软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doe: 10.13328/j.cnki.jos.000000]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

当时空数据管理遇到时空 AI: 进展、挑战与展望*

苏赛男[1,2], 李瑞远[1,2], 杨广超[1,2], 但静培[1], 龙程[3], 张钧波[4,5,7], 郑宇[4,6,7]

[1](重庆大学 计算机学院, 重庆 401331)

[2](重庆大学 时空实验室, 重庆 401331)

[3](南洋理工大学 计算与数据科学学院, 新加坡 639798)

[4](西南交通大学 计算机与人工智能学院, 成都 611756)

[5](北京京东智能城市大数据研究院, 北京 100176)

[6](西安电子科技大学 网络与信息安全学院, 西安 710071)

[7](交通数据挖掘与具身智能北京市重点实验室, 北京 100044)

通讯作者: 李瑞远, 杨广超 E-mail: ruiyuan.li@cqu.edu.cn, gchao_yang@cqu.edu.cn

摘 要: 随着时空数据规模的持续增长, 如何高效管理并挖掘其中有用信息已成为重要研究课题. 时空数据管理的核心在于实现数据的高效存储、索引和查询. 然而, 传统数据库技术难以有效应对时空数据高动态等特性. 人工智能(Artificial Intelligence, AI)技术能够有效捕捉时空数据的分布特征与查询负载等信息, 将其融入时空数据管理可提升系统的智能化水平. 时空 AI 则致力于将机器学习、深度学习与强化学习等技术应用时空数据分析, 自动识别数据中的模式、趋势与关联, 并支持时空预测、分类、聚类及异常检测等任务. 然而, 时空 AI 在数据获取、模型训练到实际应用的全过程中, 均面临数据异构性、准备复杂性及使用门槛高等挑战. 时空数据管理技术可有效缓解上述问题. 围绕以上工作, 工业界和学术界已开展了大量研究工作. 首先提出了时空数据的分类方法, 将其划分为独立数据与关联数据; 继而系统梳理了时空数据管理与时空 AI 协同的研究进展, 总结其研究背景与关键技术; 最后, 整理了该领域常用数据集, 介绍了典型应用案例, 探讨了面临的主要挑战, 并展望了未来发展方向.

关键词: 时空数据; 时空数据管理; 时空 AI; AI4DB; DB4AI

中图法分类号: TP311

中文引用格式: 李瑞远,苏赛男,杨广超,但静培,龙程,张钧波,郑宇.当时空数据管理遇到时空 AI: 进展、挑战与展望.软件学报,202x,xx(x). http://www.jos.org.cn/1000-9825/xxxx.htm

英文引用格式: Li RY, Su SN, Yang GC, Dan JP, Long C, Zhang JB, Zheng Y. When Spatio-Temporal Data Management Meets Artificial Intelligence: Progress, Challenges and Prospects. Ruan Jian Xue Bao/Journal of Software, 202x (in Chinese). http://www.jos.org.cn/1000-9825/xxxx.htm

When Spatio-Temporal Data Management Meets Spatio-Temporal Artificial Intelligence: Progress, Challenges and Prospects

LI Rui-Yuan[1,2], SU Sai-Nan[1,2], YANG Guang-Chao[1,2], DAN Jing-Pei[1], LONG Cheng[3], ZHANG Jun-Bo[4,5,7], ZHENG Yu[4,6,7]

[1](College of Computer Science, Chongqing University, Chongqing 401331, China)

[2](Start Lab, Chongqing University, Chongqing 401331, China)

[3](College of Computing and Data Science, Nanyang Technological University, Singapore 639798, Singapore)

[4](School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China)

[5](JD Intelligent Cities Research, Beijing 100176, China)

[6](School of Cyber Engineering, Xidian University, Xi'an 710071, China)

* 基金项目: 国家自然科学基金(62572086, 72242106), 山东省重大基础研究项目(ZR2024ZD03)
收稿时间: 2025-04-13; 修改时间: xxxx-xx-xx; 采用时间: xxxx-xx-xx

[7](Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing 100044, China)

Abstract: With the rapid growth of spatio-temporal data, how to efficiently manage spatio-temporal data and extract valuable information from them has become a crucial research topic. On one hand, the core of spatio-temporal data management (STDM) is to efficiently store, index, and query spatio-temporal data. However, traditional database technologies struggle to handle the unique characteristics of spatio-temporal data, such as spatio-temporal dynamics. Artificial intelligence (AI) techniques can effectively capture the distribution characteristics of spatio-temporal data and query workloads, which enhances the intelligence of spatio-temporal data management systems. On the other hand, spatio-temporal AI (STAI) refers to the application of AI techniques (e.g., machine learning, deep learning, and reinforcement learning) to spatio-temporal data analysis. STAI enables the automatic identification of patterns, trends, and relationships in data and supports tasks such as spatio-temporal prediction, classification, clustering, and anomaly detection. Nevertheless, across all stages, ranging from data acquisition to model training to ultimate deployment, STAI confronts challenges stemming from heterogeneity of spatio-temporal data, the complexity of data preparation, and the high barrier of model usage. These issues can be effectively mitigated by spatio-temporal data management technologies. Focusing on these topics, extensive research has emerged in both academia and industry. In this paper, we first propose a taxonomy for spatio-temporal data, categorizing it into independent data and associated data. Then, by investigating the research background and key techniques, we systematically review the research progress of integrating STDM with STAI. Finally, we catalog commonly used datasets in the field, introduce some representative applications, discuss the key challenges faced, and outline future research directions for the synergy achievement of STDM and STAI.

Key words: spatio-temporal data; spatio-temporal data management; spatio-temporal AI; AI4DB; DB4AI

时空数据[1]是指同时具有时间属性和空间属性的数据. 随着全球定位系统、移动设备、射频识别及遥感等空间感知和地理信息技术的迅猛发展, 时空数据呈爆炸式增长. 例如, Twitter 每天发布逾 1000 万条带有地理标记和时间戳的推文[2]; 滴滴公司每日产生超过 150 亿个位置点, 处理数据量达 70TB[3]; 京东快递员日均产生 1TB 以上的位置记录点[4]. 此类海量时空数据不仅对存储与管理提出了更高要求, 也为数据分析与智能应用带来了新的机遇. 在此背景下, 时空数据管理(Spatio-Temporal Data Management, STDM)与时空人工智能(Spatio-Temporal Artificial Intelligence, STAI)相互促进、协同发展. STDM 关注时空数据的高效存储、索引与查询, 以优化数据管理与访问效率; STAI 则借助人工智能技术挖掘时空数据中的模式和规律, 进而支持智能决策. 时空数据不仅具备海量、动态及多源异构的特征, 还呈现出复杂的时空分布规律: 在空间维度上常表现为倾斜性与层次性(如城市中心轨迹密集而偏远地区轨迹稀疏); 在时间维度上则具有显著的周期性与趋势性(如气温随季节波动、交通流的早晚高峰循环). 上述特性使得传统数据库技术难以满足 STDM 对实时处理与计算效率的需求[5]. 人工智能技术能够学习数据分布特征并感知系统工作负载, 从而优化时空数据组织与查询引擎设计, 提高 STDM 的效率. 同时, 相较于传统基于统计的方法, STAI 在时空特征学习方面虽表现优越[1], 但在实际应用中仍面临效率较低和使用不便等问题. STDM 可有效整合多源异构的时空数据, 提升 STAI 的训练效率, 降低其使用门槛.

目前, 数据库智能化(AI4DB)与面向 AI 的数据管理技术(DB4AI)已得到广泛研究. Jasny 等人[6]探讨了如何将机器学习算法有效集成至现有数据库管理系统中, 并介绍了名为 DB4ML 的内存数据库内核. Chai 等人[7]从数据库架构角度出发, 系统阐述了学习式数据库各组件的研究动机、基本思路与关键技术. Li 等人[8]详细总结了基于声明式语言模型的 AI 系统、面向 AI 优化的计算引擎与执行引擎, 以及面向 AI 的数据治理引擎的研究进展, 并在论文[9]中提出 AI 原生数据库的设计思想. Chai 等人[10]归纳了数据库技术在机器学习数据准备、模型训练与推理、模型管理的应用. Zhou 等人[11]发表了关于数据库智能化及面向 AI 的数据管理技术综述. 以上研究的对象主要为关系型数据库与通用 AI 算法, 由于时空数据与通用数据差异显著, 相关成果难以直接迁移至 STDM 系统和 STAI 应用中.

Wang 等人[1]系统梳理了深度学习在时空数据挖掘中的最新进展, 但未探讨其与时空数据管理的关联. Liang 等人[12]在全面探索了时空大数据分析生态系统, 并将相关技术划分为五个模块: 时空大数据、计算资源、处理平台、资源管理和应用程序, 但未涉及人工智能方法与时空数据平台间的交互影响. 目前, 关于 STDM 和 STAI 协同应用领域的研究仍缺乏系统性综述. 本文的核心贡献在于, 从时空数据类型的独特视角出发, 系统梳理和总结近年来在两个关键方向上的研究进展: 一是运用 AI 技术提升时空数据管理的智能化水平

(ST-AI4DM), 二是如何构建面向 STAI 应用的高效时空数据管理技术(ST-DM4AI).

本文第 1 节概述时空数据及其特性; 第 2 节界定 ST-AI4DM 和 ST-DM4AI 的研究范畴与研究分类; 第 3、4 节分别聚焦不同数据类型, 深入介绍 ST-AI4DM 和 ST-DM4AI 的代表性工作; 第 5 节梳理了该领域的常用数据集; 第 6 节介绍 ST-AI4DM 与 ST-DM4AI 的典型应用案例; 第 7 节分析该领域存在的主要问题与挑战, 并展望未来研究方向; 第 8 节总结全文.

1 时空数据及其特性

时空数据是指同时具有时间属性和空间属性的数据, 其来源广泛、类型多样. 在时空查询中, 不同类型的数据需采用不同的索引结构, 以提高查询效率; 在 STAI 应用中, 各类型适用的 AI 模型已存在差异. 因此, 合理的时空数据分类对 STDM 与 STAI 的应用具有重要意义.

如图 1 所示, 本文首先根据数据点间的时空依赖关系, 将时空数据划分为“独立数据”与“关联数据”两大类. 独立数据指数据点间无显著时空依赖的数据; 关联数据则指在空间上存在拓扑关系或时间上存在序列依赖的数据. 在此基础上, 进一步根据数据的动静态特性, 将上述两大类数据细分为静态与动态两种类型. 最终, 本研究形成了一个包含四类数据的分类体系.

(1) **静态独立数据**: 指具有固定空间位置、且不随时间产生读数的独立数据. 此类数据通常反映的是静态的空间实体, 如兴趣点(Point Of Interest, POI)数据.

(2) **动态独立数据**: 指空间位置随着时间变化的独立数据, 如签到数据、交通事故的发生地点数据等.

(3) **静态关联数据**: 指在空间上固定、与其他数据元素存在拓扑关系、且不随时间产生读数的数据. 例如, 城市的道路网络和公共设施的空间布局.

(4) **动态关联数据**: 指空间位置或数据点属性值随时间动态变化, 且不同数据点之间存在拓扑关系或者时序依赖的数据, 如车辆轨迹数据. 空间时序数据(如空气质量监测站点数据)也属于此类.

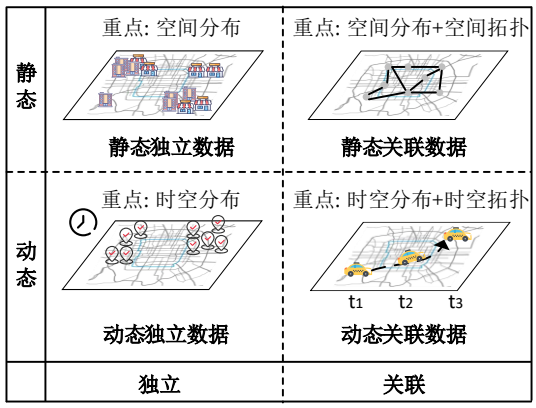


图 1 时空数据分类及需关注的重点特性

该分类方法能涵盖本综述调研的所有时空数据类型, 也为后续 STDM 和 STAI 的协同分析提供了更为简洁高效的分类思路. Wang 等人[1]从 STAI 的角度将时空数据划分为事件数据、轨迹数据、点参考数据、栅格数据与视频数据. 依据本文的分类体系, 事件数据和点参考数据可归为动态独立数据, 栅格数据(如 DEM、土地利用图等像元间存在空间邻近关联且更新较慢的数据)属于静态关联数据, 轨迹数据和视频数据则属于动态关联数据. Wang 等人的分类未包含静态独立数据(如建筑物分布、交通基础设施等), 而此类数据在时空管理与分析中同样具有重要意义.

时空数据采集方式多样, 且受多种因素的影响, 通常具备以下特征:

(1) **海量、动态、多源、异构、高维**[13]: 时空数据通常具有数据量巨大、动态变化、来源多样、结构异构以及高维等特点. 以交通轨迹数据为例, 其来源包括 GPS 设备、交通监控摄像头、用户手机信号

基站与路网感应器等. 这些数据在技术、设备、精度、采集频率及更新速度方面存在差异, 同时涵盖了结构化数据(如车辆的 GPS 坐标和速度)、半结构化数据(如移动设备采集的用户轨迹)以及非结构化数据(如交通监控视频). 在维度层面, 交通轨迹通常包含经度、纬度、时间戳、速度、加速度等信息, 每条轨迹还可能关联交通设施(如信号灯、路标)或交通状况(如高峰期、事故发生)等多维信息.

(2) **空间维度的倾斜性、层次性和距离性**[14]. 时空数据在空间上通常呈现非均匀分布的特征. 如城市中心交通流量高, 轨迹数据密集; 偏远地区则相对稀疏. 层次性表现为空间单元的多级划分, 如城市、社区、街道等不同层级之间的组织方式. 在时空数据查询和挖掘中, 空间距离是重要考量因素.

