arXiv:2603.10809v1 [cs.DB] 11 Mar 2026

超越标准数据立方体：从不规则且具有分支结构的地球系统数据中提取特征

MathildeLeuridan James Hawkes Tiago Quintino Martin Schultz

2026年3月12日

摘要

地球科学数据集的规模和结构复杂性都在迅速增长。其中包含的丰富标记数据越来越多，这些数据具有异构元数据和复杂的内部约束，会在变量与维度之间形成依赖关系。数据立方体已成为组织此类数据集的常用抽象，但传统的稠密正交数据立方体模型难以高效表示不规则、稀疏或具有分支结构的数据空间。尤其是覆盖空缺、条件维度和异构变量定义，现有方法都难以妥善表述。本文提出一种基于压缩树结构的广义数据超立方体表示，可精确且紧凑地描述复杂数据空间。我们介绍了该表示的设计，并分析其在保持遍历效率的同时捕捉稀疏性和条件关系的能力。通过具体实现，我们研究了压缩树数据超立方体的性能特征，并证明其可作为大型后端数据存储之上的快速、类似缓存的索引。在此表示的基础上，我们提出一个集成式特征提取系统，可在 Polytope 框架中直接对基于树的数据超立方体进行操作。通过将数据访问策略嵌入数据超立方体抽象本身，该系统能够实现精确的子域数据提取，并支持灵活、由用户驱动的访问模式。我们评估了该集成系统的性能，并展示它如何为复杂数据集带来新的交互方式，而传统访问模型难以支持这些方式。本研究弥合了表达能力强的数据超立方体模型与高效数据访问方法之间的差距。具体而言，它提供了一个统一框架，将基于树的数据表示与特征提取能力相结合。因此，所提出的方法为大规模异构地球科学数据集提供了可扩展、以用户为中心的访问基础，并指出了未来开发与部署的方向。

1 引言

地球观测、气候建模和数值天气预报领域的进步，带来了前所未有规模的地理空间和气象数据 [12, 39, 18]。现代卫星在数百个光谱通道和仪器上持续生成观测数据，而天气和气候模型则生成日益详尽的预报，涵盖多个集合成员、参数化方案和输出频率。以 netCDF 数据模型 [41] 以及气候与预报（CF） [8] 和气候模型输出重写器（CMOR） [35]元数据规范为代表的基础标准化工作，使得能够高效表示和交换主要沿着熟悉的纬度、经度、垂直层次和时间维度组织的数据集。然而，现代数据集日益

超出这一标准模型，增加了高度层、仪器模式或预报提前期等额外轴，从而形成结构丰富但高度复杂的数据空间。

尽管如此丰富的数据带来了科学机遇，但也造成了重大的实际挑战 [47, 19]。事实上，高维数据集往往难以可视化、分析和操作。在低维环境中轻而易举的分析操作，例如切片提取子集、组合变量或在坐标系之间进行映射，一旦数据涉及不规则网格、包含条件依赖关系，或因仪器几何结构或质量控制而存在空缺，就会很快变得繁琐。除了表示方式本身的问题外，这些困难还会直接影响从数据中筛选、查询和提取有意义的子集或特征的能力，而这些都是大多数科学工作流中的关键步骤。因此，人们愈发需要一种计算框架，既能统一表示此类数据集，又能同时实现高效、直观的数据访问。

应对这一需求的一个重要举措是发展数据立方体抽象。数据立方体概念最初源自商业分析界，提供了一种对多维表格数据建模的方法，并支持切片和透视等操作[17, 20, 36]。过去十年间，地球科学界重新诠释并拓展了这一理念，使其能够适用于栅格和观测数据集 [25, 31, 16, 15]。RasDaMan [4, 3] 以及欧洲航天局的深地球系统数据实验室（DeepESDL）[2, 6] 等项目尤其表明，数据立方体能够统一异构数据集，并简化复杂档案的访问。然而，大多数实现主要侧重于组织并呈现同质多维数据的结构和元数据，而更高层次的数据筛选和特征提取往往被视为下游的临时处理过程。

在开源科学生态系统中，xarray 库 [22] 通过为多维数组提供带标签的张量表示，进一步推广了数据立方体的概念。在气象学、气候研究和遥感领域，xarray 已成为数据分析工作流的核心工具，因为它能简化对复杂数据集执行的基本操作。然而，xarray 所依托的张量模型包含一些隐含假设。其中之一是，数据位于正交且间隔规则的坐标轴上，并构成没有结构性缺口的稠密数组。现代数据集可能是稀疏、不规则的，或仅针对维度的特定组合定义，例如变量依赖于特定仪器或集合配置；这使得这些假设日益难以成立。在这种情况下，即使是标准的筛选或提取任务，也可能需要大量预处理，或需要在数据立方体抽象之外编写专门逻辑。

一种可能的解决方案是将数据拆分为多个低维数据立方体。例如，Deep-ESDL 将每个地球物理参数存储为独立的 3 维或 4 维数据立方体。不过，尽管这种表示方式实用，但这种碎片化会掩盖参数之间的关系，妨碍从整体上统一查看数据集。更重要的是，这会使通用数据提取工作流的实现更加复杂；这类工作流必须能够在不同参数、维度和条件依赖关系之间保持一致。认识到这些局限后，xarray 最近推出了 DataTrees [49, 37], ，这是一种层次化扩展，旨在以树状结构表示相关的数据立方体。DataTrees 通过支持分支和分组提供了更高的灵活性，但仍要求结构具有均匀性。事实上，各分支必须具有相似的维度假设、坐标顺序和整体数据层次结构，这限制了它们支持在高度不规则的数据空间中进行通用筛选和特征提取的能力。

本文探讨一种基于灵活层次树结构的高维数据空间通用表示方法，旨在捕捉复杂、异质且不规则数据集的坐标结构，突破现有数据立方体框架的约束。该方法将维度依赖关系和结构规则直接编码到树中，

放宽了对正则性、完备性和均匀性的假设，同时支持对稀疏或非正交数据空间进行高效查询、合并、筛选和遍历。关键的是，这种表示不仅用于描述数据，还旨在作为端到端数据提取工作流的基础。

随后，我们简要介绍 Qube [10]，将其作为这种压缩树状数据立方体表示的具体实现，并研究其性能及其在实际应用中的潜在用途。特别是，我们展示了 Qube 在 Polytope 特征提取框架 [29, 28], 中的原型集成，说明如何直接基于通用数据立方体表示实现特征提取。这实现了一个完整的工作流程，使数据表示、筛选和特征提取紧密集成，同时不对底层数据的结构施加限制性假设。最后，我们讨论这一集成框架，强调采用更通用的数据立方体方法所带来的益处，以及其在各种数据提取和分析系统中的潜在适用性。

2 动机与背景

近年来，大量研究致力于形式化数据立方体和多维数据空间的概念，以更好地适应日益复杂的地理空间数据集。本节阐述对这类抽象概念的需求，并回顾当前的实现，同时考察多维数据的表示方式，以及数据在实践中的访问和使用方式。我们讨论高维科学数据集为何需要更加系统化的表示，并考察塑造当前方法的既有框架，贯穿全文的示例均取自气象与气候数据。特别是，我们强调多仪器观测、集合预报、异构网格和条件性数据可用性等特征如何对传统的数据立方体假设构成挑战。我们还回顾现有框架所采用的数据访问策略，包括查询、子集提取和筛选机制，并考察这些选择如何影响天气与气候应用的可用性和性能。综合来看，这些因素凸显了对更加灵活、统一的数据立方体结构的需求，使其既支持富有表现力的数据表示，又能实现高效的数据访问。

2.1 数据立方体抽象与数据表示

数据立方体概念最初在商业分析领域被提出，用于组织并高效查询多维表格数据库 [20]。早期的数据立方体模型支持沿产品、区域和时间等不同轴进行聚合、切片和透视等操作，为结构化多维数据提供了统一抽象 [17]。随着数据密集型科学学科的扩展，这一抽象被证明在其原始领域之外也具有广泛适用性。许多领域，包括生物医学成像、基因组学、环境监测、社会科学和大规模传感器网络，经常生成可沿多个坐标维度描述的数据集 [26, 24, 51]。随着时间推移，数据立方体因此演变为一种被广泛采用的概念模型，用于以结构化和可解释的形式管理复杂的高维数据集。

