Δ 的冲突列表 $\mathcal{S}_\Delta$ 中至少出现 $\frac{c_1\varepsilon_0}{w^\beta}|\mathcal{S}_\Delta|$ 次。因此，对于每个切割单元，我们在定理 9的基础数据结构中存储 $\frac{w^\beta}{c_1\varepsilon_0}$ 个高频颜色，因此所有低层切割所需的总空间为

$$\sum_{i=1}^{\lambda} O\left(\frac{n}{w^{\beta i}/\varepsilon_0}\cdot\frac{w^\beta}{c_1\varepsilon_0}\right) = O(n).$$

这也使我们能够对颜色进行重新编号：考虑低层切割中的一个 Δ。我们将 Δ 中的候选颜色从 1 开始重新编号，直到 $w^\beta\varepsilon_0^{-1}$，并将它们存储在字典中，这样给定任意（全局）颜色 C，我们就能在 O 时间内获取它在新编号中的索引。重新编号对于使用基础数据结构至关重要，因为它会返回紧凑表示。请注意，高频颜色集合是实际重频项的超集。事实上，我们为低层构建的基础结构可以输出一个包含 $\frac{w^\beta}{c_1\varepsilon_0}$ 种颜色的频率向量（以紧凑表示存储），因此我们无法逐一检查所有这些颜色。

我们使用一个辅助结构，为每个查询生成一个包含 $O(\frac{1}{\varepsilon_0})$ 个候选颜色的列表；正如我们将要证明的，只需检查这些候选颜色即可。我们称之为测试结构。这同样通过浅切割来实现：我们为层级 $\varepsilon_0^{-1}2^i$ 、$i=1$、，2，$\cdots$，$\log(w^{\beta\lambda})$ 构建浅切割。我们称这些为测试切割。对于测试切割中的每个单元 Δ，我们收集在 Δ 的冲突列表 $\mathcal{S}_\Delta$ 中至少出现 $c_1\varepsilon_0|\mathcal{S}_\Delta|$ 次的颜色列表。我们称这些颜色为候选颜色；显然，对于每个单元 Δ，我们存储的候选颜色最多为 $\frac{1}{c_1\varepsilon_0}$ 个。存储候选颜色所需的总空间（对所有测试切割求和）至多为

$$\sum_{i=1}^{\infty} O\left(\frac{n}{\varepsilon_0^{-1}2^i}\cdot\frac{1}{c_1\varepsilon_0}\right)=O(n).$$

查询算法。回想一下，当 k^* 介于 ε_0^{-1}和 $\varepsilon_0^{-1}w^{\beta\lambda}$之间时，查询会探查低层切割。在这种情况下，我们找到满足 $\varepsilon_0^{-1}w^{\beta i}$ 超过k^*的最小索引 i。我们找到包含该查询的 $\mathcal{C}_i$ 中的单元 Δ。因此可知 $|\mathcal{S}_\Delta|=O(w^\beta k)$。我们使用 q查询在$\mathcal{S}_\Delta$ 上实现的基础数据结构。然而，由于我们构建基础结构时使用的误差参数为 ε_0/w^β，基础数据结构可能会返回一个包含 $O(\varepsilon_0^{-1}w^\beta)$ 个颜色的列表，且该列表可能采用紧凑表示。我们选择足够小的 β，使基础数据结构的结果能放入 ε_0^{-1} 个字中。另一方面，由于$|\mathcal{S}_\Delta| = O(\varepsilon_0^{-1}w^{\beta\lambda}) = O(\varepsilon_0^{3/2})$，根据定理 9，基础数据结构的查询时间为 $O(\varepsilon_0^{-1})$。此外，我们还查询测试结构，以获得包含 $O(\varepsilon_0^{-1})$ 个候选颜色的列表。对于每个候选颜色，我们可以使用存储的字典在紧凑表示中查找其频率；如果存在，则通过提取操作将其报告出来。至此，查询算法结束。