(3) **时间维度上的邻近性、周期性和趋势性**[14] . 时空数据在时间维度上通常表现为邻近性, 即相邻时段的数据变化幅度有限, 如同一区域上午 10 点至 10 点半的车流量变化较小. 周期性体现为季节性等规律波动, 如气温随季节更替呈周期性变化. 趋势性则反映为时间序列上的渐进演变, 例如冬季昼长缩短导致早高峰时间逐渐推迟.

由于时空数据具有上述特性, 通用 AI4DB/DB4AI 的方案无法直接应用于时空领域. 在考虑时空数据管理与时空 AI 领域协同时, 不同类型的时空数据面临的核心挑战不同. 如图 1 所示, 静态独立数据的核心难点在于空间分布的不均匀性; 动态独立数据将挑战扩展至时空分布, 需应对海量数据在时间维度上的动态变化; 静态关联数据的核心难点在于空间分布与空间拓扑, 需同时处理空间位置及对象之间的空间拓扑关系; 动态关联数据最为复杂, 其关键在于时空分布与时空拓扑的融合, 通常涉及对象随时间演化产生的强关联规律. 后文将详细阐述现有研究如何针对上述难点实现技术突破.

2 ST-AI4DM 和 ST-DM4AI 的研究概述

为明确 ST-AI4DM 和 ST-DM4AI 的研究范畴, 并凸显不同类型时空数据所应对的核心挑战与研究重点, 基于第一节建立的时空数据分类体系, 本文构建了如图 2 所示的研究框架. 该框架将 ST-AI4DM 研究进一步划分为基于学习的时空数据存储、基于学习的时空数据索引和基于学习的时空查询优化三个方面.

(1) **基于学习的时空数据存储:** 针对时空数据海量、动态与高维特性给传统存储技术带来的挑战, 亟需优化存储架构以实现快速数据访问、高效数据压缩及空间资源的合理利用. 其中, 时空数据简化(即在保留关键时空特征的前提下减少数据冗余)与时空数据分区是提升存储效率的两项关键技术. 时空数据简化的现有研究多以降低存储成本为主要目标, 针对查询精度优化的简化方法则相对较少[15]. 传统的时空数据分区策略则常因忽略数据的内在分布特性导致效率低下, 基于学习的方法通过深度挖掘数据的分布特性, 能够更智能地优化分区方案, 从而显著提升时空数据存储的分区性能.

(2) **基于学习的时空数据索引:** 随着时空数据规模急剧增长, 传统时空索引在性能、存储开销与维护成本等方面面临诸多挑战, 例如索引构建时间随数据量激增而显著延长. 学习索引通过引入机器学习模型学习数据的累积分布函数或其他统计特性, 以替代或增强传统索引结构[16]. “AI 辅助索引”借助机器学习指导时空索引的构建与调整, 从而提升查询效率, 并增强对时空数据动态变化的适应性.

(3) **基于学习的时空查询优化:** 传统的查询优化算法通常依赖于固定的规则或启发式方法, 在处理大规模、高动态变化的时空数据时, 难以生成最优的查询执行计划. 基于学习的时空查询优化算法利用机器学习技术, 通过从历史时空查询记录中学习, 自动发现数据访问的规律, 从而为复杂时空查询生成更高效的执行计划.

除了上述三个方面, 基于学习的时空查询用户交互[17][18]近年来也受到初步关注. 该方向主要借助大语言模型等人工智能技术, 允许用户以自然语言方式直接发起时空查询, 从而显著降低时空数据管理的使用门槛, 提升人机交互的便捷性与灵活性. 然而, 目前该领域的研究尚处于起步阶段, 相关文献较为有限, 技术体系尚未成熟, 因此本文暂不对此展开详细论述. 可以预见, 随着大模型技术的不断发展, 基于学习的时空查询用户交互将成为未来时空数据管理的重要研究方向之一.

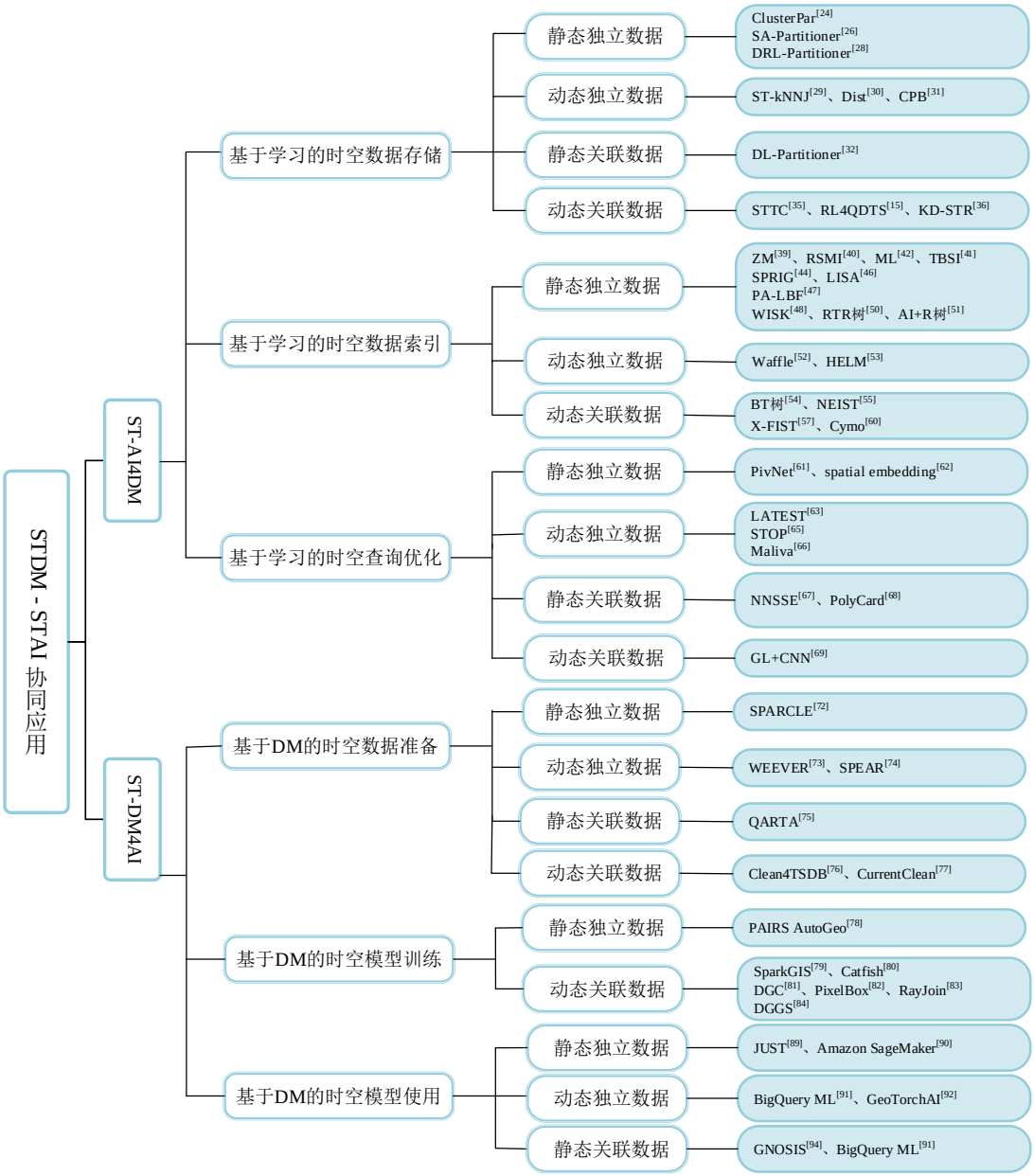


图 2 ST-AI4DM 和 ST-DM4AI 的研究分类

相应地，该框架将 ST-DM4AI 的工作划分为基于 DM 的时空数据准备、基于 DM 的时空模型训练和基于 DM 的时空模型使用三个阶段.

(1) **基于 DM 的时空数据准备:** STAI 模型的性能高度依赖于训练数据的质量与规模. 低质量的数据可能导致模型学习到错误的模式，进而影响预测的准确性和可靠性. 基于 DM 的数据准备致力于通过系统化的方法，解决原始时空数据在应用于 AI 模型前存在的各种问题.

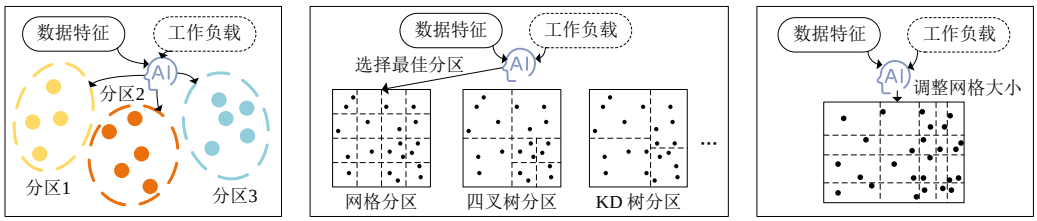
(2) **基于 DM 的时空模型训练:** 传统的 STAI 模型训练过程往往涉及繁琐的数据导入导出、手动特征构建以及不同计算平台间的切换，导致效率低下且难以扩展. 基于 DM 的模型训练旨在通过将数据管理能力与模型训练流程深度融合，从而简化并加速模型开发.

(3) **基于 DM 的时空模型使用:** 将训练好的 STAI 模型应用于实际场景并发挥其价值, 是 STAI 生命周期的最终环节, 但也常面临技术门槛高、集成复杂的挑战. 基于 DM 的模型使用专注于降低 STAI 模型的应用与部署难度, 提升其在业务系统中的可访问性与易用性.

3 ST-AI4DM

3.1 基于学习的时空数据存储

在分布式系统中, 高效的数据存储是实现可伸缩性和高性能的关键. 对于基于分布式架构的 STDM, 其基本要求之一在于能对数据进行有效的时空分区, 然后通过并行处理分区以实现横向扩展, 如 Spatial Hadoop[19]、Scala-GiST[20]、SATO[21]、Simba[22]等空间数据管理系统采用基于采样的分区方法. 分区方法对于许多空间分析操作的性能影响重大, 例如在计算几何操作中, 分区将复杂的几何计算分解为多个子问题从而实现并行处理[23]. 传统分区方法通常依赖简单的启发规则或采样策略, 难以有效应对复杂、动态变化和高维的时空数据. 基于学习的数据分区通过学习数据特征、数据分布或访问模式, 生成更智能、更适应负载的分区策略. 如图 3 所示, 基于学习的分区策略可进一步细分为以下三类: (1) **模型代替分区.** 模型直接学习从原始数据到分区结果的映射关系. 该方法自动化程度高, 但是模型的泛化能力受限, 仅在特定数据分布下表现较好. (2) **AI 辅助分区**. 模型评估多种传统分区技术在特定数据集和查询工作负载下的表现, 从中选择最优方案. 该方法融合统计学知识, 具备较强的可解释性, 模型训练完成后推理成本较低. 其缺点在于特征工程复杂度较高, 需要设计合适的特征来描述数据分布和关联性. (3) **AI 优化分区构建.** 该方案将分区过程建模为优化问题, 通过自动调整传统分区方法中的关键参数(如网格大小、树的深度或划分阈值), 以动态优化分区结构. 该方法能直接弥补传统分区在适应性上的缺陷, 但是模型训练成本高, 且结果可解释性较弱, 且实现难度较大, 需要深度修改数据库内核以支持实时的分区决策和数据重分布.



(a) 模型代替分区 (b) AI辅助分区选择 (c) AI优化分区构建

图 3 基于学习的时空分区方法

与此同时, 海量时空数据的有效管理是另一核心挑战. 对于动态关联数据, 其数据量通常极为庞大且包含大量冗余信息. 数据简化因此成为关键的预处理技术, 旨在以最小的信息损失减小数据规模.

3.1.1 静态独立数据

静态独立数据的空间位置相对固定且不随时间产生读数, 其空间分布特性通常是设计高效空间分区方案的核心依据.

模型替代分区. ClusterPar[24]是经典的模型替代分区的方法, 它基于 K-Means 聚类[25]算法对静态独立数据进行空间分区, 其分区数量与计算系统线程数一致. 模型输入为包含空间对象的数据集, 输出为每个对象的分区标识符. 该方法的核心思想是利用静态独立数据的距离特性, 将数据划分为多个子集, 以便在并行计算环境中高效处理.

AI 辅助分区选择. Belussi 等人提出的基于偏斜度感知的 SA-Partitioner[26]利用盒计数函数[27]量化数据集分布偏斜度, 基于数据集的特征参考量 E_0 和 E_2 构建了一个基于规则的决策树模型, 用于自动选择合适的分区方式. 具体决策路径如图 4 所示.

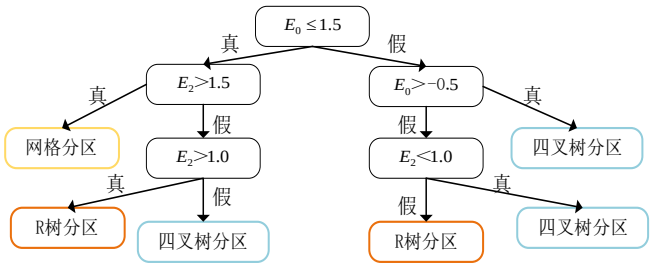


图 4 决策树模型示意图

AI 优化分区构建. DRL-Partitioner[28]是一种通过 AI 优化网格分区方案，它将空间数据分区优化问题建模为一个深度强化学习任务. 在该框架中，状态空间 s_t 被定义为 t 时刻当前分区方案的配置及底层数据的分布. 对于网格中的每个单元 (i, j)，动作 a_t ={x, y, *direction*}表示智能体在时刻 t 选择执行的操作，即在网格(x, y)中添加一条分割线，*direction* 指示边界线的延伸方向(right 表示水平向右，down 表示垂直向下)；奖励是根据当前分区方案执行工作负载的总运行时间评估. 图 5 展示 DRL-Partitioner 的运行示例：t=0 时，s(2,3)={0,0,0.2}表明网格单元(2,3)的顶部无水平边界(即 h=0)，左侧无垂直边界(即 v=0)，单元格(2,3)属于初始分区 P_1. 动作 a_1= {0,3,down}表示在(0,3) 添加向下垂直边界，由于单元格(2,3)所属分区缩小，P(2,3)减小，状态更新为 s(2,3)={0,1,0.1}. 动作 a_2= {2,3,right}表明在单元格(2,3)添加向右水平边界，单元格(2,3)的状态更新为 s(2,3)={1,1,0.2}. 上述过程不断重复，直至智能体生成了预设数量的分区或达到预设的训练步数，以期找到使执行工作负载最小的分区方案.