这一演进体现在 xarray [22] 等科学软件框架，以及 Zarr [50], 等存储格式中：它们将数据集表示为带标签张量或具有命名坐标的多维数组。这些基于张量的模型已成为许多数据驱动工作流的基础，因为它们支持直观的索引、高效的数值计算，以及与高性能计算和机器学习生态系统的集成 [11, 45, 27]。对于密集、均匀且定义在正交坐标轴上的数据集，例如定期

网格化模拟、标准化成像数据或同质观测记录，这些表示方式兼具清晰的概念结构与实用效率。

然而，许多现实世界数据集都与这些假设存在显著偏离。在不同领域中，数据可能是稀疏的、异质的，或仅针对变量的特定组合而定义。在医学领域，患者记录通常包含缺失值、特定于病情的测量结果，或长度不一的记录。基因组数据集表现出不规则覆盖和比对特征。传感器网络数据可能是间歇性的，或受到设备特定配置的约束。尤其是气象与地球系统数据集，往往会同时呈现其中的诸多挑战。气候模拟通常会生成单条长时间序列，而天气预报则同时引入初始化时间和预报提前期，从而形成多个时间轴。地球观测产品还会增加与仪器模式、轨道几何或观测配置相关的其他维度。因此，单一的稠密正交张量无法忠实地表示组合后的数据空间。

在气象与气候数据集中，还会出现进一步的结构变异性。不同变量可能在不同的预报提前期可用，垂直坐标可能定义在模式层、气压层上，或仅定义于地表，而观测数据可能仅在特定质量约束或传感器配置下存在。这会形成分支数据空间，其中数据集不同子集的维度、坐标定义和数据可用性各不相同。尽管许多子空间具有共同的结构模式，但整体数据空间会变得不规则且稀疏，并包含难以用传统张量模型表达的条件依赖关系。

为了解决其中的一些局限，xarray 引入了 DataTrees [49, 37], 这一层次化抽象，将相关数据立方体组织成树结构。这样一来，数据集组可以共享共同的维度，同时通过分支表示结构各异的子集。基于树的数据立方体模型是捕捉关系型及条件定义的数据空间的重要一步；对于组织由多个相关产品组成的复杂地球系统数据集，它们尤其适用。

尽管如此，现有的层次化数据立方体模型仍施加了显著的结构约束。它们通常假定轴顺序固定、同级分支的维度相同，并对何为有效子空间作出僵化的界定。当数据集的维度本身随域而变化、不同分支需要不同的坐标集，或连续施加的约束动态改变相关维度时，这些模型便难以表示此类数据。在这种情况下，若不对数据进行碎片化处理或重复，就无法自然地表达其结构。这些局限表明，我们需要更通用、灵活的表示方式，以便在保留共享结构并支持高效导航的同时，编码异构、稀疏且结构可变的数据空间。

2.2 数据立方体访问与数据提取

除纯粹的表示之外，处理多维数据集时同样关键的一个方面，是如何从底层数据存储中访问并提取数据。在实践中，用户很少需要完整的数据立方体。相反，他们寻求对应于感兴趣区域、时间窗口、垂直层次、集合成员或条件约束组合的特定子集。因此，有效的数据立方体框架必须支持数据访问策略，使用户能够既高效又直观地仅检索与其科学问题相关的信息。传统的数据立方体框架通过沿坐标轴进行切片、索引和聚合等操作来支持访问 [17, 46]。在基于张量的模型中，这些操作往往与数组索引和分块策略紧密耦合 [38, 32], ，后者对规则且稠密的数据集效果良好。然而，随着数据空间变得更加不规则并具有条件结构，这些访问模式越来越难以泛化。查询稀疏或分支

数据通常需要自定义逻辑、预处理或对多个数据立方体进行手动遍历，从而降低可用性并增加数据访问的计算成本。
为应对这些挑战，特征提取作为一种更高层次的数据访问策略，在气象学和气候科学领域日益受到重视。特征提取框架不再直接暴露原始多维数组，而是允许用户直接根据科学特征指定其感兴趣的数据子集，例如轨迹、剖面、垂直廓线或空间区域。这一转变体现在诸如 OGC环境数据检索（EDR）标准等接口的发展中 [5], ，该标准使用户能够通过统一API从复杂的地理空间数据集中请求特定特征。

在机构层面，类似理念也已体现在欧洲中期天气预报中心的 Polytope 特征提取框架等系统中 [29, 28], ，该框架允许用户针对高维天气与气候数据集描述复杂的提取请求。基于 RasDaMan [4, 3] 或 Sentinel Hub [34], 等平台构建的数据服务中也存在相关方法，用户可以通过更高层次的裁剪和子集提取操作请求空间或时间子集。这些发展体现了特征提取正逐渐成为与现代数据立方体和数据系统交互的一种重要方式。

然而，许多现有的数据立方体实现将特征提取视为数据已从后端存储中检索出来之后才应用的后处理步骤。在此类系统中，大量数据会先被访问并传输，然后再经过裁剪或后处理，以生成最终的用户输出。虽然这种方法对于较小的数据集通常已经足够，但对于天气和气候数据而言却越来越低效，因为后端存档可达拍字节规模，而用户请求通常只针对几兆字节的信息。在此类应用中，访问不必要数据的成本主导了工作流，从而限制了可扩展性和性能。

数据立方体抽象与高效数据访问策略之间的这一差距表明，需要新的数据立方体框架。在此类框架中，特征提取应被视为一项关键的数据立方体操作，并与底层数据模型紧密集成。访问策略应能够利用数据结构本身的知识，从而仅对相关子空间进行选择性遍历和检索。以这种方式弥合表示与基于特征的数据访问之间的鸿沟，对于在气象学、气候科学及其他数据密集型学科中实现可扩展且用户友好的工作流至关重要。

3 数据立方体的泛化

传统的数据立方体模型假设数据占据一个稠密、正交且完全定义的多维空间。正如前文所示，对于现代科学数据集而言，这些假设正日益不再成立，尤其是在气象学和气候科学领域：数据可用性取决于条件约束，维度会因子集而异，而且空间中的很大部分可能稀疏或未定义。将此类数据集表示为完整的数据立方体，要么会导致其被拆分为多个彼此不连通的数据立方体，要么需要用缺失值进行低效填充。

为了解决这些局限性，我们提出对数据立方体概念进行广义化，放宽对完备性和正交性的全局假设。我们不再将数据立方体视为单个多维数组，而是将数据空间表示为一种压缩的层级结构，明确编码分支和条件关系。这种表示称为数据超立方体，能够在可能的情况下保留共享结构，同时允许在必要时出现分化。因此，它为高效查询和面向特征的数据访问提供了自然的基础。在本节余下部分，我们将介绍这种广义数据立方体表示，形式化其结构，并讨论它在以下方面的优势

以提升气象数据系统之间的表达能力、性能和复用性。

3.1 数据超立方体

广义数据超立方体将多维数据空间表示为压缩树结构，而非稠密数据立方体。树的每一层对应一个维度，分支则表示由数据约束造成的条件结构。从根节点到叶节点的路径定义了数据实际存在的有效坐标组合。

实际上，许多地球物理数据集都具有维度间的约束，这意味着并非所有坐标组合都有意义或包含数据。例如，考虑一个包含一组大气变量的气象数据集。有些变量（如2米温度或地面气压）仅在地表定义，而另一些变量（如温度、风或湿度）则定义在多个气压层上。因此，垂直维度取决于变量；若将所有变量都视为共享同一组垂直坐标，就必须为仅在地表定义的变量引入虚拟层，或者将数据集拆分成多个相互独立的数据立方体，从而掩盖变量之间的关系。相比之下，广义数据超立方体树的形式化表示既能高效地表示稠密、正交的子数据立方体，也能适应稀疏、异构且不规则的数据空间。