正确性。基础数据结构会找出所有在 q 以下至少出现 $\frac{c_1\varepsilon_0|\mathcal{S}_\Delta|}{w^\beta}=\frac{O(c_1\varepsilon_0 k w^\beta)}{w^\beta}\le\varepsilon_0 k$ 次的颜色，只要将 c_1 选为足够小的常数。因此，任何出现 $\varepsilon_0 k$ 次的颜色都会由基础数据结构正确报告。

空间分析。根据定理 9，低层浅切割中的每一层将占用 $\mathcal{C}_i$ $O(n)$ 空间。共有 λ 个低层浅切割，因此低层切割所用的总空间为 $O(n\lambda)$。

对于半空间范围，回想一下，我们需要选择 λ，使得 $w^{\beta\lambda}=\varepsilon_0^{-1/2}$。由于 β 被选为较小的值，因此对于半空间，我们可以选择 $\lambda=O(\log_w\frac{1}{\varepsilon_0})$，从而得到 $O(n\log_w\frac{1}{\varepsilon_0})$ 的空间上界。

对于支配范围，只需选择 λ，使得 $w^{\beta\lambda}\ge\log^4\frac{1}{\varepsilon_0}$。在这种情况下，正如我们之前所示，通过将 β 选为足够小的常数，只需将 λ 选为足够大的常数，从而得到 $O(n)$ 的空间上界。

D.2.1.3 基础切割。

基础切割 $\mathcal{C}_0$ 是一个 ε_0^{-1}-浅切割。我们将单元的冲突列表显式存储在 $\mathcal{C}_0$ 中。根据引理 1，在这种情况下，我们能在时间 $O(\log n)$ 内找到 $\mathcal{C}_0$ 以及包含 q 的单元 $\Delta\in\mathcal{C}_0$。在这里，我们可以显式访问 $\mathcal{S}_\Delta$，并直接回答 AHHS 查询，所需时间为 $O(\log n+\varepsilon_0^{-1})$，且不会出错。

D.2.2 将它们整合起来

对于三个不同的浅切割层级，总空间复杂度为 $O(n\lambda)$。对于支配范围，我们看到可以将 λ 选为常数。由此得到一个空间复杂度为 $O(n)$ 的数据结构。对于半空间，我们选取了 $\lambda = O(\log_w \frac{1}{\varepsilon_0})$，因此得到一个空间复杂度为 $O(n\log_w \frac{1}{\varepsilon_0})$ 的数据结构。

D.3 二维 AHHS 查询

在平面中，我们实际上可以进一步降低空间复杂度。不过，这需要修改基础数据结构。这里的主要思路是，二维中 ε-近似的规模足够小，因此我们可以尝试更激进的近似。

D.3.1 二维 AHHS 查询的基础解决方案

本小节旨在证明二维专用基础数据结构。

▶ **定理 21**。设 P 中有一个彩色点集，并令 ε 为参数。设 $B = \varepsilon|P|$。我们可以将 P 存储在一个线性大小的数据结构中，使得给定一个包含 $F = \frac{1}{\varepsilon B^{1/3}}$ 种颜色的 q_c，可以在 $O(F)$ 时间内，将 q_c 中各颜色的频率估计到误差不超过 $\varepsilon|P|$。

主要思路和摘要。主要观察是，在二维中，给定点集 P和一个值 ε，存在大小为 $O(\varepsilon^{-4/3})$ 的 ε- 近似，因此，如果我们将它们存储在用于半空间范围计数查询的数据结构中，查询时间将降至$O(\varepsilon^{-2/3})$，即小于 ε^{-1}。直观地说，这意味着问题的困难情形是存在大量频率较低的不同颜色。然而，证明基础定理仍然相当不容易，因为我们的目标是实现非常快的查询时间。我们的主要观察是，当点较少时， ε-近似的大小足够小，因此我们可以承担略高的空间开销，并以此换取更低的查询时间。通过谨慎选择参数，我们得到了所声称的定理。