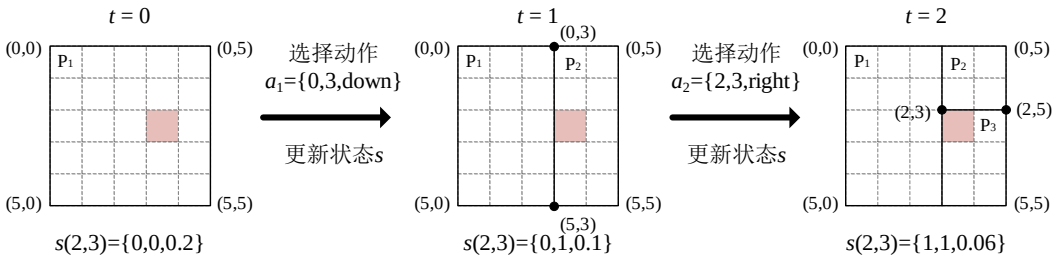


图 5 DRL-Partitioner 的运行示例[28]

3.1.2 动态独立数据

动态独立数据具有时间属性，数据量通常较大，且多采用分布式存储方式. 针对此类数据的分区构建，基于 AI 的方法主要侧重于参数自动调优与结构增量维护两个层面.

AI 优化分区构建. ST-*k*NNJ[29]是一个基于 Spark 的分布式框架，旨在解决“时空 k 近邻连接”问题. 该框架提出了一种结合扫描线算法(时间划分)与四叉树(空间划分)的时空分区方法. ST-*k*NNJ 利用贝叶斯优化器配合预测模型，能够快速搜索出最优参数组合，并利用增量观测数据周期性更新预测器. 为应对动态独立数据的持续更新，Dist[30]设计了一种基于图论的分区策略，将时空立方体建模为加权图，利用改进的 KL 算法兼顾负载均衡与簇内聚. 在参数优化层面，DiST 提出了基于 XGBoost 的成对排序学习框架评估候选参数对的相对优劣，能够更高效地联合优化分区粒度等算法参数与 Spark 系统配置，从而克服传统方法在高维参数空间下的搜索瓶颈. 为应对独立数据的持续更新，TIN VU 等人提出了基于成本的增量分区框架 CBP[31]，该框架引入基于线性回归的学习型成本模型，通过拟合少量的查询样本数据预测不同分区状态下的范围查询耗时. 基于模型的预测结果，CPB 利用贪婪算法计算重组收益，智能选择对查询性能影响最大的“劣质”分区并局部重组. 该方法实现了在动态数据流下的分区质量自适应优化.

3.1.3 静态关联数据

静态关联数据是与其他数据元素存在拓扑关系但不产生新的读数的数据. 处理这类数据时, 除了考虑空间分布, 还需特别关注数据点之间的空间关联性, 以优化涉及拓扑关系的复杂操作的性能. 由于模型直接代替分区这种端到端模型生成的结果容易破坏原有数据的拓扑一致性(如邻接、连通性等语义信息), 静态关联数据尚缺乏模型直接代替分区的研究.

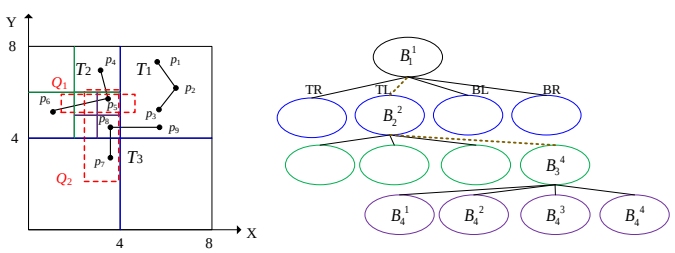
AI 辅助分区选择. 受 SA-Partitioner[26]的启发, TIN VU 等人通过训练全连接模型 DL-Partitioner[32]提出了更智能的 AI 辅助分区选择方法. DL-Partitioner 分为训练和应用两个阶段: (1) 在离线训练阶段中, DL-Partitioner 分别采用直方图和基于分形的方法对数据集进行汇总. 基于分形的汇总方法使用盒函数[27]和莫兰指数[33]来描述数据集的复杂性和分布特征, 相较于 SA-Partitioner[26]仅依赖盒计数函数来评估偏斜度, DL-Partitioner 的特征向量选择更为丰富和全面. DL-Partitioner 使用多维度衡量其分区质量, 如总面积、总边距、总面积重叠以及分区基数的标准差. (2) 在应用阶段中, 其利用训练好的模型预测新数据集的最佳分区技术, 模型的输出对应 Kd 树、R*-Grove、STR、Z 曲线、网格和 RR*树中的最佳分区模型. 相较于 SA-Partitioner[26]基于规则的决策树模型, DL-Partitioner 能更灵活地适应各种复杂的数据分布, 避免了人工设定规则可能带来的局限性. 该技术成功应用于开源的大规模时空数据探索性分析系统 Beast[34], 证明了其在处理实际复杂空间数据时的高效性和鲁棒性.

3.1.4 动态关联数据

动态关联数据是指在时间维度上持续演化, 且数据间存在空间或语义关联的数据. 在对该数据进行分区时, 关键在于保留数据间的时空相似性. 基于学习的数据简化技术能有效降低动态关联数据存储与计算开销.

模型代替分区. STTC[35]是一种面向轨迹的聚类算法, 旨在提取时空特征、快速发现相似模式并自动确定聚类数量. 其核心包含两阶段过程: (1) 基于速度-空间约束的轨迹过滤. 首先, 计算每条轨迹段的欧式距离及平均速度. 通过设定速度阈值剔除平均速度异常的轨迹段, 符合阈值条件的轨迹段则被存入处理队列. (2) 基于空间邻近度的聚类生成. 从队列中依次取出轨迹段, 依据欧式距离搜索空间上与之相似的轨迹段形成新的聚类. 最终, 每个聚类都包含一组空间形态相似的轨迹段, 记录了其原始轨迹 ID 及对应的数据点信息.

基于学习的时空数据简化. RL4QDTS[15]是基于多智能体强化学习的轨迹数据简化方法, 其主要目标是让给定查询集在原始数据集 D 上的查询结果与简化数据集 D' 上的查询结果一致. 该方法在 D 上构建一个八叉树, 将三维时空(二维空间与一维时间)递归地均匀划分为八个子节点, Agent-Cube 负责在八叉树中进行导航式决策; Agent-Point 负责从 Agent-Cube 传递的数据点中选择一个加入 D'. 图 6 以四叉树为例简化阐述了此过程. 图 6(a)中的原始数据库 D 含轨迹 T_1、T_2 和 T_3 及查询 Q_1 和 Q_2, 查询结果分别为 $Q_1(D)=T_2$, $Q_2(D)=<T_2,T_3>$. 当存储预算为 7 时, 由于初始化过程插入了三条轨迹的起止点, 剩余存储预算为 1. 图 6(b)的四叉树节点被标记为 TR、TL、BL 和 BR(右上、左上、左下和右下). Agent-Cube 沿着虚线遍历节点, 选择子节点. Agent-Point 将评估该子节点内各轨迹候选点(如 T_2 的点 p_5 和 T_3 的点 p_8)的时空特征值, 并选择时空特征值最大的点加入 D', 在图例中时空特征值最大的点为 p_5, 经过该过程后, 存储预算用尽, 点插入过程停止, 简化后的数据库查询结果为 $Q_1(D')=T_2$, $Q_2(D')=<T_2,T_3>$, 与在 D 中的查询结果一致.



(a) 原始数据库D (b) 四叉树

图 6 RL4QDTS 的运行示例[15]

针对时序数据等动态关联数据，kD-STR[36] 提出了一种基于分层分区技术的摘要生成方法. 该方法通过在时空维度识别具有相似特征的实例区域，并对各区域内的实例进行建模，最终生成紧凑的数据摘要. kD-STR的工作流程可分为三个主要步骤: (1) 数据分区: 使用Voronoi图对传感器所在的空间进行划分. 随后，它将具有相同聚类标签的时空邻近实例组合成同质区域. 上述过程会生成一个分层分区树. (2) 区域建模: 为每个识别出的特征聚类内的实例拟合一个数学模型. 模型的输入是时空坐标，输出是预测的特征值. (3) 迭代规约: 在每次迭代中，算法会评估两种选择(增加现有模型的复杂度或依据分区树增加区域数量)，并选择能够最大程度优化目标函数的操作. 当增加模型复杂度或区域数量都无法进一步优化目标函数时，算法停止迭代，并输出最终的区域和模型集合.

3.1.5 总结

本小节对基于学习的时空数据存储方法进行了回顾，具体总结如表 1 所示.

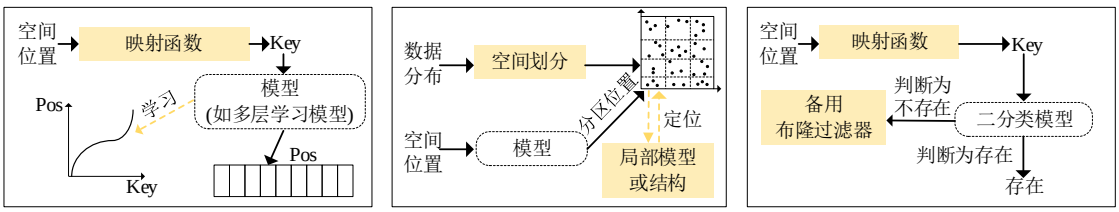
表 1 基于学习的数据存储研究总结

数据类型	研究内容	名称	模型	是否工作负载感知
静态独立数据	模型代替分区	ClusterPar[24]	K-Means	否
	AI辅助分区选择	SA-Partitioner[26]	决策树	否
	AI优化分区构建	DRL-Partitioner[28]	深度强化学习	是
动态独立数据	AI优化分区构建	ST-kNNJ[27]	贝叶斯优化	是
		Dist[30]	排序学习框架	是
		CPB[31]	线性回归	是
静态关联数据	AI辅助分区选择	DL-Partitioner[32]	全连接模型	否
动态关联数据	模型代替分区	STTC[35]	K-Means	否
	数据简化	RL4QDTS[15]	强化学习	是
		kD-STR[36]	层次聚类	否

3.2 基于学习的时空数据索引

基于学习的数据索引主要分为两类: **学习索引**与 **AI 辅助索引**. Kraska 等人[16]首次提出利用模型替代传统索引结构(如 B 树、哈希索引和布隆过滤器等)，这类索引被称为“学习索引”. 在数据分布相对稳定的前提下，模型能较为精准地学习并预测数据位置，从而提高查询速度并显著降低内存占用. 然而，如文献[37][38]所述，在面对频繁更新的动态数据时，模型可能失效；相较之下，AI 辅助索引保留了传统索引的核心结构，同时借助 AI 技术优化构建过程或提升查询效率，在处理复杂动态数据集时适应性更强.

如图 7 所示，学习索引可进一步细分为以下三类: (1) **基于累积分布函数(Cumulative Distribution Function, CDF)的学习索引.** 其核心思想是利用 CDF 预测数据位置以加速检索. 针对高维的时空数据，首先通过映射函数将其降维到一维，再采用分层模型架构学习数据分布与物理存储间的映射关系. (2) **基于网格分区的学习索引.** 将数据划分为若干个局部区域，利用学习模型建立从空间位置到分区的预测映射，并在每个分区内部建立局部模型或结构以实现数据点的快速定位. (3) **基于布隆过滤器的学习索引.** 将布隆过滤的任务视为二分类的概率预测问题，并辅以一个小型备用布隆过滤器以消除漏报.

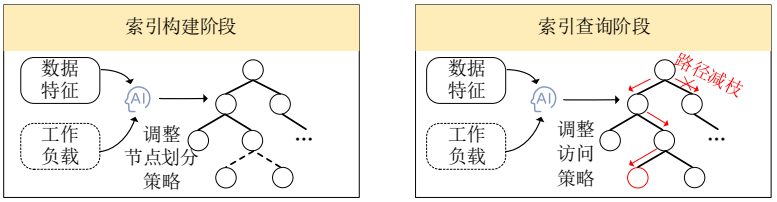


(a) 基于CDF的学习索引 (b) 基于网格分区的学习索引 (c) 基于布隆过滤器的学习索引

图 7 学习索引分类图

如图 8 所示, AI 辅助索引可以进一步细分为以下两类: (1) **索引构建优化.** 即利用 AI 模型优化索引的

划分策略、结构设计或参数配置. (2) **查询感知索引.** 在查询过程中引入 AI 决策模块, 实现候选过滤、路径剪枝或目标预测等优化.



(a) 索引构建优化 (b) 查询感知索引

图 8 AI 辅助索引分类图

基于本小节调研的文献汇总, 目前缺乏针对静态关联数据的基于学习的索引结构研究. 这是由于基于静态关联数据的路径规划等查询依赖数据的拓扑结构, 主要依靠图算法解决. 下面将分别介绍不同时空数据类型下基于学习的时空索引方法.

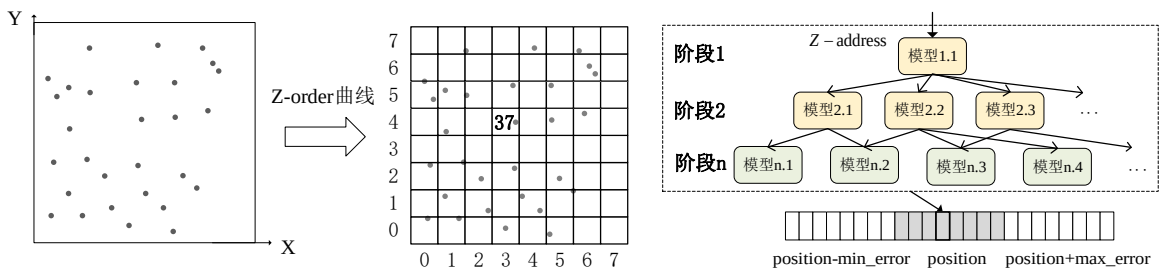
3.2.1 静态独立数据

静态独立数据具有查询频率高和更新频率低的特点, 索引设计的核心是最大化查询效率, 且需要高效支持点、范围及 KNN 查询等关键空间操作, 以满足如地图服务等真实应用中的检索需求.