形式上，数据超立方体可以定义为一棵有根有向树

$$\mathcal{T} = (V, E),$$

其中，每个节点 $v \in V$ 都与一个维度 $d(v)$ 以及一组可接受坐标值 $C_v \subseteq \mathcal{C}_{d(v)}$相关联。边 $(v_i, v_j) \in E$ 表示通过对后续维度施加额外约束来进行细化。因此，每条根到叶路径都定义了整个域中的一个有效子空间，而叶节点随后可以关联有效载荷，例如数据数组、存储引用或各种元数据。在此背景下，完全稠密的数据立方体对应于扁平树，其中每个维度层级都穷举列出了其坐标域。
在广义表示中，上述示例中的结构变异性通过超立方体树中的分支来表达。树的较高层级可以编码共享维度，例如预报提前期，而后续分支则区分地面变量与气压层变量。一条分支在地表处终止，另一条则引入与气压层对应的额外垂直维度。因此，每个叶节点都代表一个具有明确定义维度结构的内部一致子空间。共享维度只表示一次，仅在数据结构确实不同时才引入分歧。

与经典数据立方体不同，这种表示方式在维度置换下并非不变。维度沿树出现的顺序既定义了数据空间的逻辑结构，也决定了用于查询的最佳遍历顺序。树中位置较高的维度充当早期过滤器，可在某些子空间与特定请求无关时将其整体剪枝。因此，选择合适的维度排序是一项重要的设计决策，可以根据常见的访问模式进行优化。在气象和气候工作流中，可以将不同卫星仪器或气象模型类型等较高层级的区分放在树的顶部，以便在遍历早期排除与不兼容的采集系统或建模框架相对应的分支。沿这些区分划分数据空间后，每个由此形成的子超立方体中的其余维度通常会更加同质，从而提高各分支内部查询和组织的效率。

该数据　超立方体表示为建模复杂的 x

气象数据空间提供了一种紧凑且富有表现力的方式。通过显式编码结构依赖关系，它既能实现高效遍历，也能进行稳健筛选，为更高级的数据提取和分析流水线奠定基础。

此类基于树的数据立方体表示已在欧洲中期天气预报中心的不同业务场景中得到采用。它们支撑着 MARS档案库中数据可用性的内部表示 [40] ，也通过 Qubed软件 [10]用作 Destination Earth STAC目录 [1] 的后端。在这些系统中，数据超立方体已展现出高效编码大型异构气象数据存储库的能力，同时支持可扩展的数据发现与访问。

3.2 数据超立方体操作

前文介绍的数据超立方体表示建立在压缩树结构之上，支持一组定义明确的操作。这些操作对于高效构建数据超立方体，以及在底层数据随时间演变时保持其最优性，都至关重要。在实践中，数据超立方体树必须逐步构建，在新数据可用时重新计算，并针对多样化的用户查询进行高效遍历。其中一个关键部分就是对树结构本身执行的操作。基本操作包括遍历、沿维度进行筛选和切片，从而实现对数据空间的选择性探索。

从构造的角度来看，考虑一个定义在一组维度$D = \{d_1, \ldots, d_k\}$上的数据超立方体，其中每个维度 d_i 的基数为 n_i。在实际中，生成的树表示不一定是均匀的。事实上，由于稀疏性或缺失坐标组合，树的不同分支可能在不同的树深度终止。令 $k_{\max}$ 表示树中任意根到叶路径的最大深度。朴素的构建过程会独立插入每个可用的数据元组，为其坐标在树中创建一条对应路径。如果底层数据集包含 N 个这样的元组，这意味着 N 是叶节点的数量，那么每次插入最多需要遍历或创建一条长度为$k_{\max}$的路径。因此，总构建成本的上限为

$$T^{\text{naive}}_{\text{construct}} = \mathcal{O}(k_{\max} N).$$

类似地，这种表示中的提取或遍历操作可能需要访问很大比例的 N 个叶节点；而在两个分别包含 N_1 个和 N_2 个叶节点的数据超立方体之间进行并集或交集等集合运算，则需要

$$T^{\text{naive}}_{\cup}, T^{\text{naive}}_{\cap} = \mathcal{O}(N_1 + N_2).$$

要组合多个数据集、协调重叠的数据空间，或将表示限制为维度和坐标的共享子集，就需要更高级的集合运算，例如树的并集和交集。这些运算使我们能够从较简单的数据空间逐步构建复杂的数据空间。在整合异构数据集或汇总一段时间内的数据时，并集操作发挥着核心作用；例如，将连续的预报周期或观测产品添加到现有数据存储中。在这种情况下，必须将多个数据超立方体树合并为单一表示，既保留共享结构，又仅在数据存在差异之处引入分支。相比之下，交集操作旨在将数据超立方体限制在一个公共子空间中，例如识别数据集之间重叠的可用性，或应用根据用户查询得出的约束。

为确保效率，这些集合运算必须伴随树结构的压缩与重新优化。在并集或交集之后，所得的树可能包含冗余分支或未压缩的重复子树，从而降低遍历性能。令压缩将结构相同的子树折叠，从而使具有 N 节点的树缩减为一个

仅包含

$$M \ll N$$

不同的结构节点。
压缩操作本身包括识别结构相同的子树，并用共享表示替换它们。这需要比较树中跨越相同坐标子空间的分支，以检测子超立方体之间的相似性。在最坏情况下，必须对未压缩表示中的所有节点进行比较，因此压缩步骤的复杂度与节点数量呈线性关系，

$$T_{\text{compress}} = \mathcal{O}(N).$$

然而在实践中，压缩通常采用自底向上的方式进行，从树的叶节点开始逐层向上推进。由于相同的子树更可能出现在树的叶节点附近，以这种方式进行压缩可以减少在树的更高层需要比较的不同中间节点数量。因此，在较深层最大限度提高冗余度的树层次结构，不仅能减少最终的结构节点数量M，还会降低压缩过程本身的实际成本。

压缩后，后续的构造和变换操作便可以基于这些唯一的子超立方体，而非整个笛卡尔空间来执行，从而得到

$$T_{\text{construct}}^{\text{compressed}} = \mathcal{O}(k_{\max} M), \qquad T_{\cup}^{\text{compressed}}, T_{\cap}^{\text{compressed}} = \mathcal{O}(M_1 + M_2).$$

因此，压缩数据超立方体之间的并集或交集不再需要遍历全部 N 个节点，而只需遍历 M 个不同的结构节点，从而使计算代价按与 $(N_1 + N_2)/(M_1 + M_2) \gg 1$ 成比例的因子降低。

因此，压缩是一项必不可少的操作，因为它会折叠相同的子树并提取共享的树结构，从而在并集、交集或任何其他变换之后恢复高效的数据超立方体压缩树对象。因此，压缩是数据超立方体构建过程中不可或缺的一部分。它确保所得表示保持紧凑，并与典型访问模式良好对齐。它还表明，可实现的效率直接取决于维度在树层次结构中的排列方式，因为能够最大化共享子超立方体的维度排序会最小化 M，而沿高度可变维度的早期分支则会增大 M，并降低构造、提取和集合运算的性能。

3.3 数据超立方体性能

数据超立方体表示的性能是任何实际数据工作流中的核心问题。构造、压缩、并集和交集等操作，是构建、更新和维护复杂数据空间的一致表示的基础步骤。由于这些操作会在数据超立方体的整个生命周期中反复执行，其计算特性会直接影响下游应用和数据访问的可扩展性与性能。因此，了解如何高效执行这些操作，对于评估此类广义数据超立方体框架的实际用途至关重要。在运行环境和时间要求严格的场景中，性能考量尤为重要，因为数据空间会持续演变，必须以尽可能低的延迟生成并遍历最新表示。在此类情境下，超立方体构建或遍历中的低效很快就会成为系统总体成本的主要部分，限制系统及时、一致地提供数据产品的能力。

3.3.1 数据工作流

接下来，我们以 Qubes [10] 作为压缩树形数据超立方体表示的一种具体实现，研究其性能。尽管分析聚焦于 Qubes，但研究结果也适用于更广泛的一类广义数据超立方体框架；这类框架使用分层压缩树来编码结构、稀疏性和条件关系。

在本文讨论的工作流中，Qube 主要充当大型底层数据存储之上的快速索引和缓存。因此，其有效性不仅取决于表达能力，也取决于底层数据发生变化时，能否高效地构建、维护和重用。由此，了解 Qube 核心操作的性能表现，是评估其是否适用于实际运行的关键。