处理“高频”颜色。如果某种颜色出现至少$\varepsilon B|P|$ 次，我们就称其为高频颜色。显然，高频颜色的数量至多为 $|P|/(\varepsilon B|P|) = (\varepsilon B)^{-1}$。考虑一种高频颜色 j，并假设它在点集 P中出现了 n_j 次。我们为这种颜色存储一个ε_j-近似 $\mathcal{E}j$，其中 $\varepsilon_j = \frac{\varepsilon|P|}{n_j}$。$\mathcal{E}j$的大小为 $\min\left\{n_j, O\left(\frac{n_j}{\varepsilon|P|}\right)^{4/3}\right\}$，我们将$\mathcal{E}j$ 存储在定理 15 中的数据结构里。

现在考虑一个查询，并考虑如何处理 q_c 中的高频颜色。对于每种高频颜色 $j \in q_c$，我们只需统计$\mathcal{E}j$ 中的点数，并利用该值获得具有正确加性误差的估计。查询时间为 $O(\sqrt{|\mathcal{E}_j|}) = O\left(\frac{n_j}{\varepsilon|P|}\right)^{2/3}$。因此，处理所有高频颜色所花费的总查询时间为

$$\sum_{j\in q_c,\, j \text{ is frequent}} O\left(\frac{n_j}{\varepsilon|P|}\right)^{2/3} \le \frac{1}{\varepsilon B} O\left(\frac{\varepsilon|P|B}{\varepsilon|P|}\right)^{2/3} = O\left(\frac{1}{\varepsilon B^{1/3}}\right)$$

其中，不等式来自观察到 $\sum n_j \le |P|$，并且由于指数 $2/3$ 小于 1，函数 $x^{2/3}$ 是凹函数，因此当所有 n_j 都相等时，左侧表达式达到最大值。由于颜色数至多为 $(\varepsilon B)^{-1}$，因此当每个 $n_j = |P|/(\varepsilon B)^{-1} = \varepsilon B|P|$ 时，最大值达到。由此，我们可以在所声称的查询时间内处理高频颜色。空间显然也是线性的。

处理低频颜色。这是该数据结构中更棘手的部分。考虑一种低频颜色 j，它出现了 n_j 次。和之前一样，我们为这种颜色构建一个 ε_j-近似 $\mathcal{E}_j$，其中 $\varepsilon_j = \frac{\varepsilon|P|}{n_j}$，且 $\mathcal{E}_j$ 的大小为 $O\left(\frac{n_j}{\varepsilon|P|}\right)^{4/3}$。令 X 为所有低频颜色 j 对应的 $\mathcal{E}_j$ 的集合。我们可以渐近地将 X 的大小界定为

$$\sum_{j \text{ is infrequent}} \left(\frac{n_j}{\varepsilon|P|}\right)^{4/3} \leq \frac{1}{\varepsilon B}\left(\frac{\varepsilon B|P|}{\varepsilon|P|}\right)^{4/3} = \frac{B^{1/3}}{\varepsilon}$$

其中，不等式由詹森不等式得出，即函数 $x^{4/3}$ 是凸函数，因此左侧的表达式在每个 n_j 取最大值或最小值时达到最大值；然而，由于每种低频颜色最多出现 $\varepsilon B|P|$ 次，当 n_j中的 $\frac{1}{\varepsilon B}$ 个取 $\varepsilon B|P|$、其余取 0 时，最大值便可达到。

然后，我们利用点线对偶将 X 变换到对偶空间。我们在对偶空间中构造一个 $B^{-2/3}$-切割 Z。根据引理 16， Z 有 $O(B^{4/3})$ 个三角形覆盖整个平面，其中每个三角形与 $O(X/B^{2/3})$ 条 $= O(1/(\varepsilon B^{1/3}))$ 直线相交。对于每个三角形，我们存储从其下方经过的各颜色直线数量（$F = O(1/(\varepsilon B^{1/3}))$ 个频率），以及与其相交的直线集合（$O(1/(\varepsilon B^{1/3}))$ 条直线）。总计，我们将渐近地使用

$$\frac{1}{\varepsilon B^{1/3}} \cdot B^{4/3} = \frac{B}{\varepsilon} \leq |P|$$

空间。为了界定查询时间，考虑与查询半空间对偶的查询点 q。现在注意到，在切割中定位到包含 q 的三角形后，我们只需查看三角形中存储的频率，以及与三角形相交的直线集合。总共需要花费 $O(1/(\varepsilon B^{1/3}))$ 时间。