基于 CDF 的学习索引. 以 ZM 索引[39]为例, 它采用 Z-order 曲线将多维数据映射到一维有序整数值 Z-address, 然后构建多阶段模型建立 Z-address 和有序数组中位置(position)的映射关系. 对于一个二维空间中的点(x,y), 其 Z-address 通过将 x 和 y 的二进制表示进行交叉编码. 例如当 y=4, x=3 时, 其二进制表示分别为 100 和 011, 则其 Z-address 为 10-01-01, 对应的十进制为 37. 这种映射方式保证了空间数据的大小关系. 在多阶段模型中, 每个阶段包含一个或多个模型, 每个模型都是一个人工神经网络或线性模型. 如图 9 示, 模型根据输入的 Z-address 给出查询健的初步估计 $p\hat{o}s$, 通过公式(1)确定模型索引 i, 直到最后一个阶段估计出 Z-address 在有序数组中的预测位置. 通过训练过程对每个键执行模型并记录最小误差 min_error(最差的过度预测)和最大误差 max_error(最差的不足预测), 可以保证: 若某个查找键存在, 则其位置必定在范围[position−min_error, position+max_error]之内, 接着可以通过二分查找或其变体来找到查询键的真实位置.

$$i = \frac{M \cdot p\hat{o}s}{N} \tag{1}$$

其中, M 是下一阶段的模型数量, N 是数据集的总大小, $p\hat{o}s$ 是上一阶段模型的输出.



(a) 空间数据 (b) Z-address (c) 分阶段模型

图 9 ZM 索引结构

不均匀空间数据分布对基于空间填充曲线的索引结构会产生负面影响, 为此, RSMI[40]引入了 R 树的批量加载技术[41], 将数据点映射到秩空间中, 并在秩空间中利用 Z-order 曲线对数据点重新排序, 使得数据点之间的曲线值间隙更加均匀. ML 索引[42]通过缩放方法将彼此相近的点分组在一起, 并将它们投影在一维中, 以便通过一维值的接近度来保留相似性. TBSI[43]引入 Transformer 架构, 利用自注意力机制自动捕捉复杂

空间分布中的全局依赖关系，从而在无需分层结构的情况下实现对不均匀数据分布的高效检索.

基于网格分区的学习索引. SPRIG[44]、DGLSI[45]和 LISA[46]分别是基于内存和磁盘的索引. 以 SPRIG 为例，图 10 详细展示了其索引构建和查询处理的过程. 索引构建阶段包含以下三个步骤: (1) 空间划分. SPRIG 将数据空间划分为 $n\times m$ 的网格布局，并通过查询感知的成本模型决定 n 和 m 的取值，确保每列的数据点大致相等. (2) 网格 ID 映射. SPRIG 利用空间插值函数拟合网格中的数据点，将二维空间位置映射到一维的网格 ID，用于支持空间数据的快速定位. (3) 局部定位. SPRIG 维护了内部表 T，其每个条目都包含一个指向单元格中第一条记录的指针以及该单元格中的记录数量. 在查询处理阶段，针对范围查询，SPRIG 首先使用空间插值函数预测查询点所在的单元格 ID，并在预测的单元格及其邻近单元格中进行局部二分搜索以确定实际位置. 针对 *k*NN 查询，SPRIG 的处理流程如下: (1) 映射. 使用空间插值函数定位起始单元格，并逐步扩展至相邻单元格. (2) 选择枢纽. 采用基于枢轴的过滤技术与最近点剪枝，减少需检查的记录与单元格. (3) 排序. 借助优先队列动态维护 k 个最近邻点，直到找到足够的最近邻点.

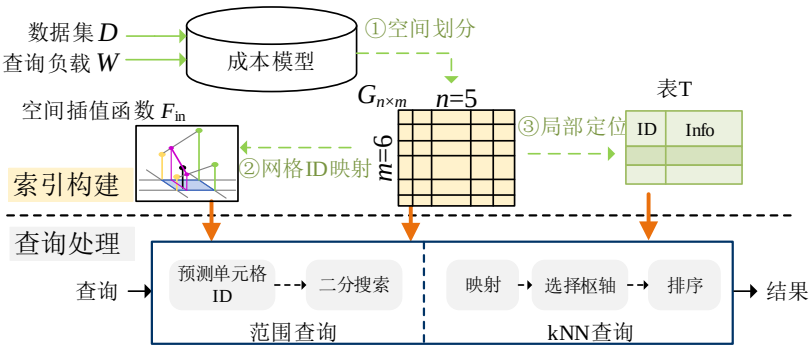


图 10 SPRIG 索引流程[44]

基于布隆过滤器的学习索引. 以 PA-LBF[47]为例，如图 11 所示，其结构由以下三部分组成: (1) 基于前缀的分类. PA-LBF 首先使用 Z-order 曲线将多维空间数据映射至一维，随后提取数据前缀并将其划分成 N 类. (2) 适应性学习过程. 针对每类数据，PA-LBF 构建自适应的子学习型布隆过滤器(Learned Bloom Filter, LBF)，用于对数据的后缀进行训练. 根据数据分布的复杂性不同，子 LBF 的层数 k 可动态调整. (3) 备用滤波器. PA-LBF 采用计数布隆过滤器(Counting Bloom Filter, CBF)作为备用滤波器，构建了整体 CBF 和不相交 CBF，以适应更新操作不频繁与频繁两种场景. 查询过程为: 如果查询项的前缀与已知前缀匹配，则将查询项发送到相应的子 LBF 进行处理. 子 LBF 对后缀进行分类，判断查询项是否存在. 如果子 LBF 返回“存在”，则直接返回结果；如果返回“不存在”，则将查询项输入到 CBF 中进行进一步检验.

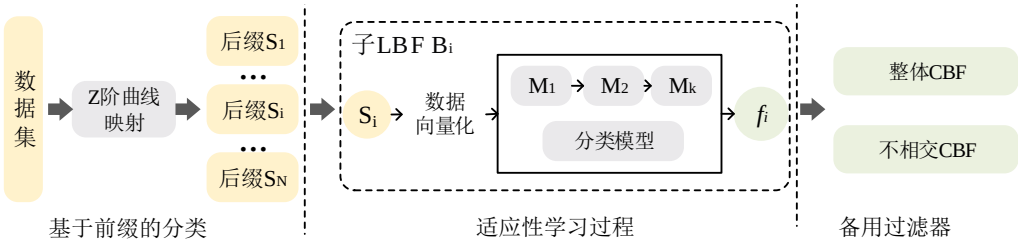


图 11 PA-LBF 索引结构[47]

索引构建优化. WISK[48]通过学习历史查询的工作负载来优化数据布局和构建层次结构，以期最小化空间关键词查询的处理成本. WISK 的工作流程主要分为两个阶段: (1) 分区优化: WISK 旨在学习给定查询工作负载的最佳数据布局，以最小化查询处理成本. 它首先训练机器学习模型来近似地理文本对象的 CDF. 基于学习到的 CDF 定义一个可微分的成本估计函数，并利用随机梯度下降等技术优化数据分区，从而生成最

小化查询处理成本的底层簇. (2) 层次结构设计. 底部簇被输入到一个基于强化学习的算法中, 该算法通过自底向上的方式将簇打包构建成一个分层索引结构, 以进一步提高查询时的剪枝效率. 针对 WISK 在查询分布发生变化时性能下降的问题, RLSKI[49]提出了一种基于多目标强化学习的数据驱动索引构建框架. 与 WISK 依赖历史工作负载不同, RLSKI 更侧重从数据分布特征中直接学习最优的分区策略. RLR 树[50]保留了 R 树的原始数据结构与查询算法, 通过建模两个独立的马尔可夫决策过程, 分别优化 R 树在构建与更新阶段的子树选择策略和节点分裂策略. 相较于传统的R树及其变体, 当持续插入同分布的数据时, RLR树的性能优势会随着数据量的增加而愈加显著.

查询感知索引. AI+R 树[51]通过机器学习模型感知并优化查询访问效率, 旨在减少 R 树在范围查询中因节点重叠而导致的无关叶节点访问. 该方法引入“重叠比”量化该问题, 即使用包含结果的叶节点数量除以访问的叶节点总数来估计查询的访问效率. 以图 12 为例, 处理查询 Q 时, R 树同时搜索 N_5 和 N_6, 但结果仅存在于 N_5 中, 故重叠比为 0.5. AI+R 树利用二分类模型判断查询 Q 的访问效率: 若效率较低, 则启用 AI 树, 通过多标签分类器直接预测相关叶节点 ID, 然后仅访问这些节点; 反之则通过 R 树确定访问节点.

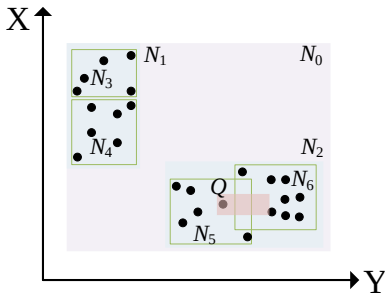


图 12 具有重叠节点的 R 树示例

3.2.2 动态独立数据

动态独立数据更新频繁, 此类数据的索引设计需要同时兼顾高效的数据查询和低成本的维护. AI 辅助索引相关的研究主要集中在如何高效构建索引以应对数据的动态更新.

索引构建优化. Waffle[52]将地理空间划分为固定大小的单元, 并将相邻单元组合成块, 作为内存中的基本分配单元. 该设计有效利用了访问的空间局部性, 显著提升了缓存命中率, 从而提高了查询性能. Waffle索引的性能高度依赖于其配置参数, 因此引入了基于强化学习的在线调整系统 Waffle Maker, 用于动态选择最优的配置参数, 并决定何时触发索引重建. 进一步地, HELM[53]引入多智能体强化学习机制. 通过多个智能体之间的协作实时监控空间数据的动态演变, 在线自动寻找最优的分区边界和配置参数, 并保证在索引优化和重构过程中, 查询和更新操作仍能高性能运行.

3.2.3 动态关联数据

动态关联数据的查询需求远比静态数据复杂, 典型的如未来位置预测查询和相似性查询等. 针对这些特定查询, 基于学习的索引设计必须与查询处理过程深度融合.

索引构建优化. 针对轨迹数据的范围查询, BT 树[54]利用强化学习, 根据轨迹数据的内在特征与历史查询负载, 灵活选择经度、纬度或时间等维度划分数据空间. 针对未来位置预测查询, NEIST[55]利用循环神经网络(Recurrent Neural Network, RNN)精确建模移动对象的轨迹演化过程, 以提升查询效率. 如图 13 所示, NEIST 采取了以下方案改进: (1) 轨迹预测模型. 采用 Seq2Seq 模型学习移动对象的历史轨迹, 并预测其未来位置. (2) 轨迹后缀树索引. 为了降低大规模预测的开销, NEIST 构建了一个后缀树来索引具有相似后缀的轨迹, 并采用局部敏感哈希对轨迹进行聚类, 从而使聚类内的轨迹能够共享相似的预测结果, 大幅减少了重复计算. (3) 基于时间槽的 TPR 树(TS-TPR). TS-TPR 将 NEIST 预测的轨迹数据按时间槽组织, 将具有相似未来路径的对象进行分组, 并利用时间槽内的线性预测辅助查询, 从而更高效地支持对未来时空数据的范围查询.

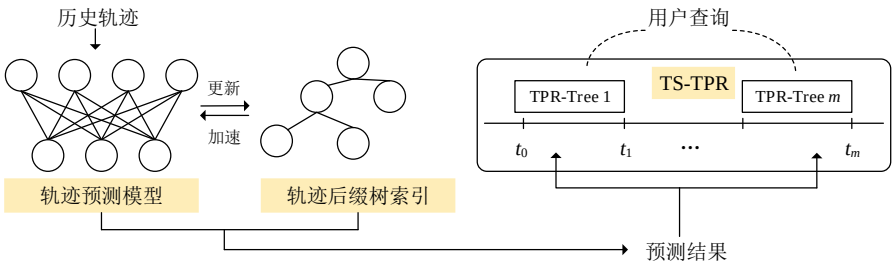


图 13 NEIST 框架[55]

查询感知索引. X-FIST[57]旨在加速大规模轨迹数据相似性查询, 它通过扩展 Flood[59]索引实现, Flood 核心机制是利用分段线性模型[59]预测多维数据在有序数组中的位置. X-FIST 为每个完整轨迹及其子轨迹计算最小边界矩形(Minimum Bounding Rectangle, MBR), 在此基础上构建 Flood MBR 和 Flood MBR-Sub, 分别用来索引完整轨迹的 MBR 和子轨迹的 MBR. 查询时 X-FIST 的处理流程如下: (1) 轨迹级剪枝. 利用 Flood MBR 索引, 快速排除与查询轨迹 MBR 不相交或不被查询轨迹 MBR 严格覆盖的轨迹. (2) 子轨迹级剪枝: 通过 Flood MBR-Sub 索引, 基于查询子轨迹与候选子轨迹 MBR 间的相交判断, 进一步缩小候选轨迹的范围. (3) 精确验证. X-FIST 采用动态规划技术对剩余的候选轨迹进行最终的相似性验证. Cymo[60]将时空空间划分为多个子空间, 利用历史工作负载, 结合 CNN 和 LSTM 预测每个子空间的查询模式频率, 并根据预测结果为每个子空间从“时间优先”“空间优先”或“多索引”中选择最优索引.

3.2.4 总结

本小节对基于学习的时空数据索引进行了回顾, 具体总结如表 2 所示.