我们通过考察 Qube 在典型数据工作流中的生命周期来分析其性能。首先，我们研究 Qube 构建的成本，因为这是进入该数据结构的起点。随后，我们考察压缩，重点关注其计算特性以及它在大规模场景下维持效率方面的作用。压缩带来了固有的权衡：虽然会增加构建成本，但能显著降低后续遍历、查询和集合运算的成本，而这些操作往往在典型使用模式中占据主导。接着，分析扩展到更复杂的操作，例如并集及其他集合运算；在合并数据空间或增量更新数据空间时，这些操作必不可少。最后，我们通过具体的计时结果，展示在 Destination Earth计划 [48] 的背景下，天气预报和气候科学中大规模地球系统数据超立方体的构建与操作情况。

3.3.2 Qube 构建

Qube 的构造时间主要取决于树大小以及树中发生分支的位置。在图 1(a) 中，我们首先通过构造一棵仅在叶节点处分支的扁平树并改变其数量，来研究叶节点数量的影响。正如预期，构造时间随叶节点数量线性增加，因为这种设置需要构建数量线性增长的节点，而这是主要成本。

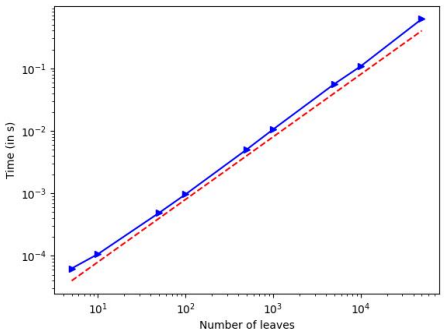
在图 1(b) 中，我们接着通过比较在不同深度处分支的树，考察分支层级如何影响构造时间。越靠近根节点的分支会导致更高的构造时间，而靠近叶节点的分支则更快。这种时间的线性下降是预期之中的，因为更早的分支总体上会产生更多节点。

最后，图 1(c) 在保持平衡树的同时，通过改变每个深度的节点数量，将这两种效应结合起来。构造时间再次随节点总数线性增长。然而，由于节点数量随每层分支因子呈指数增长，即使分支的小幅增加也会导致巨大的构造成本。重要的是，对于稠密 Qube，这一成本在很大程度上得以避免：原生压缩将有效叶节点数量从 n 降至 1，从而大幅减轻高度分支树的开销。

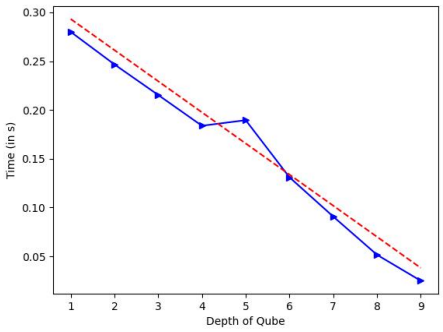
3.3.3 压缩

压缩是 Qube 的核心特性，可实现对数据的高效表示。因此，我们评估对初始未压缩的 Qube 进行压缩的开销。实验设置与上一节相同，但此处测量的是压缩时间，而非构造时间。

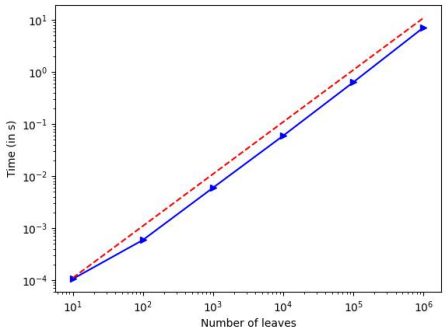
观察到的趋势与构造过程的趋势非常相似。在图 2(a)中，压缩代价随叶节点数量线性增长，这反映了处理数量呈线性增长的分支所需的工作量。在图 2(b)中，我们进一步观察到，树中较高位置的分支



(a) 扁平 Qube 的叶节点数量不断增加。



(b) 在不同 Qube 深度处分支。



(c) 宽 Qube 的叶节点数量不断增加。

图 1：构建不同类型 Qube 所需的时间。虚线表示这些计时结果对应的预期线性行为。

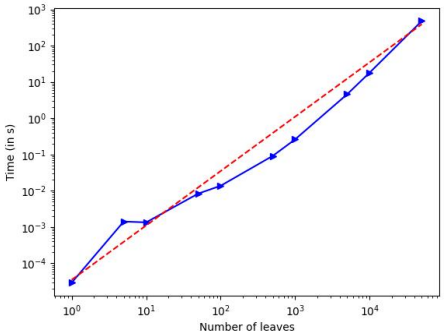
会降低压缩效率，因为检测并合并等价子树需要更深层的递归。尽管如此，这一代价仍呈线性增长；随着分支向叶节点移动，所需的嵌套比较会减少，从而使压缩成本逐步降低。

图 2(c) 综合了这些影响，再次显示压缩时间与 Qube 中叶节点总数之间呈线性关系。这种行为源于对嵌套分支的线性遍历，以及对等价子树的顺序合并。值得注意的是，对于已经较为密集或部分压缩的 Qube，压缩速度显著更快，因为需要比较和合并的分支更少。

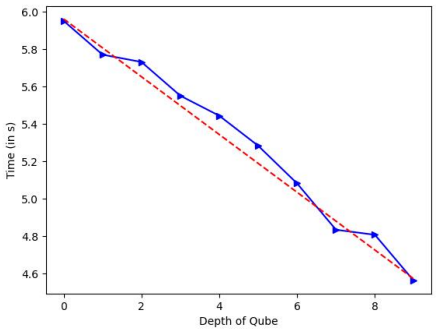
在实践中，压缩只需执行一次。获得最大压缩Qube 后，后续操作便可受益于这种最优表示，从而高效得多。此外，压缩的绝对代价仍然适中；在实际场景中通常还会更低，因为 Qube 往往越接近叶节点越密集，而图 2(b) 所示，压缩恰恰在这些位置最为高效。

3.3.4 集合运算

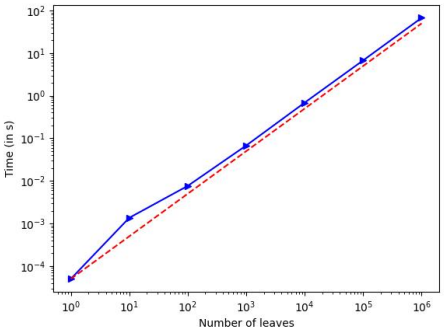
构建 Qube 后，最受关注的主要操作是集合运算。由于这些操作具有相似的性能特征，我们以并运算作为



(a) 扁平 Qube 的叶节点数量递增。



(b) 在不同 Qube 深度处分支。



(c) 宽Qube 的叶节点数量递增且分支因子较高。

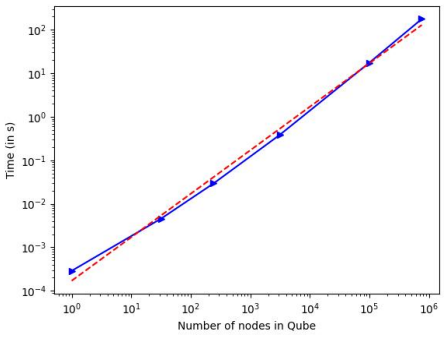
图 2：压缩不同类型 Qube 所需的时间。虚线表示这些计时结果对应的预期线性行为。

一个代表性案例。在图 3(a)中，我们首先比较完全展开的树与高度压缩的 Qube 执行并集运算的开销。结果表明，并集所需时间随每层树的节点数线性增长，这证实了理论界限：集合运算的规模随输入 Qube 中节点总数线性增长。

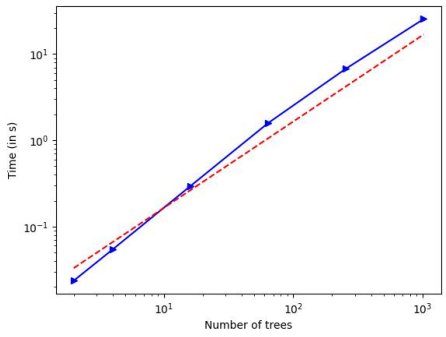
在图 3(b) 中，我们接着研究依次对多个固定大小的 Qube 执行并集运算时，性能如何扩展。总成本同样呈线性增长，因为每次并集运算都是依次应用于上一次运算的结果。这种模式在实践中很常见，尤其是在由多个稠密子 Qube 构建复杂 Qube 时。虽然线性扩展符合预期，但这也表明仍有优化空间。事实上，先对更相似的 Qube 执行并集运算，或采用可批量处理操作的其他合并策略，都可以降低更复杂工作流中的总体成本。