D.3.2 二维 AHHS 查询的数据结构

在针对二维查询的数据结构中，我们使用了为三维问题开发的大部分组成部分。特别地，我们可以使用 $O(n)$ 空间，使得对于给定的半平面 h，我们能在 $O(\log n + \varepsilon_0^{-1})$ 时间内找到一个包含 $O(\varepsilon_0^{-1})$ 个候选颜色的列表。类似地，我们还构建了一个层级 0 的浅切割 $\mathscr{C}_0$，使我们能够在 h 至多包含 ε_0^{-1} 个点时回答查询。不过，我们只构建了 D.2 小节中的第一个较低层级浅切割 $\mathscr{C}_1$。因此，如果 h 包含至多 $\varepsilon_0^{-1}w^{\beta}$ 个点，我们就能以最优运行时间回答查询。处理其余查询的方式则是我们与三维数据结构不同之处。

现在我们构建另一种浅切割层次结构。令 $g_0 = \varepsilon_0^{-1}w^{\beta}$。我们使用参数 g_0、g_1、$\cdots$构建一系列浅切割，这些参数稍后确定。具体而言，我们构建一个 g_i-浅切割 $\mathscr{C}_i'$，直到 g_i 超过 ε_0^{-3}；然后切换回三维解。如前所述，这部分数据结构的总空间为 $O(n)$。对于 $\mathscr{C}_i'$ 中的每个单元 Δ，我们在定理 21 的基础数据结构中存储 Δ 的冲突列表，参数如下：我们有 $|P| = \Theta(g_i)$，希望设置 ε，使查询时间等于 $O(1/\varepsilon_0)$，因此必须满足

$$\frac{1}{\varepsilon B^{1/3}} \leq \frac{1}{\varepsilon_0} \Leftrightarrow \varepsilon B^{1/3} \geq \varepsilon_0 \Leftrightarrow \varepsilon\left(\varepsilon|P|\right)^{1/3} \geq \varepsilon_0 \Leftarrow \varepsilon\left(\varepsilon g_i\right)^{1/3} \geq \varepsilon_0 \tag{10}$$

因此，按照满足不等式 10 的方式设置参数，可以确保所用基础数据结构的查询时间为 $O(\varepsilon_0^{-1})$。

为保证近似因子，我们注意到，已知查询位于浅切割 $\mathscr{C}_{i-1}'$ 之外，这意味着查询中至少有 g_{i-1} 个点。因此，

如果我们能够确保以下条件，那么我们的近似因子也将达到预期要求：

$$\varepsilon|P| \leq \varepsilon_0 g_{i-1} \Leftarrow \varepsilon\Theta(g_i) \leq \varepsilon_0 g_{i-1}. \tag{11}$$

不等式 10 和 11 给出了关于 g_i 的递归。为简化说明，我们可以直接重新缩放常数 ε_0，以去掉不等式 11 中 $\Theta(\cdot)$ 记号里的常数，并将不等式改写为等式。由此得到 $\varepsilon = \frac{\varepsilon_0 g_{i-1}}{g_i}$，进而得到

$$g_i \leq g_{i-1}\left(\varepsilon_0 g_{i-1}\right)^{\frac{1}{3}} \quad \text{and recall that we have } g_0 = \frac{w^\beta}{\varepsilon_0}. \tag{12}$$

现在，通过简单的归纳法可得

$$g_i = \frac{1}{\varepsilon_0} \cdot \left(w^\beta\right)^{(4/3)^i}. \tag{13}$$

回想一下，一旦 g_i 达到 ε_0^{-3}，我们就切换到三维解，从这时起，剩余层级将占用 $O(n)$ 空间。因此，该数据结构的空间复杂度为 $O(nj)$，其中 j 是满足 $g_j \geq \varepsilon_0^{-3}$ 的最小索引。根据方程 13，显然 $j = O(\log\log_w \varepsilon_0^{-1})$，因此空间上界为 $O(n\log\log_w \varepsilon_0^{-1})$。