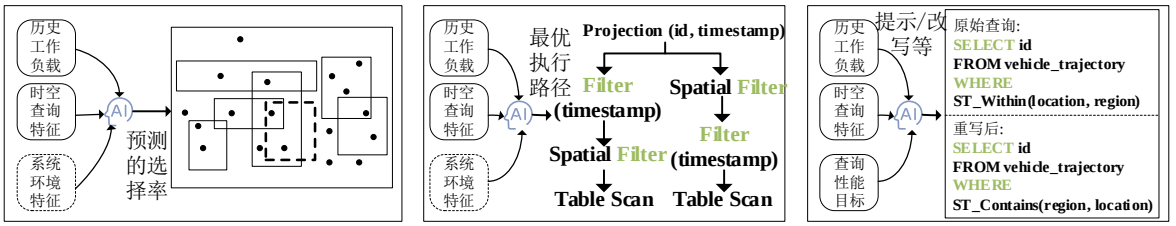
表 2 基于学习的数据索引研究总结

数据类型	研究分类	索引类型	名称	查询类型	模型	是否设计更新机制
静态独立数据	学习索引	基于 CDF	ZM[39]	点查询 范围查询	多阶段模型	否
			RSMI[40]		多层感知机	是
			ML[42]		递归回归模型	否
			TBSI[41]		Transformer	否
		基于网格分区	SPRIG[44]	范围查询 kNN查询	空间插值函数	否
			LISA[46]		分片预测函数 格子回归模型	是
		基于布隆过滤器	PA-LBF[47]	存在性查询	梯度提升	是
	AI 辅助索引	索引构建优化	WISK[48]	空间关键词查询	强化学习	是
			RLR 树[50]	范围查询 KNN查询	强化学习	否
		查询感知索引	AI+R 树[51]	范围查询	多标签分类器 二元分类器	R树 可更新
动态独立数据	AI 辅助索引	索引构建优化	Waffle[52] HELM[53]	范围查询 kNN 查询	强化学习	是
动态关联数据	AI 辅助索引	索引构建优化	BT 树[54]	范围查询	强化学习	否
			NEIST[55]	时空范围查询	RNN模型	否
		查询感知索引	X-FIST[57] Cymo[60]	相似性查询	分段线性模型	否

3.3 基于学习的时空查询优化

传统的查询优化器依赖于“计划枚举-基数估计-成本模型”这一经典范式. 近年来, 研究人员开始探索利用机器学习技术来革新时空查询优化, 通过数据驱动的方式学习复杂的时空数据分布与查询行为规律[58]. 本小节综述基于学习的时空查询优化技术, 依据核心优化任务将其分为以下三类: (1) **参数估计**. 该类研究致力

于精准预测选择率、基数和成本等核心指标, 通常将参数估计建模为回归问题. 如图 14(a)所示, 以选择率估计为例, 模型输入包括历史工作负载中的查询及其对应的选择率, 以及当前查询的特征信息, 模型输出为[0,1]范围内的实数, 表示预测的选择率. (2) **执行计划选择.** 该类方法旨在为查询直接推荐最优的执行路径或策略. 如图 14(b)所示, 其核心思想是从历史工作负载中学习查询与其最优执行计划之间的映射关系, 当新查询到来时, 模型提取其特征, 并基于已学到的模式推荐一个高效的执行路径. 由于不同系统环境可能导致执行效果差异, 模型通常也会考虑系统环境的相关特征. (3) **查询重写.** 该类研究通过智能地改写查询语句本身或改写特定的约束. 与前两者不同, 它不改变底层的执行引擎逻辑. 如图 14(c)所示, ST_Contains(A, B)与ST_Within(B, A)在语义上等价, 但特定数据库可能对其中一个有专门的优化, 学习模型可以基于对数据库特性和工作负载的理解, 自动将查询重写为性能更优的等价形式.



(a) 参数估计 (b) 算法/执行计划选择 (c) 查询重写

图 14 基于学习的时空查询优化方法

3.3.1 静态独立数据

针对静态独立数据, 由于其分布相对稳定, 基于学习的查询优化研究重点在于通过离线学习提升复杂空间计算的效率.

参数估计: PivNet[61]旨在快速且准确地估计查询点到其k个最近邻点的距离(即 k-NN 距离). 对于大规模数据集, 直接计算 k-NN 距离需要大量的磁盘或内存访问. PivNet 的核心思想基于预设的“枢轴点”: 一个查询点与其邻近枢轴点的 k-NN 距离具有高度相关性. 模型能以 O(1)的复杂度一次性输出完整的 k-NN 距离向量, 实现了无需访问原始数据的高效估计. 为增强模型的泛化能力, Spatial Embedding[62]引入了空间嵌入的概念以及两个机器学习模型(M1 和 M2): (1) M1 是一个无监督模型, 旨在捕捉空间数据集的内在特征, 生成一个紧凑、信息丰富的向量表示. M1 只需训练一次. 并采用大量合成数据与真实世界数据进行训练, 因而具有良好的泛化能力. (2) M2 是一个有监督的轻量级模型. 它的输入为 M1 生成的空间嵌入, 并结合与任务相关的简单参数(如查询窗口的坐标). 由于空间嵌入已经包含了数据集的大部分复杂特征, M2 仅需极少量的训练数据点即可达到较高精度, 从而显著降低了为新任务构建模型的成本和时间.

3.3.2 动态独立数据

动态独立数据的特点是数据分布和查询负载都具有高度的动态性和时效性, 基于学习的查询优化研究侧重于提升模型的增量学习能力和自适应能力.

参数估计: 在处理时空文本数据流(如用户签到信息)时, 数据和查询负载的高度动态性给选择率估计带来了较大挑战. LATEST[63]在移动时间窗口上构建基于 Hoeffding 树(快速的决策树算法)[64]的增量学习模型, 辅助底层系统在多个估计结构之间切换, 以维持估计的高准确性.

执行计划选择: Breslin[65]等人提出了基于学习的查询优化器 STOP, 其核心思想是通过识别与当前查询相似的历史查询, 推荐已知的最优执行计划. 为准确度量查询的相似性, 作者设计了一个特征提取框架, 从三个维度来描述一个时空查询: (1) 代数特征. 将查询转换为代数表达式树, 并统计各类操作符(如 join, project 等)的出现频率. (2) 时空图模式特征. 将查询中的空间和时间部分视为图中具有更高权重的特殊边与节点, 以更精确地捕捉时空查询的结构特征. (3) 查询基数特征. 估算经空间过滤与语义过滤后可能返回的结果数.

查询重写: 时空可视化查询中, 为了提供流畅的交互体验, 系统必须近乎实时地返回结果(通常要求响应时间在 500 毫秒以内). Maliva[66]是新型中间件系统, 旨在解决时空可视化领域关键挑战: 确保用户的查询请

求在严格的时间限制内得到响应以维持系统的交互性. Maliva 位于用户前端和后端数据库之间, 当收到用户的查询请求时并不直接将其发给数据库, 而是智能地重写该查询, 以寻找可快速执行的版本. Maliva 利用马尔可夫决策过程训练出一个智能体, 使其学会在不花费过多决策时间的前提下, 有效探索各类查询重写方案(如添加数据库提示或使用数据采样).

3.3.3 静态关联数据

静态关联数据对象之间存在较稳定的拓扑关系, 最典型的场景是空间连接查询, 这类数据的查询优化挑战在于准确估计连接操作的基数和成本, 因为连接操作的计算复杂度极高, 微小的估计误差都可能导致执行计划的选择出现巨大偏差.

参数估计: NNSSE[67]是基于神经网络的选择率估计方法. 其核心思想是为每个用户自定义函数训练一个专属的神经网络, 以学习查询参数的特征向量与实际选择率之间的映射关系. 例如, 对于一个圆对象, 其特征向量可由圆心坐标和半径构成. 研究以 contains(包含)、intersectIn(相交)和 overlap(重叠)三个空间函数为例, 验证了该方法的有效性. PolyCard[68]针对复杂多边形的相交查询, 通过自适应采样技术将变长顶点转换为固定维度的几何特征向量, 并结合合成数据增强策略训练轻量级 MLP 模型, 实现了微秒级的空间基数估计.

3.3.4 动态关联数据

动态关联数据是时空数据中最复杂的类型, 兼具数据与查询的动态性以及对象间的复杂关联, 典型任务是轨迹的相似性查询与连接. 基于学习的查询优化研究重点在于设计模型来应对数据的稀疏性和高维性.

参数估计: GL+CNN[69]旨在提升轨迹相似性查询与相似性连接中基数估计(即估计两个轨迹集合间相似轨迹的数量)的准确性和效率, 并缓解训练样本不足时估计偏差较大的问题. 其核心包含两大增强策略: 查询切片和数据切片, 并将二者整合在一个全局-局部模型框架中. 查询切片将查询轨迹的特征向量按空间、时间与文本三个维度分割成三个切片, 分别学习各维度切片的嵌入表示后再进行融合; 数据切片将整个轨迹数据集通过聚类划分成多个互不重叠的“数据切片”, 并为每个数据切片单独训练一个局部模型, 最终的总基数是所有局部模型估计结果的总和. 尽管数据切片能提升精度, 但若为每个查询逐一运行所有局部模型, 将导致显著的计算开销. 为此, GL+CNN 设计全局模型, 其任务是在给定查询时快速预测哪些数据切片最可能包含相似轨迹, 仅当被全局模型选中的局部模型才会被激活并执行估计, 从而在保证精度的同时提升效率.

3.3.5 总结

本小节对基于学习的时空查询优化进行了回顾, 具体总结如表 3 所示.

表 3 基于学习的数据查询研究总结

数据类型	研究内容	名称	查询类型	模型	是否感知查询负载
静态独立数据	参数估计	PivNet[61]	kNN 查询	全连接神经网络	是
		Spatial embedding[62]	多种查询	自编码器+ DNN/CNN	否
动态独立数据	参数估计	LATEST[63]	范围查询	Hoeffding树(增量学习)	否
	执行计划选择	STOP[65]	时空SPARQL查询	SVM/kNN/随机森林	是
	查询重写	Maliva[66]	时空可视化查询	强化学习	是
静态关联数据	参数估计	NNSSE[67]、PolyCard[66]	用户自定义函数	神经网络	否
动态关联数据	参数估计	GL+CNN[69]	轨迹相似性查询	K-means、DNN	是

4 ST-DM4AI

如图 15 所示, 尽管时空 AI 的任务场景多样, 但其核心流程可归纳为数据准备、模型训练和模型使用三个阶段. 在数据准备阶段, 原始异构的时空数据通过清洗、过滤和转换据转化为适合 AI 模型使用的高质量、结构化输入. 在模型训练阶段, 基于处理后的时空数据, 模型通过空间学习与时间学习模块分别捕捉数据中复杂的空间相关性和时间依赖性. 在模型推理阶段, 在模型使用阶段, 用户通过调用已训练的模型进行预测或决策, 模型推理平台接收输入数据进行计算, 并将结果返回给用户.

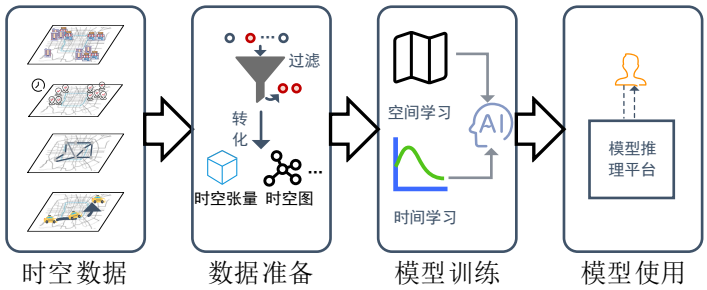


图 15 STAI 模型三阶段示意图

4.1 基于STDM的时空数据准备

根据 Anaconda 的调查[70], 数据分析师约 37.75%的努力投入于数据准备, 而时空数据的数据准备更为复杂. 尽管已有研究致力于构建面对特定应用的时空数据集, 如基于高分辨率卫星图像创建深度学习数据集[71], 但对广大研究者而言, 目前仍缺乏能将现实世界时空数据高效转化为 STAI 模型输入的标准化途径. 如图 16 所示, 在传统的数据准备模式中, 开发者通常需要编写大量定制化的 Python 或 R 脚本, 将数据从存储系统中提取到外部环境进行清洗. 这种模式不仅导致了巨大的编程维护成本, 更存在高昂的I/O开销. 基于STDM的数据准备在数据管理平台内部就完成数据清理, 输出高质量的时空数据. STDM 在数据准备阶段的突出贡献在于确保时空数据的**准确性**和**实时性**. 其中, 准确性是各类时空数据的基础要求, 而实时性保障则主要针对具有流式特征的动态数据, 静态数据因其时不变特性通常仅聚焦于精确性校验.

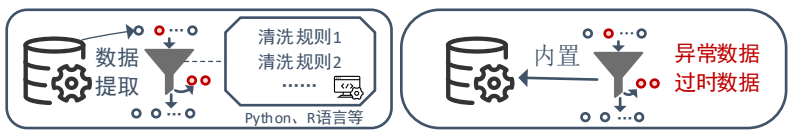


图 16 传统的时空数据准备与基于 STDM 的数据准备对比

4.1.1 静态独立数据

时空数据准确性: 静态独立数据的数据准备流程的核心是确保其空间位置和属性的精确性. 传统数据清洗规则通常要求属性值间存在足够的“共同出现”频次以学习有效依赖关系. 然而, 对于以精确坐标为核心的静态独立数据, 记录之间极少拥有完全相同坐标, 导致“共同出现”现象极为罕见, 致使传统方法难以有效识别与修正错误. 为此, SPARCLE 框架[72]将“空间感知”注入到基于规则的数据清洗系统核心引擎中. 其目标通过引入“空间邻域”和“距离加权”得以实现: 前者放宽了传统意义上“共同出现”的条件, 允许在一定空间邻近度内的记录被视为“共同出现”, 而非仅仅要求精确数值匹配; 后者则根据记录的相对距离, 为其满足依赖规则的程度赋予不同的权重.