3.3.5 实际考虑事项

在实践中，为现实世界数据集构造 Qube，需要结合上述讨论的多种操作，包括增量构建、压缩和合并。因此，整体性能不仅取决于数据规模，还取决于其结构特



(a) 两个 Qube 的并集，其中每个 Qube 的总节点数逐渐增加。



(b) 数量逐渐增加的中等大小 Qube 的并集。

图3：将 Qube 合并为并集所需的时间。虚线表示这些计时结果对应的预期线性行为。

性以及操作的应用顺序。虽然朴素的构造和合并操作通常会随条目数量线性增长，但对于包含数百万个元素的数据集，这种方式很快就会变得不切实际，因此必须采用更周全的构造策略。

一项关键的优化策略是尽可能在构造过程的早期利用子超立方体之间的结构相似性。先压缩较小的 Qube，再将其合并到更大的结构中，可以及早折叠相同或近乎相同的子树。这不仅能减少内存使用量，还能降低后续操作的成本。早期压缩还能限制中间表示的增长，并避免对冗余结构重复处理。

另一种有效的方法是将合并操作批量执行，优先组合较小、成本较低的 Qube。这样可以推迟对大型树执行成本较高的并集操作，直到中间结果中已发现并压缩了足够多的结构。同样，也可以在将生成的分支连接到更高层级的树节点之前，对 Qube 的较低层级进行压缩和合并。只有当数据结构开始出现分化时才执行成本较高的并集操作，可确保这些变换仅作用于树中较浅层的子集，从而提高整体效率。

这些策略已应用于 Destination Earth 数字孪生中，Qube 在其中充当快速的类缓存索引，为交互式目录探索提供支持。为构建这些索引，我们扫描欧洲中期天气预报中心 FDB数据存储所公开的扁平元数据索引 [43, 44]，并结合增量构建、压缩和合并，构建能够反映现有数据结构的 Qube。对于包含约 860 万条数据记录的气候数字孪生 [21, 7], , Qube 构建大约需要一天。相比之下，规模较小的极端事件数字孪生 [13] 每天大约一小时即可重新生成。

尽管进行了这些优化，Qube 构建在计算上仍然代价高昂，应被视为一项高成本的索引步骤，而非轻量级操作。不过，构建完成后，Qube 能够紧凑地表示数据结构，并支持极快的选择与遍历操作。因此，与直接访问底层数据存储相比，Qube 作为查询接口非常高效。总体而言，Qube 最适合被视为更新较慢、但速度很快的索引缓存。它们生成成本高昂，但在.....中使用和重复使用时效率极高

交互式和业务工作流。

4 集成式数据提取系统

上一节介绍的基于树的数据超立方体范式，为复杂、稀疏且不规则的数据空间提供了一种紧凑而富有表现力的表示方式。我们讨论了如何将结构约束和条件性数据可用性直接编码到数据表示本身中，并以 Qube 作为这一范式的具体实例，展示了如何高效地构造和重复使用这种表示。

本节基于上述基础，介绍一个完整的特征提取系统，该系统将感知结构的数据超立方体表示直接集成到业务数据访问管道中。该系统不再将数据超立方体视为被动的数据容器，而是将数据结构、索引和访问策略整合到一个连贯的工作流中。这样，特征提取算法便可以直接根据数据超立方体表示来判断数据的可用性和布局，而无需依赖静态且由外部维护的数据规则。

本节安排如下。首先，我们介绍整体系统架构，以及 Polytope [30], Qubed [10] 与 GribJump [9]软件之间的交互，并重点说明感知结构的数据超立方体表示如何融入提取工作流。随后，我们分析系统的性能特征，尤其关注输入/输出效率，以及基于树的表示如何影响访问模式。最后，我们讨论这种方法对用户交互的影响，说明它如何支持更丰富的新型访问模式，并使用户能够从复杂数据集中请求明确定义的科学特征，而无需详细了解数据布局或存储约束。

4.1 系统架构

集成式特征提取系统以通用的、基于树的数据超立方体表示为核心，为可用数据提供统一且忠实的视图。用户请求与底层数据存储之间的所有交互都通过这棵数据超立方体树进行中介；它编码的是数据空间的实际结构，而非理想化的稠密抽象。在我们的实现中，这种表示通过 Qube 实现。系统显式编码稀疏性、结构约束和分支，使其能够直接推理哪些坐标组合确实存在，而不依赖外部配置规则。

最终系统由三个紧密集成的组件构成：Polytope 提供特征提取所需的几何与算法核心；Qubed 维护可用数据的紧凑型基于树的索引；GribJump 则支持对底层存储后端进行细粒度的字节级访问。这些组件协同工作，将特征提取从面向字段或数据块的操作，转变为针对数据空间精确表示进行结构化遍历和筛选的过程。

Qube 由 Qubed软件 [10] 构建和维护。Qubed 通过扫描 FDB数据存储 [43, 44] 所公开的扁平元数据索引来构建这一表示，并将可用键组织为压缩树结构。树中的每个节点对应一对维度及其关联坐标，分支则表示数据集内的条件结构，例如仅针对特定模式、分辨率或预报步长存在的维度。由于数据集结构通常演变缓慢，所得 Qube 可作为预计算阶段的一部分进行构建，并缓存以供多次提取请求复用。

特征提取由 Polytope [29], 执行，它直接在此 Qube 表示上运行。以科学特征（如点、轨迹、区域或时间序列）表达的用户请求，由 Polytope 转换为抽象数据空间上的一组约束。Polytope 并不通过对坐标值取笛卡尔积来构建数据超立方体的稠密正交视图，而是遍历 Qube 树，并通过剪除与请求不兼容的分支来逐步过滤该树。这一过程既确保系统仅访问数据集中有效的坐标组合，也确保数据中编码的所有结构约束在整个提取工作流中得到遵守。

筛选操作会生成一个精简 Qube，展示针对特定用户请求，需要从后端存储中提取的确切数据子集。对于生成的树中的每个叶节点，Poly- tope 随后会计算相应的数据存储索引。这些索引会传递给 GribJump [9]„ 后者通过提供对 FDB 数据存储中数据的字节级访问来实现后端数据访问层。它不会加载完整的 GRIB 场或大气层，而只提取满足请求所需的字节范围。这既减少了系统输入输出，也免去了提取后裁剪的需要。该访问模型仍与现有存储后端和文件格式完全兼容，因为它直接操作底层数据表示，无需采用新的存储布局。

该系统的一项关键架构原则，是将数据的逻辑组织与物理存储明确解耦。Qube 表示描述逻辑结构、可用性和约束，而 GribJump 负责对已存储数据进行物理访问。Polytope 充当协调层，在这两个领域之间进行转换，确保提取逻辑由数据结构驱动，而非基于对底层存储布局的假设。

与 xarray [22], Zarr [50]或 Ras- DaMan [4, 3], 等现有多维数据框架相比，这些框架通常依赖稠密数组、预定义的分块策略或僵化的数据模型，而本架构将数据立方体的概念推广至稀疏、异构且有条件定义的数据空间。该系统将基于树的数据超立方体、几何筛选算法和字节级数据访问整合到单一框架中，为从各种气象与气候数据集中高效提取特征提供了灵活的基础。

完整的提取工作流如图 4 所示：首先根据 FDB 元数据构建一次 Qube 索引并将其缓存。随后，Polytope 通过对该索引进行结构化遍历和筛选，处理用户定义的提取请求。最后，将得到的有效数据索引集合传递给 GribJump，由其仅从数据存储中读取所需字节，以组装最终的提取特征。

4.2 系统性能

先前工作 [28] 中对 Polytope–GribJump 系统进行的许多性能评估，仍适用于新的提取流水线。不过，一个重要区别在于，以前在提取过程中构建数据立方体表示所需的高开销预计算阶段如今已被绕过。系统改为读取现有的 Qube 表示，该表示充当数据空间的缓存视图。这显著降低了性能开销，因为预计算阶段如今几乎可以瞬间完成；这对于实现真正的交互式工作流尤其有益。