4.1.2 动态独立数据

时空数据准确性: 针对动态时空数据的流式特性, 传统离线清洗的高延迟不再适用. Kaminsky 等人[73]提出的 WEEVER 算法通过增量检测否定约束, 实现了对新插入数据的即时质量校验, 避免了全量重算的计算开销. 在处理涉及时间序列或数值属性的不等式约束时, WEEVER 引入了 LT 树索引结构, 极大优化了动态场景下不等式谓词的验证效率.

时空数据实时性: 传统的数据流管理系统通常需要重启查询才能处理移动对象, 导致系统响应慢、结果过时等问题, SPEAR[74]在分布式环境中实现对动态独立数据的实时管理. SPEAR 定义了了分布式流式时空数据类型 $\text{StreamingST}_{(n)}$和相应函数, 可直接处理分布式查询状态的实时变化(如时空范围查询和最近邻查询), 并提出基于 GeoHash 的动态空间分区策略, 能够实时根据数据负载调整 GeoHash 精度, 有效缓解分布式环境中的数据倾斜问题, 实现高效并行处理. 当查询对象发生移动或改变状态时, SPEAR 能够保持持续的高处理速率, 为下游的 STAI 任务提供实时数据支持.

4.1.3 静态关联数据

时空数据准确性: QARTA[75]是一个构建高精度路网(包括空间拓扑和边权重)的系统. 构建高精度路网拓扑和边权重的算法通常基于默认路网数据的绝对准确的假设, 但现实场景中静态数据本身可能也存在误差. 当轨迹与路网不重合时, 传统算法难以判断 GPS 数据和地图数据哪个更准确. QARTA 系统使用随机森林分类器来判断轨迹点和路网段的准确性: 若路网更准, 则执行地图匹配; 若路网缺失或错误, 则触发地图更新.

4.1.4 动态关联数据

时空数据准确性: Clean4TSDB[76]是专为 Apache IoTDB 等时间序列数据库设计的数据清理系统. Clean4TSDB 构建了"画像-检测-修复"清理流程, 其核心模块如下: (1) 数据质量约束发现: 系统首先从数据中自动学习和发现数据质量规则, 引入 TSDD 新型约束, 可捕捉多维时间序列数据之间复杂的上下文和数值关系. (2) 违规模式检测: 利用发现的约束, 系统可以检测多种复杂的错误模式, 如尖峰错误、持续性错误、集体性错误和非平稳错误等, 并能量化数据违规的程度. (3) 多维时间序列修复: 系统将数据修复问题构建为一个优化问题, 目标是在满足所有数据质量约束的前提下, 对原始数据的改动最小.

时空数据实时性: CurrentClean[77]是一个识别和清理陈旧数据的概率系统, 通过学习和预测更新模式来推断陈旧值. 它引入的时空概率模型考虑了数据更新的空间相关性(如相邻数据单元间的更新关系)和时间相关性(如数据单元在不同时间点的更新频率), 并定义了一组推理规则, 以模拟现实数据中常见的时空更新模式. 一旦识别出陈旧数据, CurrentClean 会通过学习历史更新值为当前状态提供建议值, 从而恢复数据的时效性.

4.2 基于STDM的时空模型训练

现有的时空数据处理框架尽管为时空数据的管理和查询提供了基础支持, 但其对 AI 模型训练的支持能力相对有限. 如图 17 所示, 复杂的深度时空模型往往参数量巨大, 导致训练过程极度消耗计算资源(GPU/内存)且周期漫长. 基于 STDM 的时空模型训练旨在从**模型效果**与**模型效率**两个维度优化 STAI 的训练过程: 针对静态数据, 由于数据规模和复杂度相对可控, 核心目标在于提升模型效果; 针对动态数据, 受限于时空非均匀性与实时性要求, 核心挑战在于突破效率瓶颈. 且下文针对动态关联数据的研究多数可以被有效复用来提升动态独立数据的模型训练效率. 此外, 静态关联数据在 STAI 中通常仅作为模型的上下文信息,目前学术界尚缺乏针对该类数据进行 STDM 驱动训练优化的专门研究, 故不作深入展开.

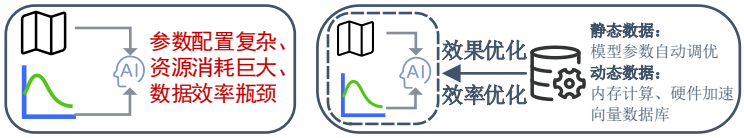


图 17 传统的时空模型训练与基于 STDM 时空模型训练对比

4.2.1 静态独立数据

STAI 模型效果优化: 在模型训练的初始阶段, "选择哪个模型、如何配置参数"是一个耗时且复杂的任务. PAIRS AutoGeo[78]框架通过以下方式解决了如何选择模型的问题: (1) 提供预定义模型集: PAIRS AutoGeo 内置了一个模型库, 其中包含了一系列针对地理空间数据优化的预定义模型. 这种预设的模型选择大大降低了用户从零开始寻找和构建合适模型的难度. (2) 支持用户选择或自动优选: 用户可以请求框架训练一组模型, 并让系统自动选择其中表现最佳的模型进行后续操作. (3) 自动化参数调优: 对于所选模型, PAIRS AutoGeo 会自动进行参数调优. 例如, 在随机森林分类器训练中, 框架默认会根据模型在验证集上的性能, 通过网格搜索来优化 n_estimators 和 max_depth 等参数.

4.2.2 动态关联数据

STAI 模型效率优化: 对于大规模且对实时性要求较高的动态关联数据, STAI 的训练速度往往受限于数据加载和复杂空间操作的效率. STDM 通过内存计算和硬件加速等技术, 显著提升了这些基础操作的速度. 例如, SparkGIS[79]利用 Apache Spark 的内存处理能力, 显著提高了空间查询的效率、可扩展性和性能. Catfish[80]利用远程直接内存访问机制优化了分布式系统中的 R 树处理, 通过低延迟查询和工作负载分流来提升吞吐量.

在硬件加速技术上, DGC[81]系统提出针对大规模动态图神经网络的高效分布式训练方案, 旨在解决动态图的时空非均匀性. 其核心在于一种新的图划分方法和运行时优化技术. 在图划分阶段, DGC 将动态图划分为“图块”, 使用类似于成本估计的思想, 使用多层感知机准确预测每个块的结构编码器和时间编码器的执行时间, 并使用启发式算法将块分配给 GPU, 以平衡负载并最小化通信成本. PixelBox[82]算法和 RayJoin[83]分别通过 GPU 的并行计算能力和光线追踪硬件(RT Core), 加速了多边形相交和空间连接等复杂的空间操作. 上述技术的应用, 有效解决了 AI 模型训练中常见的性能瓶颈, 确保了数据能以更快的速度传输和处理.

特别地, 在轨迹数据存储与处理领域, 利用向量数据库加速相似性查询已成为研究趋势. DGGS[84]提供了一种多分辨率、层次化的地球表面划分方式, 通过将复杂的连续轨迹转化为离散的网格序列, 不仅简化了空间拓扑关系的表达, 更有效地将高维时空匹配问题转化为低维的序列查找或向量索引问题, 从而显著提升了在大规模数据集下的查询性能.

4.3 基于STDM的时空模型使用

模型训练完成后, 如图 18 所示, 传统模型使用流程中数据和模型在不同平台间切换带来的 I/O 开销巨大, 且模型的使用门槛较高, 现代时空数据管理正积极推动数据存储与分析功能的深度整合, 用户无需关注底层的分布式计算或复杂的模型推理流程, 即可通过熟悉的接口完成任务. 这主要体现在以下两种调用模式: (1) SQL 调用: 允许用户通过 SQL 或类 SQL 语言(如图查询语言 Cypher)直接使用 STAI 模型. 这种形式将复杂执行逻辑和模型调用过程交给 STDM 系统在底层自动完成. (2) API 调用. 对于更复杂或需要利用外部先进深度学习框架的场景, STDM 通过提供 API 接口, 将数据管理平台和数据分析整合统一平台.

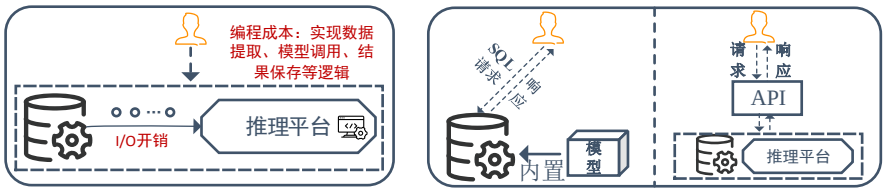


图 18 传统的时空模型使用与基于 STDM 的模型使用对比

4.3.1 静态独立数据

SQL 调用. JUST[89]通过扩展 SQL 接口, 允许用户直接在空间数据集上执行聚类操作. 例如, 用户可以通过以下 SQL 查询语句调用 DBSCAN 算法, 其中 minPts 和 radius 为算法参数:

SELECT *st_DBSCAN(geom, minPts, radius)*

FROM *<tableName|viewName>*

这种方式使得用户无需编写额外的程序代码, 即可在数据库环境中完成复杂的 STAI 聚类任务.

API 调用. 对于一般数据库难以完成的高维张量计算与复杂卷积操作, Ishneet 等人[90]利用 Amazon SageMaker 的地理空间功能及其预训练的土地覆盖分割模型, 通过简单的 API 调用, 使用户无需进行繁琐的数据标注、模型训练和部署就能高效地处理和分析卫星图像.

4.3.2 动态独立数据

SQL 调用. 针对具有流式特征的独立数据, Google BigQuery ML[91]等云数据仓库扩展了 SQL 语法, 支持直接对动态独立数据进行时序预测. 用户仅需使用 CREATE MODEL 和 ML.FORECAST 等标准 SQL 语句, 即可在数据驻留层直接训练并调用 ARIMA 或 RNN 模型预测未来趋势, 实现了动态数据的即存即用.

API 调用. 针对复杂的时空事件预测等任务, GeoTorchAI[92]提供了一套基于 Python 的 API 接口, 深度集成了 PyTorch 与 Apache Sedona. 通过调用 API 可以直接将存储在分布式系统中的数据转换为可训练的张量, 并调用内置的 ST-ResNet 或 ConvLSTM 模型进行推理.

4.3.3 动态关联数据

SQL 调用. 文献[93]综述了视频数据管理系统通过集成深度学习框架支持视觉任务, 用户可通过扩展

SQL 语法直接调用这些功能. 以 GNOSIS[94]为例, 它将视频事件检测视为图匹配问题, 并利用 RedisGraph 数据库执行查询. GNOSIS EPL(基于 openCypher)通过 CONTENT 子句隐式调用模型, 用户无需编写底层代码即可组合多模型逻辑.

API 调用. Google BigQuery[91]提供了基于 Python 的 BigQuery DataFrames 接口. 该接口支持将复杂的时空查询逻辑(如轨迹清洗、时空连接)自动下推至数据库内核执行, 并通过与 Vertex AI 的深度集成, 允许用户通过 API 调用预训练的深度学习模型对轨迹进行异常检测或目的地预测.

4.4 总结

本小节对 ST-DM4AI 进行了回顾, 具体总结如表 4 所示.

表 4 ST-DM4AI 研究总结

研究内容	数据类型	细分方向	名称
基于STDM的时空数据准备	静态独立数据	时空数据准确性	SPARCLE[72]
	动态独立数据	时空数据准确性	WEEVER[73]
		时空数据实时性	SPEAR[74]
	静态关联数据	时空数据准确性	QARTA[75]
	动态关联数据	时空数据准确性	Clean4TSDB[76]
		时空数据实时性	CurrentClean[77]
基于STDM的时空模型训练	静态独立数据	模型效果优化	PAIRS AutoGeo[78]
	动态关联数据	模型效率优化	SparkGIS[79]、Catfish[80]、DGC[81] PixelBox[82]、RayJoin[83]、DGGS[84]
基于STDM的时空模型使用	静态独立数据	SQL调用	JUST[89]
		API调用	Amazon SageMaker[90]
	动态独立数据	SQL调用	BigQuery ML[91]
		API调用	GeoTorchAI[92]
	动态关联数据	SQL调用	GNOSIS[94]
		API调用	BigQuery ML[91]

5 数据集

为了促进时空数据管理与时空 AI 协同应用领域的快速发展, 比为相关研究人员提供参考, 本节总结了常用的公开数据集, 如表 5 所示.

ST-AI4DM 的核心目的在于利用 AI 技术提升数据库自身的管理效率, 例如优化数据存储结构、提升时空索引性能或改进查询响应时间. 针对此类研究, 数据集通常作为 AI 模型学习优化任务的训练数据. 理想的数据集需要能够涵盖不同类型的地理实体, 并反映多样化的空间分布特征. 数据集主要分为: (1) 真实世界数据: 以 OSM[95]为例, 它是一个全球性、可编辑的地理数据库, 数据来源多样(包括用户手动调查、GPS 记录、航空摄影等), 具有丰富的空间语义和复杂的拓扑结构, 适合验证索引和存储系统的普适性. (2) 合成数据: 这类数据集通常由 Spider[96]等数据生成器创建, 旨在模拟具有特定且多样化空间分布的数据对象, 包括均匀分布、线性分布、对角线分布等以及它们的组合. 合成数据有助于研究者训练能够适应多种数据形态与分布的 AI 模型, 从而系统评估不同管理策略的性能. 而对于基于学习的时空查询优化研究, 历史查询负载同样是至关重要的训练数据. 由于公开的、同时包含大型时空库查询日志的数据集极为稀少, 目前仅有少数研究能够从实际日志中提取相关负载. 现有研究多采用模拟生成方式构建历史查询负载, 以确保查询类型与覆盖区域同底层数据分布相匹配. 如 LATEST 研究[63]便通过结合真实空间范围与从数据集中抽取的关键词, 模拟了多种空间/关键词/混合查询比例的工作负载, 以验证其自适应框架的有效性.

ST-DM4AI 的核心目标是通过改进时空数据管理, 为 AI 模型提供更高效、更高质量的数据服务. 因此, 该类研究所用数据集需能够体现 AI 建模全流程中的关键难点, 以验证数据管理策略的有效性. 基于 STDM 的时空数据准备工作旨在为 AI 模型提供高质量的数据集, 以 AustinCode[97]数据集为例, 其 93,414 条记录里包含了 13,968 条错误记录, 这能有效验证时空数据清洗系统的错误检测与修复能力. 基于 STDM 的时空模型训练旨在提升 AI 模型的训练效率, 其核心挑战在于处理海量数据时避免存储与计算间的低效数据迁移. 为此,

NAIP[98]等 TB/PB 级的卫星影像数据集成为关键的评测基准. 基于 STDM 的模型使用侧重于数据管理层如何支持 AI 模型被用户轻松、高效地调用和集成, 因此表 5 未对这类数据集进行统计.

表 5 数据集表

研究分类	研究类型	数据类型	数据集
ST-AI4DM	基于学习的时空数据存储	静态独立数据	OSM[95]、imis-3months[99]、Buildings、Lakes、Roads[100]
		静态关联数据	UCR-Star[101]
		动态关联数据	Geolife[102]、T-Drive[103]、Chengdu[104]、OSM[95]
	基于学习的时空数据索引	静态独立数据	POST[105]、Tiger[106]、HKI[107]、imis-3months[99]、Foursquare[108]、OSM[95]、BPD[109]、UCR-Star[101]
		动态关联数据	Geolife[102]、T-Drive[103] 、Chengdu[104]、Porto[110]、ATC[111]
	基于学习的时空查询优化	静态独立数据	Crime[112]
		动态独立数据	UCR-Star[101]、GoT[113]、NYC Taxi[114]
		静态关联数据	UCR-Star[101]
		动态关联数据	Foursquare[108]
ST-DM4AI	基于 STDM 的时空数据准备	静态独立数据	AustinCode[97]、Boston-311[115]、Chicago-Building[116]、NYC-Crash[117]
		动态独立数据	Sensor[118]
		动态关联数据	SWaT[119]、Brest-AIS[120]、MarineTraffic[121]、Porto[110]、Air[122]
	基于 STDM 的时空模型训练	静态独立数据	NAIP[98]
		动态关联数据	Brest-AIS[120]、MarineTraffic[121]

6 典型应用案例

6.1 ST-AI4DM应用案例

通用数据库领域已出现 openGauss[123]等将 AI 技术深度集成到数据库内核的成熟系统, 但在时空数据管理领域, 此类全栈式智能化系统仍处于探索阶段. 鉴于此, 本文选取 SJML 框架[124]作为 ST-AI4DM 的典型案例. 如图 19 所示, 该框架是首个针对分布式空间连接操作提出的全流程机器学习优化框架.

在数据存储环节, SJML 摒弃了传统静态分区策略, 引入了分区策略选择模型(M3)和分区数量调优模型(M4). 该机制能够根据输入数据的空间分布特征, 智能决策重分区策略, 并自动预测最佳的分区粒度, 从而在数据组织和存储阶段实现数据负载均衡.

在查询优化环节, SJML 构建了基数估计模型(M1)和计算代价预测模型(M2), 通过提取卷积直方图等深层特征, 该框架能够以较高精度预估查询结果集大小及几何计算量, 为优化器提供了精准的决策依据. 此外, SJML 实现了执行算子的自适应选择, 能通过分类模型智能选择在当前硬件与数据下最优的连接算法, 还通过平面扫描排序模型(M5)动态调整局部连接时的扫描轴方向(即沿 X 轴或 Y 轴).

实验结果表明, 传统基于理论公式的方法仅适用于理想的均匀分布数据, 在处理真实世界的倾斜数据时, 其基数与代价估计误差高达 35%, 算法选择准确率仅为 32.2%; 相比之下, SJML 框架的估算误差为 4.49%, 最优算法预测准确率为 82%. 这证明了在多源异构与数据高度倾斜的复杂时空场景下, 基于学习的优化框架具有显著优于传统规则的实践价值.

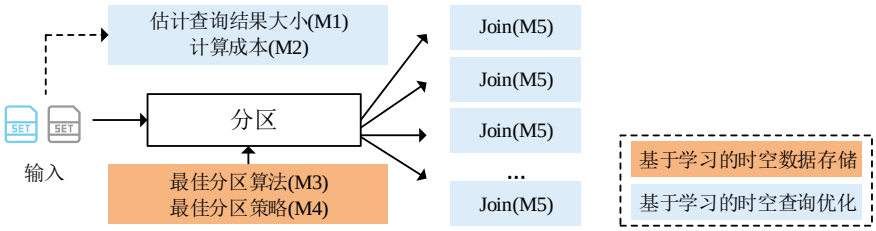


图 19 SJML 框架图[124]

6.2 ST-DM4AI应用案例

如图 20 所示, ST4ML[125]是基于 Spark 构建的分布式系统, 通过“选择-转换-提取”的三阶段流水线, 实现

前文 ST-DM4AI 框架中的数据准备、模型训练和模型使用三个环节.

在时空数据准备环节, 在选择阶段, ST4ML 利用磁盘元数据索引和内存 R 树索引, 从海量原始数据中精确过滤出与任务相关的时空子集, 避免全量加载导致的资源浪费. 在“转换”和“提取”阶段, ST4ML 将非结构化的数据(如轨迹点坐标)高效转化为 AI 模型所需的结构化特征(如时空张量等), 直接提供 AI 就绪的数据集.

在时空模型训练环节, 由于 ST4ML 将能数据管理与计算逻辑统一在同一内存环境中, 消除了在时空数据库与时空计算引擎之间频繁迁移数据的开销, 构建了端到端的高效数据管道.

在时空模型使用环节, ST4ML 提供了用户友好的编程接口, 用户可以轻松调用内置的特征提取器, 无需深入理解底层分布式系统的复杂性.

实验表明, 在数据准备环节 ST4ML 能够过滤掉 42%至 98%的无关数据, 并节省约 60%的数据加载时间. 在特征提取任务中, ST4ML 的性能比传统扩展方案提升了 27 倍至 39 倍. 同时, ST4ML 极大简化了开发流程, 将实现相同业务逻辑的代码量减少了约 50%以上. 目前该系统已成功部署于阿里巴巴城市大脑实验室, 在实际的交通监控与流量预测业务中展现了极高的可用性, 并能够通过标准数据格式对接 TensorFlow 和 PyTorch 等主流 AI 框架.

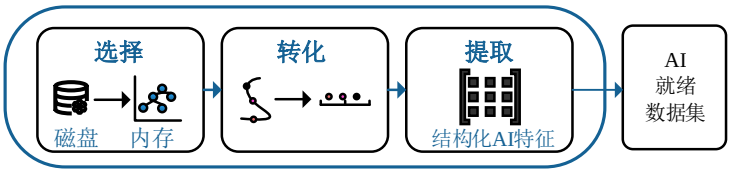


图 20 ST4ML 三阶段示意图

7 挑战与未来展望

7.1 挑战

在推动时空数据管理与人工智能协同发展的研究中, 尽管已有诸多进展, 但 ST-AI4DM 和 ST-DM4AI 都面临着一系列深层次且亟待突破的挑战. 对于 ST-AI4DM 而言, 其核心挑战如下:

(1) **跨环境性能优化的挑战:** 时空数据库因其海量、高动态的特性, 普遍采用 NoSQL 数据库与分布式环境进行管理. 这一现状带来了如何在不同硬件与系统架构下, 确保 AI 驱动的时空数据分区、索引及查询优化均能达到最佳性能的挑战. AI 模型需具备学习不同计算资源和网络拓扑特征的能力, 以实现跨环境的性能优化. 尽管面向通用数据库的“旋钮调优”(即通过调整配置参数以优化性能和资源利用率)已是相对成熟的研究领域[126], 但由于时空数据管理系统环境的异构性, 跨环境的“旋钮调优”研究仍相对稀缺.

(2) **动态时空环境下的模型维护成本高昂:** AI4DM 组件通常是采用针对特定数据快照训练的静态模型. 随着新数据的不断写入, 模型对数据分布的拟合度可能会迅速衰减, 进而导致查询性能退化. 然而, 模型重训练在计算资源和时间成本上往往难以承受. 特别是在高吞吐量的时空场景下, 频繁的重训练会抢占数据库的计算资源. 因此, 如何在保证模型准确性的前提下, 将模型维护的开销控制在系统可接受的范围内, 成为当前亟待突破的关键技术挑战.

对于 ST-DM4AI 而言, 其核心挑战则体现在:

(1) **面向表示学习的高效数据准备:** 随着深度学习模型在时空 AI 领域的广泛应用, 对向量化和矩阵化数据特征的需求日益增长. 尽管已有如 ST4ML[125]这样专用的系统能够从大规模时空数据中高效提取特征, 但对于大多数现有的通用时空数据库系统而言, 其在设计时主要面向时空查询任务而非 AI 训练. 因此, 导致用户在使用通用数据库时, 仍需经历繁琐的数据导出、预处理等流程. 如何在更广泛的时空数据管理系统中原生集成这种复杂的转换能力, 以提升整体效率, 仍是当前面临的关键挑战.

(2) **时空模型管理挑战:** 随着 STAI 应用的深入, AI 模型的复杂性, 尤其是其庞大的参数量, 给时空模型

管理带来了显著挑战. 具体而言, 如何高效管理模型的空间适用范围与时间有效性, 使得系统能根据查询的时空条件检索适用的时空模型. 其核心在于构建高效的模型管理系统, 以支持时空 AI 模型的存储、版本控制、检索、分析与共享. 面对数量众多、参数各异的时空 AI 模型, 传统的数据管理范式难以直接适用.

7.2 未来展望

未来的研究与应用将致力于突破现有瓶颈, 在 ST-AI4DM 与 ST-DM4AI 两个层面均展现出令人期待的发展前景. 在深化 ST-AI4DM 方面, 未来的发展将聚焦于:

(1) **跨环境的自适应调优框架:** 针对时空数据库在多源异构硬件环境下的性能适配难题, 未来的研究将致力于突破 ST-AI4DM 跨环境应用的局限. 迁移学习技术有望使 AI 模型能够从源环境(如通用云服务器)中提取共性特征, 并快速迁移至目标环境(如边缘计算节点或新型分布式架构), 从而实现低成本的跨硬件性能泛化. 为了克服针对时空管理系统在“旋钮调优”仅局限于局部参数微调的瓶颈, 将探索基于深度强化学习的端到端自治调优系统, 能够根据异构环境中的实时的时空负载波动, 动态调整缓存、分区及索引参数.

(2) **轻量化增量学习与在线演化机制:** 为了应对时空数据的高动态性, 未来的研究将从“周期性重训练”转向“轻量化增量学习”. 其核心目标在于赋予 ST-AI4DM 组件快速适应新数据分布的能力. 借鉴持续学习思想, 可研究仅需微调少量参数或更新局部模型结构即可适配新数据的算法, 将轻量级的训练任务下沉到数据库存储引擎中, 实时更新模型的统计特征, 从而实现模型的实时维护.

在构建 ST-DM4AI 方面, 未来的发展将致力于:

(1) **原生支持 AI 表示学习的流批一体化平台:** 为解决 STAI 应用对历史与实时数据融合分析的迫切需求, 但当前流批处理割裂带来的挑战, 未来的一个核心方向是构建能够统一管理和分析历史批数据与实时流数据的时空数据平台. 通过在系统内部原生支持向量化和张量化操作, 平台不仅能统一管理历史与实时数据, 还能直接输出 AI 模型所需的中间表示, 从而消除数据搬运与格式转换的性能瓶颈.

(2) **构建 STAI 模型管理系统:** 针对 STAI 模型在空间适用性与时间有效性界定方面的挑战, 未来的数据管理系统将构建 STAI 模型管理系统. 该系统能将模型的空间适用边界与时间有效窗口作为时空模型的核心属性进行注册. 在此基础上, 引入高效的多维时空索引结构对模型库进行组织, 从而支持系统能够像查询时空数据一样, 毫秒级地根据查询条件从海量模型中筛选出候选集合, 实现对时空 AI 模型的高效存储、检索与应用.

8 总结

本文旨在总结近年来时空管理系统智能化(ST-AI4DM)和面向 STAI 应用的高效时空数据管理技术(ST-DM4AI)的研究进展. 我们从数据的角度出发, 对相关研究内容进行分类, 并深入阐述各类研究的动机和所采用的方法. 对于 ST-DM4AI 而言, 研究的重点在于基于学习的时空数据存储、基于学习的时空数据索引和基于学习的时空数据查询优化; 对于 ST-DM4AI 而言, 研究的重点在基于 STDM 的时空数据准备、基于 STDM 的时空模型训练和基于 STDM 的时空模型使用. 最后, 本文汇总以上研究所涉及的数据集, 给出了典型应用案例, 并深入分析当前研究面临的挑战, 同时对未来的研究方向进行展望.

References:

[1] Wang S, Cao J, Philip S Y. Deep learning for spatio-temporal data mining: A survey. IEEE transactions on knowledge and data engineering, 2020, 34(8): 3681-3700.

[2] Eldawy A, Mokbel M F. The era of big spatial data. 2015 31st IEEE International Conference on Data Engineering Workshops. IEEE, 2015: 42-49.

[3] Li R, He H, Wang R, et al. Trajmesa: A distributed nosql-based trajectory data management system. IEEE

Transactions on Knowledge and Data Engineering, 2021, 35(1): 1013-1027.

[4] Ruan SJ, Xiong Z, Long C, Chen YH, Bao J, He TF, Li RY, Wu SN, Jiang ZY, Zheng Y. Doing in one go: delivery time inference based on couriers' trajectories. In: Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining, 2020, 2813-2821.

[5] Cao B.Y., Feng H.S., Liang J.H., et al. Spatio-temporal big data storage and indexing using Hilbert curves and Cassandra technology. *Journal of Wuhan University (Information Science Edition)*, 2021, 46(5): 620-629.

[6] Jasny M, Ziegler T, Kraska T, et al. DB4ML-an in-memory database kernel with machine learning support. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. 2020: 159-173.

[7] Chai M.K., Fan J., Du X.Y. Learning-based database systems: Challenges and opportunities. Journal of Software, 2020, 31(3): 806-830.