引入 Qubed 抽象只会带来极少的运行时开销，因为它是在提取过程中接受查询的只读缓存，而非主动处理组件。不过，与所有数据访问系统一样，整体性能仍与访问模式密切相关，也取决于这些模式与底层存储布局的契合程度。当树结构反映数据的物理存储方式时，自然会凸显出能够将邻近数据归为一组的访问路径，从而实现更高效的检索，并为最佳使用方式提供指引。

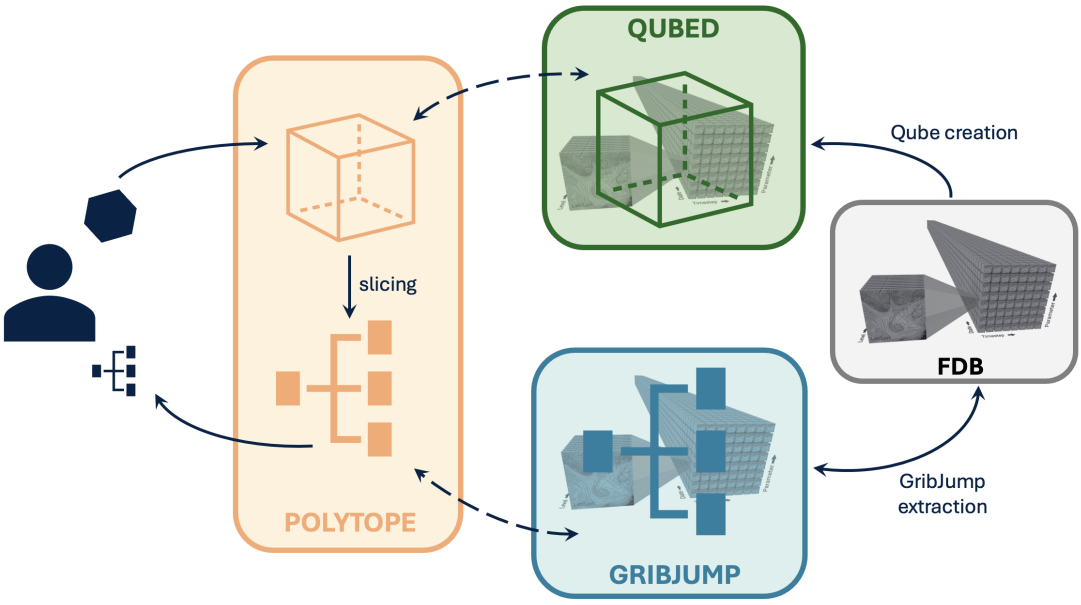


图 4: 包含 Polytope、GribJump 和 Qubed 组件的完整特征提取系统 .

模式。

在实践中，与原生存储布局相匹配的请求（例如单个场内的空间子集），处理速度通常快于跨越多个场的请求。Polytope 通过尽可能减少总输入/输出量，并将对连续存储数据的访问分组，来减轻这一影响；不过，算法开销仍会随着请求特征的规模和几何复杂度增加。这些影响在典型用法中十分明显。使用原生访问方法提取单个完整场通常只需不到一秒，而基于 Polytope 的提取可能需要几秒，因为它还涉及额外的几何处理。不过，Polytope 返回的数据仅占很小一部分，因此在许多场景中已经具有优势。

随着提取操作扩展到多个场，优势变得更加明显。例如，使用传统的全场访问方式提取涵盖 96 个场的完整预报时间序列需要数秒，而基于 Polytope 的方法对每个场仅提取单个点，所需时间几乎不会随规模增加而增长。当扩展到涉及数百个场的集合预报时，传统提取可能需要数分钟，而特征提取仍保持在数秒量级。这说明了该方法对于实践中常见、但原生数据布局支持不佳的访问模式十分有效。

与此同时，性能仍会受到底层存储系统和硬件配置的影响。数据排序、压缩方案、缓存行为以及存储介质性能等因素无法完全抽象化。因此，在新的后端或平台上部署该框架时，需要仔细检查访问模式并开展专门的性能分析。这种依赖性并非该系统所独有，同样适用于其他广泛使用的数据访问框架，包括基于 xarray 和 Zarr 的工作流。在所有此类系统中，当用户访问模式偏离数据布局和索引建立时所采用的假设，性能都可能下降。

因此，用户应当意识到，提取性能不仅取决于特征提取算法本身，还取决于数据如何在存储

层面被组织与访问。尽管该框架对尽量减少不必要的数据访问提供了有力保障，但最优性能最终仍需要对完整数据访问栈的整体把握，从用户请求模式一直到物理存储层。

4.3 以用户为中心的工作流与新访问模式

除了性能之外，所提出的特征提取框架的一个核心优势在于，它从根本上改变了用户与大规模天气与气候数据集交互的方式。数值天气预报和气候数据存储中的传统访问机制通常以完整场或大气层为粒度提供数据。因此，即使只需要其中很小的子集，用户也常常不得不检索大量数据，在开展任何科学分析之前，便给本地存储、数据传输和后处理带来了沉重负担。

本文开发的框架通过将特征提取从后处理步骤转变为数据访问层中的核心操作，解决了这一局限。压缩树形数据超立方体抽象在这一转变中发挥着重要作用。通过明确编码实际存在的数据及其内在约束和分支结构，树视图使用户能够安全而自然地浏览和查询数据空间。由于提取系统为用户查询的有效性提供了强有力的语义保证，因此可以确保请求只访问现有数据。

从用户的角度来看，这带来了更直观、更面向应用的交互模型。用户无需再使用文件、场或数据块等存储特定概念来构造请求，而是可以直接根据感兴趣的科学特征表达需求，包括点时间序列、轨迹、区域或任意时空子集。Polytope 会在内部将这些高层请求转换为精确的字节级访问模式，并利用 Qubed 表示来推理数据的可用性与结构。因此，用户只会获得所需的确切数据，且数据格式可直接用于下游分析。这种设计显著降低了使用气象数据集所需的工作量。事实上，用户不再需要详细了解数据格式、网格定义或后端存储约定，也能高效检索气象数据。他们可以专注于科学问题，而提取流水线则会透明地处理识别、筛选和访问相关数据的复杂工作。这对非专业用户、跨学科研究人员和应用开发者尤其有益；对他们而言，天气和气候数据是输入，而非主要研究对象。

这种方法的优势在交互式和探索性工作流中尤为明显。由于特征提取仅检索小型、有针对性的数据子集，请求通常能够足够快速地得到响应，从而支持迭代分析、可视化和原型开发。用户可以在多个位置探索时间序列、比较集合成员或细化感兴趣区域，而无需反复下载大型数据集。这种交互性难以通过传统的全场访问实现，并催生了新一类工作流，包括基于网络的应用、笔记本和实时决策支持系统。

该框架还通过降低本地资源需求来提高可访问性。现代气候模拟的全分辨率场很容易超出标准用户环境的内存和存储容量。相比之下，基于特征的访问返回紧凑的数据集，可在普通的本地硬件或轻量级基于云的环境中处理。这拓宽了对大规模天气和气候数据的访问，同时通过尽量减少不必要的数据移动和重复，支持更可持续的数据使用。

最后，特征提取服务具有通用性，因此能够以一致的方式跨多个数据系统运行。通过将抽象数据表示与物理存储分离，该框架可以充当异构后端之上的通用访问层。这使其非常适合作为标准化数据访问接口的后端，例如开放地理空间联盟（OGC）的环境数据检索（EDR）API{v1}

并可与数据目录[1], 、通知服务 [23] 以及下游处理管道 [42]无缝集成。在 Destination Earth 数字孪生 [48], 等实际运行场景中，这种统一的访问模式使用户能够通过单一、一致的接口发现、请求和使用数据，即使底层数据分布在多个系统中也是如此。

总体而言，该框架将数据访问的重点从大批量数据移动转向信息交付。用户可以请求所需的精确数据，并以可立即使用的形式获取，从而能够更高效、更便捷且以用户为中心地与大型科学数据存储进行交互。这一理念是该系统运营价值的核心，也彰显了它作为天气、气候和地球系统科学领域未来数据访问服务基础组件的潜力。

5 讨论与未来工作

特征提取正日益成为天气与气候科学中重要且广泛采用的数据访问模式。随着数据集的规模和复杂性不断增长，用户正逐步从全场访问转向更具选择性、由应用驱动的查询，只检索分析严格需要的数据。在哥白尼数据存储库（CDS） [33], 等业务系统中，这一趋势十分明显：相当一部分用户请求的是空间或时空子集，而非完整场。因此，高效、稳健地支持这类访问模式，是现代数据服务的一项关键要求。本文提出了一个完整且通用的特征提取框架，它通过直接对数据空间的树形抽象进行推理，适用于任意数据超立方体表示。该框架将逻辑数据组织与物理存储解耦，并避免对数据完备性、布局或正则性作出假设，从而将传统数据立方体模型推广至稀疏、异构且有条件定义的数据集。特征提取不再被视为数据检索后的后处理步骤，而是作为一项核心操作，直接嵌入数据空间的表示与遍历之中。

所提方法的一项核心优势在于其通用性。所有特定于数据和存储的细节都封装在通用抽象之后，因此同一提取流水线可以在不同数据集、存储系统和运行环境中重复使用。这使该框架尤其适用于多种数据源并存、且各自采用不同格式和惯例的环境，同时也能应对用户访问模式的显著差异。随着数据基础设施不断演进，只需进行少量修改便能在不同平台上重新部署该系统，这是一项关键优势。

运行服务中观察到的用户行为进一步印证了这一方法的实际价值。在 CDS 中，访问 ERA5 单层数据集的用户约有 62% 会应用区域约束，其中约 22% 的用户针对小于 $10° \times 10°$ 的区域。这些访问模式与特征提取天然契合，并可直接受益于数据传输量和后端输入/输出的减少。尽管具体的性能提升取决于硬件和存储配置，该框架因此显然有望缓解输入/输出瓶颈，并提升哥白尼生态系统的可扩展性。

这一潜力已通过基于 Polytope 的提取服务在 Destination Earth计划中的实际部署得到验证 [48, 14], ，该服务每天都用于访问天气与气候数字孪生数据。在这一场景中，特征提取是主要的数据访问机制，既支持交互式工作流，又能与现有目录和服务无缝集成。基于这一经验，该框架计划整合到统一的 CDS 中，预计将使大量用户受益，

为用户提供更快速、更高效、更便捷的数据提取。
本研究揭示了未来工作的几个方向。首要任务是将该框架集成到更广泛的后端数据存储和硬件平台中，并对其进行评估。虽然该系统旨在尽可能减少输入/输出，并使访问模式与底层存储布局相匹配，但总体性能不可避免地取决于后端的具体特性。因此，有必要针对不同存储技术开展系统评估，以了解其中的权衡并指导部署策略。

第二个方向是基于观察到的用户访问模式进一步优化。通过分析用户在实际应用中如何与数据交互，可以调整数据超立方体表示、遍历顺序或缓存策略，使其更好地适应主导工作流。这为超越通用性能调优的访问模式优化开辟了空间。

最后，该框架很适合将更丰富的元数据集成到数据超立方体表示中。将网格定义、物理数据位置、路由约束或特定服务注释等附加信息直接嵌入基于树的抽象中，能够让提取系统做出更明智的决策，并以完全信息驱动的方式运行。此类富含元数据的数据超立方体将进一步减少对硬编码假设的依赖，并支持在异构系统间实现真正通用且自描述的数据访问。

总的来说，这项工作展示了压缩树形数据超立方体表示、几何特征提取与字节级数据访问相结合，如何形成一个连贯、可扩展且以用户为中心的数据访问框架。该方法弥合了后端存储系统与用户驱动的工作流之间的鸿沟，为天气、气候和地球系统科学领域未来的数据服务提供了切实可行的基础；在这些领域，高效、灵活且可持续地访问大型数据集正变得日益重要。

致谢

本文所述工作是在欧盟“Destination Earth”计划框架下开展的，与欧盟委托欧洲中期天气预报中心实施该计划部分内容的相关任务有关，并由欧盟提供资金支持。文中所表达的观点和意见仅代表作者本人，并不一定反映欧盟或欧盟委员会的立场。欧盟和欧盟委员会均不对这些观点和意见承担责任。

参考文献

[1]Qubed目录浏览器。https://qubed.lumi.apps.dte.destination-earth.eu/，2025年。于2026年2月4日访问。[2] Anca Anghelea、Ewelina Dobrowolska、Gunnar Brandt、Martin Reinhardt、Miguel Mahecha、Tejas Morbagal Harish和Stephan Meissl。深地球系统数据实验室（DeepESDL）。国际摄影测量、遥感与空间信息科学档案，48：13–18，2024年。[3] Peter Baumann、Andreas Dehmel、Paula Furtado、Roland Ritsch和Norbert Widmann。多维数据库系统RasDaMan。载于*1998*年*ACM SIGMOD*数据管理国际会议论文集，第575–577页，1998年。

[4] Peter Baumann、Paula Furtado、Roland Ritsch和Norbert Widmann。RasDaMan多维数据库管理方法。载于*1997*年*ACM*应用计算研讨会论文集，第166–173页，1997年。[5] M Burgoyne、D Blodgett、C Heazel和C Little。OGC API 环境数据检索标准。开放地理空间联盟，*Wayland*，马萨诸塞州，美国，*OpenGIS*®实施规范 *OGC*。[6] 深层地球系统数据实验室（DeepESDL）团队。地球系统数据实验室。https: // earthsystemdatalab.net/，2025年。于2025年12月30日访问。[7] Francisco J Doblas-Reyes、Jenni Kontkanen、Irina Sandu、Mario Acosta、Mohammed Hussam Al Turjmam、Ivan Alsina-Ferrer、Miguel Andrés-Martínez、Leo Arriola、Marvin Axness、Marc Batlle Martín等。面向气候变化适应的Destination Earth 数字孪生。*EGUsphere*，2025：1–41，2025年。[8] Brian Eaton、Jonathan Gregory、Bob Drach、Karl Taylor、Steve Hankin、John Caron、Rich Signell、Phil Bentley、Greg Rappa、Heinke Höck等。NetCDF 气候与预报（CF）元数据规范，2003年。[9] 欧洲中期天气预报中心。GribJump。https://github.com/ecmwf/gribjump，2025年。于2026年2月4日访问。[10] 欧洲中期天气预报中心（ECMWF）。Qubed。https:// github.com/ecmwf/qubed，2025年。于2026年1月5日访问。[11] Guillaume Eynard-Bontemps、Ryan Abernathey、Joseph Hamman、Aurelien Ponte和Willi Rath。PANGEO 大数据生态系统及其在法国国家空间研究中心的应用。载于太空大数据（*BiDS'19*）……将数据转化为洞见……*2019*年*2*月*19*日至*21*日，德国慕尼黑，2019年。[12] Marzieh Fathi、Mostafa Haghi Kashani、Seyed Mahdi Jameii和Ebrahim Mahdipour。天气预报中的大数据分析：系统综述。《计算工程方法档案》，29(2)：1247–1275，2022年。[13] Estíbaliz Gascón、Irina Sandu、Benoît Vannière、Linus Magnusson、Richard Forbes、Inna Polichtchouk、Annelize Van Niekerk、Birgit Sützl、Michael Maier-Gerber、Michail Diamantakis等。Destination Earth计划在更好预测极端天气方面的进展。技术报告，哥白尼会议，2023年。[14] Thomas Geenen、Nils Wedi、Sebastian Milinski、Ioan Hadade、Balthasar Reuter、Simon Smart、James Hawkes、Emma Kuwertz、Tiago Quintino、Emanuele Danovaro等。数字孪生：业务天气预报系统迈向Destination Earth 核心的历程。《计算机科学进展》，240：99–109，2024年。[15] Gregory Giuliani、Bruno Chatenoux、Antonio Benvenuti、Pierre Lacroix、Mattia Santoro和Paolo Mazzetti。利用卫星地球观测时间序列数据在国家层面监测土地退化以支持可持续发展目标15——探索数据立方体的潜力。《地球大数据》，4(1)：3–22，2020年。[16]Gregory Giuliani、Bruno Chatenoux、Andrea De Bono、Denisa Rodila、Jean-Philippe Richard、Karin Allenbach、Hy Dao和Pascal Peduzzi。构建地球观测数据立方体：从瑞士数据立方体（SDC）生成分析就绪数据中获得的经验教训

(ard). *Big* 地球数据 ，1(1-2)：100–117 ，2017年。

[17] Jim Gray、Surajit Chaudhuri、Adam Bosworth、Andrew Layman、Don Reichart、Murali Venkatrao、Frank Pellow和Hamid Pirahesh。数据立方体：一种推广分组、交叉表和小计的关系聚合操作符。数据挖掘与知识发现，1(1):29–53，1997年。[18] Hua-Dong Guo、Li Zhang和Lan-Wei Zhu。面向气候变化研究的地球观测大数据。《气候变化研究进展》，6(2):108–117，2015年。[19] Huadong Guo。地球大数据：地球与信息科学的新前沿。《地球大数据》，1(1-2):4–20，2017年。[20] Venky Harinarayan、Anand Rajaraman和Jeffrey D Ullman。高效实现数据立方体。*ACM SIGMOD Record*，25(2):205–216，1996年。[21]Jörn Hoffmann、Peter Bauer、Irina Sandu、Nils Wedi、Thomas Geenen和Daniel Thiemert。Destination Earth——支持气候服务的数字孪生，2023年。[22] Stephan Hoyer和Joe Hamman。xarray：Python中的N维带标签数组与数据集。《开放研究软件期刊》，5(1):10–10，2017年。[23] Claudio Iacopino、James N Hawkes、Tiago Quintino和Baudouin Raoult。Aviso：借助面向数值天气预报数据可用性的高吞吐量通知系统连接高性能计算与云。载于美国气象学会会议摘要，第101卷，第10–8页，2021年。[24] Mehmet Kaya和Reda Alhajj。采用一种新颖的基于数据立方体的建模方法发展多维学术信息网络。《信息科学》，265:211–224，2014年。[25] Brian Killough。开放数据立方体倡议概述。载于*IGARSS 2018-2018 IEEE*国际地球科学与遥感研讨会，第8629–8632页。IEEE，2018年。[26] Soohyung Kim、Hyukki Lee和Yon Dohn Chung。面向电子病历的隐私保护数据立方体：实验评估。《国际医学信息学杂志》，97:33–42，2017年。[27] Tennessee Leeuwenburg、Nicholas Loveday、Elizabeth E Ebert、Harrison Cook、Mohammadreza Khanarmuei、Robert J Taggart、Nikeeth Ramanathan、Maree Carroll、Stephanie Chong、Aidan Griffiths等。scores：一个借助xarray验证和评估模型与预测的Python软件包。*arXiv*预印本 *arXiv:2406.07817*，2024年。[28]Mathilde Leuridan、Christopher Bradley、James Hawkes、Tiago Quintino和Martin Schultz。从大规模气象数据存储库中进行特征提取的高效算法的性能分析。载于高性能科学计算平台会议论文集，第1–9页，2025年。[29] Mathilde Leuridan、James Hawkes、Simon Smart、Emanuele Danovaro、Martin Schultz和Tiago Quintino。Polytope：一种在超立方体上高效进行特征提取的算法。《大数据期刊》，12(1):1–25，2025年。[30]Mathilde Leuridan、Adam Warde、Oisin-M、James Hawkes、James Varndell、Peter Tsrunchev、Chris Bradley、Dušan Figala、Iain Russell、Kai Kratz、Simon Smart和Yordan Radev。ecmwf/polytope：2.1.1，2025年10月。

[31] Adam Lewis、Simon Oliver、Leo Lymburner、Ben Evans、Lesley Wyborn、Norman Mueller、Gregory Raevksi、Jeremy Hooke、Rob Woodcock、Joshua Sixsmith 等。《澳大利亚地球科学数据立方体——基础与经验教训》。《环境遥感》，202:276–292，2017年。[32] Zhenlong Li、Fei Hu、John L Schnase、Daniel Q Duffy、Tsengdar Lee、Michael K Bowen 和 Chaowei Yang。《一种利用 MapReduce 高效处理基于大数组的气候数据的时空索引方法》。《国际地理信息科学杂志》，31(1):17–35，2017年。[33] Angel Lopez、Carlo Buontempo、Martin Suttie、Baudouin Raoult、Edward Comyn-Platt 和 James Varndell。《用于改善哥白尼计划气候与大气数据及服务访问的现代化数据存储基础设施》。技术报告，哥白尼会议，2022年。[34] Roxanne Suzette Lorilla。《Sentinel Hub 应用新功能报告》。*Zenodo*（欧洲核子研究组织），2025年。[35] Chris Mauzey、Charles Doutriaux、Denis Nadeau、Karl E Taylor、Paul J Durack、Edward Betts、Antonio S Cofino、Piotr Florek、Emma Hogan、Jose M Rodriguez Gonzalez 等。《气候模式输出重写器（CMOR）》。*Zenodo*，2025年。[36] Konstantinos Morfonios、Stratis Konakas、Yannis Ioannidis 和 Nikolaos Kotsis。《数据立方体的 ROLAP 实现》。*ACM* 计算调查（*CSUR*），39(4):12–es，2007年。[37] Thomas Nicholas 和 Julius Busecke。《Xarray-Datatree：面向多模式科学的层次数据结构》。发表于第*103*届美国气象学会年会。美国气象学会，2023年。[38] Ekow J Otoo、Doron Rotem 和 Sridhar Seshadri。《面向数据仓储的大型多维数组最优分块》。发表于*ACM* 第十届数据仓储与联机分析处理国际研讨会论文集，第25–32页，2007年。[39] Jonathan T Overpeck、Gerald A Meehl、Sandrine Bony 和 David R Easterling。《21世纪的气候数据挑战》。《科学》，331(6018):700–702，2011年。[40] Baudouin Raoult。《Mars：欧洲中期天气预报中心的气象档案——大型档案管理经验》。2013年。[41] Russ Rew 和 Glenn Davis。《NetCDF：科学数据访问接口》。《*IEEE* 计算机图形与应用》，10(4):76–82，1990年。[42] Iain Russell、Tiago Quintino、Baudouin Raoult、Sándor Kertész、Pedro Maciel、James Varndell、Corentin Carton de Wiart、Edward Comyn-Platt、Olivier Iffrig、James Hawkes 等。《介绍 Earthkit》，2024年。[43] Simon D Smart、Tiago Quintino 和 Baudouin Raoult。《面向气象与气候数据的可扩展对象存储》。发表于高性能科学计算平台会议论文集，第1–8页，2017年。[44] Simon D Smart、Tiago Quintino 和 Baudouin Raoult。《面向百亿亿次级数值天气预报与气候模拟的高性能分布式对象存储》。发表于高性能科学计算平台会议论文集，第1–11页，2019年。[45] José Unpingco。《用于概率、统计学和机器学习的 *Python*》，第1卷。施普林格，2016年。

[46] Wil MP Van Der Aalst。《流程立方体：对用于流程挖掘的事件数据进行切片、切块、上卷和下钻》。发表于亚太业务流程管理会议，第1–22页。施普林格，2013年。[47] Tiffany C Vance、Thomas Huang 和 Kevin A Butler。《地球科学中的大数据：新兴实践与前景》。《科学》，383(6688):eadh9607，2024年。[48]Nils Wedi、Peter Bauer、Irina Sandu、Jörn Hoffmann、Sophia Sheridan、Rafael Cereceda、Tiago Quintino、Daniel Thiemert 和 Thomas Geenen。《Destination Earth：面向天气与气候的高性能计算》。《科学与工程计算》，24(6):29–37，2023年。[49] xarray-contrib。《xarray-contrib/datatree：面向 xarray 的树状层次数据结构原型实现》。https://github.com/xarray-contrib/datatree，2024年。GitHub 存储库（已存档）。访问日期：2025年12月30日。[50] Zarr 开发者。《Zarr：分块、压缩的 N 维数组》。https://zarr.dev/，2025年。访问日期：2025年12月30日。[51] Peixiang Zhao、Xiaolei Li、Dong Xin 和 Jiawei Han。《图立方体：多维网络的数据仓储与联机分析处理》。发表于 *2011*年 *ACM SIGMOD* 国际数据管理会议论文集，第853–864页，2011年。