[8] Li G.L., Zhou X.H. Survey of data management techniques for supporting artificial intelligence. Journal of Software, 2021, 32(1): 21-40.

[9] Li G.L., Zhou X.H. XuanYuan: An AI-native database system. Journal of Software, 2020, 31(3): 831-844.

[10] Chai C, Wang J, Luo Y, et al. Data management for machine learning: A survey. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(5): 4646-4667.

[11] Zhou X, Chai C, Li G, et al. Database meets artificial intelligence: A survey. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(3): 1096-1116.

[12] Liang H, Zhang Z, Hu C, et al. A Survey on Spatio-temporal Big Data Analytics Ecosystem: Resource Management, Processing Platform, and Applications. IEEE Transactions on Big Data, 2023.

[13] Zeng M.X., Hua Y.X., Zhang J.S., et al. Research on dynamic behavior expression models and methods for multi-granularity spatio-temporal objects. *Journal of Geo-information Science*, 2021, 23(1): 104-112.

[14] Zheng, Y. Urban Computing. Cambridge, MA: MIT Press, 2019.

[15] Wang Z, Long C, Cong G, et al. Collectively simplifying trajectories in a database: A query accuracy driven approach. 2024 IEEE 40th International Conference on Data Engineering (ICDE). IEEE, 2024: 4383-4395.

[16] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The Case for Learned Index Structures. In Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18). Association for Computing Machinery, New York, NY, USA, 489–504.

[17] Yu S, Liu J, Gao C, et al. LLMTrajQuery: an LLM-based generative approach to semantic trajectory queries. International Journal of Geographical Information Science, 2025: 1-29.

[18] Liu MY, Xu JQ, Tong YX. Natural Language Query Transformation Method for Spatial Databases Based on Large Language Model. Ruan Jian Xue Bao/Journal of Software, 2026, 37(3): 1121–1142.

[19] Eldawy A, Alarabi L, Mokbel M F. Spatial partitioning techniques in SpatialHadoop. Proceedings of the VLDB Endowment, 2015, 8(12): 1602-1605.

[20] Lu P, Chen G, Ooi B C, et al. Scalagist: Scalable generalized search trees for mapreduce systems [innovative systems paper]. Proceedings of the VLDB Endowment, 2014, 7(14): 1797-1808.

[21] Vo H, Aji A, Wang F. SATO: a spatial data partitioning framework for scalable query processing. Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems. 2014: 545-548.

[22] Xie D, Li F, Yao B, et al. Simba: Efficient in-memory spatial analytics. Proceedings of the 2016 international conference on management of data. 2016: 1071-1085.

[23] Eldawy A, Li Y, Mokbel M F, et al. CG_Hadoop: computational geometry in MapReduce. Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2013:

294-303.

[24] Zein A A, Dowaji S, Al-Khayatt M I. Clustering-based method for big spatial data partitioning. Measurement: Sensors, 2023, 27: 100731.

[25] Hamerly G, Elkan C. Learning the k in k-means. Advances in neural information processing systems, 2003, 16.

[26] Belussi A, Migliorini S, Eldawy A. Skewness-based partitioning in SpatialHadoop. ISPRS International Journal of Geo-Information, 2020, 9(4): 201.

[27] Belussi, Alberto, Sara Migliorini, and Ahmed Eldawy. Detecting skewness of big spatial data in SpatialHadoop. Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2018.

[28] Hori K, Sasaki Y, Amagata D, et al. Learned spatial data partitioning. Proceedings of the sixth international workshop on exploiting artificial intelligence techniques for data management. 2023: 1-8.

[29] Li R, Li J, Zhou M, et al. Learning-Based Distributed Spatio-Temporal k Nearest Neighbors Join. IEEE Transactions on Big Data, 2024.

[30] Li, Jiajun, et al. DiST: Efficient Distributed Spatio-Temporal Clustering With Automatic Parameter Optimization. IEEE Transactions on Knowledge and Data Engineering (2025).

[31] Tin Vu, Ahmed Eldawy, Vagelis Hristidis, and Vassilis Tsotras. 2021. Incremental partitioning for efficient spatial data analytics. Proc. VLDB Endow. 15, 3 (November 2021), 713–726.

[32] Vu T, Belussi A, Migliorini S, et al. Using deep learning for big spatial data partitioning. ACM Transactions on Spatial Algorithms and Systems (TSAS), 2020, 7(1): 1-37.

[33] P. A. P. Moran. Notes on continuous stochastic phenomena. Biometrika, 1950, 37:17–23.

[34] Eldawy A, Hristidis V, Ghosh S, et al. Beast: Scalable exploratory analytics on spatio-temporal data. Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021: 3796-3807.

[35] Khan, Sajid Ali. "Clustering algorithm on spatiotemporal trajectories." 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, 2019.

[36] Steadman L, Griffiths N, Jarvis S, et al. KD-STR: A method for spatio-temporal data reduction and modelling. ACM/IMS Transactions on Data Science, 2021, 2(3): 1-31.

[37] Al-Mamun A, Wu H, He Q, et al. A survey of learned indexes for the multi-dimensional space. ACM Computing Surveys, 2025, 58(4): 1-37.

[38] Liu Q, Li M, Zeng Y, et al. How good are multi-dimensional learned indexes? An experimental survey. The VLDB Journal, 2025, 34(2): 17.

[39] Wang H, Fu X, Xu J, et al. Learned index for spatial queries. 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2019: 569-574.

[40] Qi J, Liu G, Jensen C S, et al. Effectively learning spatial indices. Proceedings of the VLDB Endowment, 2020, 13(12): 2341-2354.

[41] Qi J, Tao Y, Chang Y, et al. Theoretically optimal and empirically efficient r-trees with strong parallelizability. Proceedings of the VLDB Endowment, 2018, 11(5): 621-634.

[42] Davitkova A, Milchevski E, Michel S. The ML-Index: A Multidimensional, Learned Index for Point, Range, and Nearest-Neighbor Queries. EDBT. 2020: 407-410.

[43] Hu Y, Tang P, Meng Y, et al. TBSI: a Transformer-based spatial learned index for efficient construction and query. International Journal of Geographical Information Science, 2025: 1-29.

[44] Zhang S, Ray S, Lu R, et al. SPRIG: A learned spatial index for range and kNN queries. Proceedings of the

17th International Symposium on Spatial and Temporal Databases. 2021: 96-105.

[45] Zhao X, Lam K Y. Density based learned spatial index for clustered data. Information Systems, 2025: 102606.

[46] Li P, Lu H, Zheng Q, et al. LISA: A learned index structure for spatial data. Proceedings of the 2020 ACM SIGMOD international conference on management of data. 2020: 2119-2133.

[47] Zeng M, Zou B, Kui X, et al. PA‐LBF: Prefix‐Based and Adaptive Learned Bloom Filter for Spatial Data. International Journal of Intelligent Systems, 2023, 2023(1): 4970776.

[48] Sheng Y, Cao X, Fang Y, et al. WISK: a workload-aware learned index for spatial keyword queries. Proceedings of the ACM on Management of Data, 2023, 1(2): 1-27.

[49] Sheng Y. Towards Efficient Spatial and Textual Query Processing: Learned Indexing and Cardinality Estimation[D]. UNSW Sydney, 2025.

[50] Gu T, Feng K, Cong G, et al. The rlr-tree: A reinforcement learning based r-tree for spatial data. Proceedings of the ACM on Management of Data, 2023, 1(1): 1-26.

[51] Al-Mamun A, Haider C M R, Aref W G. Query Processing Tradeoffs over an ML-Enhanced R-tree. Proceedings of the 8th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery. 2025: 174-184.

[52] Choi D, Yoon H, Lee H, et al. Waffle: in-memory grid index for moving objects with reinforcement learning-based configuration tuning system. Proceedings of the VLDB Endowment, 2022, 15(11): 2375-2388.

[53] Guo N, Sun W, Cai F, et al. HELM: Hybrid Spatial Index of Moving Objects at Large Scales Tuned with Multi-Agent Reinforcement Learning. Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. Singapore: Springer Nature Singapore, 2025: 231-246.

[54] Gu T, Feng K, Yang J, et al. BT-Tree: A Reinforcement Learning Based Index for Big Trajectory Data. Proceedings of the ACM on Management of Data, 2024, 2(4): 1-27.

[55] Wu S, Pang Z, Chen G, et al. NEIST: A neural-enhanced index for spatio-temporal queries. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(4): 1659-1673.

[56] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez, “Indexing the positions of continuously moving objects,” in SIGMOD,2000.

[57] Ramadhan H, Kwon J. X-FIST: Extended flood index for efficient similarity search in massive trajectory dataset. Information Sciences, 2022, 606: 549-572.

[58] Zhang X, Eldawy A. Spatial Query Optimization With Learning. Proceedings of the VLDB Endowment, 2024, 17(12).

[59] Nathan V, Ding J, Alizadeh M, et al. Learning multi-dimensional indexes. Proceedings of the 2020 ACM SIGMOD international conference on management of data. 2020: 985-1000.

[60] Guo Y, Shao Z. Cymo: A Flexible Storage Model with Query-Aware Indexing for Spatio-Temporal Big Data. IEEE Transactions on Knowledge and Data Engineering, 2026.

[61] Amagata D, Arai Y, Fujita S, et al. Learned k-nn distance estimation. Proceedings of the 30th International Conference on Advances in Geographic Information Systems. 2022: 1-4.

[62] Belussi A, Migliorini S, Eldawy A. A generic machine learning model for spatial query optimization based on spatial embeddings. ACM Transactions on Spatial Algorithms and Systems, 2024, 10(4): 1-33.

[63] Patil M, Magdy A. LATEST: learning-assisted selectivity estimation over spatio-textual streams. 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021: 1607-1618.

[64] Domingos P, Hulten G. Mining high-speed data streams. Proceedings of the sixth ACM SIGKDD international

conference on Knowledge discovery and data mining. 2000: 71-80.

[65] Quoc H N M, Serrano M, Breslin J G, et al. A learning approach for query planning on spatio-temporal IoT data. Proceedings of the 8th International Conference on the Internet of Things. 2018: 1-8.

[66] Bai Q, Alsudais S, Li C, et al. Maliva: Using Machine Learning to Rewrite Visualization Queries Under Time Constraints. EDBT. 2023: 157-170.

[67] Lakshmi S, Zhou S. Selectivity estimation in extensible databases-a neural network approach. VLDB. 1998, 98: 24-27.

[68] Ji Y, Amagata D, Sasaki Y, et al. PolyCard: A learned cardinality estimator for intersection queries on spatial polygons. Journal of Intelligent Information Systems, 2025, 63(3): 873-891.

[69] Tian R, Zhang W, Wang F, et al. Cardinality estimation of activity trajectory similarity queries using deep learning. Information Sciences, 2023, 646: 119398.

[70] Anaconda. 2022. State of Data Science Report 2022. https:. www.anaconda. com/resources/whitepapers/state-of-data-science-report-2022. (Accessed on 3/27/2024).

[71] Basu S, Ganguly S, Mukhopadhyay S, et al. Deepsat: a learning framework for satellite imagery. Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. 2015: 1-10.

[72] Yuchuan Huang and Mohamed F. Mokbel. 2024. Sparcle: Boosting the Accuracy of Data Cleaning Systems through Spatial Awareness. Proc. VLDB Endow. 17, 9 (May 2024), 2349–2362.

[73] Kaminsky, Youri, Eduardo HM Pena, and Felix Naumann. Incremental Detection of Denial Constraint Violations. Proceedings of the VLDB Endowment 18.4 (2024): 1000-1012.

[74] Baig, Furqan, et al. SPEAR: Dynamic spatio-temporal query processing over high velocity data streams. 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021.

[75] Musleh M, Abbar S, Stanojevic R, et al. QARTA: an ML-based system for accurate map services. Proceedings of the VLDB Endowment, 2021, 14(11): 2273-2282.

[76] Ding X, Song Y, Wang H, et al. Clean4TSDB: A Data Cleaning Tool for Time Series Databases. Proceedings of the VLDB Endowment, 2024, 17(12): 4377-4380.

[77] M. Milani, Z. Zheng and F. Chiang, CurrentClean: Spatio-Temporal Cleaning of Stale Data. 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 2019, 172-183.

[78] Zhou W, Klein L J, Lu S. Pairs autogeo: an automated machine learning framework for massive geospatial data. 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020: 1755-1763.

[79] Furqan Baig, Hoang Vo, Tahsin Kurc, Joel Saltz, and Fusheng Wang. 2017.Sparkgis: Resource aware efficient in-memory spatial query processing. In Proceedings of the 25th ACM SIGSPATIAL international conference on advances in geographic information systems. 1–10.

[80] Mengbai Xiao, Hao Wang, Liang Geng, Rubao Lee, and Xiaodong Zhang. 2019.Catfish: Adaptive RDMA-enabled R-Tree for low latency and high throughput.In 2019 IEEE 39th International Conference on Distributed Computing Systems(ICDCS). IEEE, 164–175.

[81] Chen, Fahao, Peng Li, and Celimuge Wu. Dgc: Training dynamic graphs with spatio-temporal non-uniformity using graph partitioning by chunks. Proceedings of the ACM on Management of Data 1.4 (2023): 1-25.

[82] Furqan Baig, Chao Gao, Dejun Teng, Jun Kong, and Fusheng Wang. 2020. Accelerating spatial cross-matching on cpu-gpu hybrid platform with cuda and openacc. Frontiers in big Data 3 (2020), 14.

[83] Liang Geng, Rubao Lee, and Xiaodong Zhang. 2024. RayJoin: Fast and Precise Spatial Join. In Proceedings of the 38th ACM International Conference on Supercomputing (Kyoto, Japan) (ICS '24). Association for Computing Machinery, New York, NY, USA, 124–136.

[84] Mostafa A, Mokbel M F, Uribe A E. TrajSplit: Scalable and Accurate Trip Extraction from Raw GPS Trajectories. 2025 26th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2025: 156-167.

[85] Moyroud N, Portet F. Introduction to QGIS. QGIS and generic tools, 2018, 1: 1-17.

[86] Lemenkova P. Python libraries matplotlib, seaborn and pandas for visualization geospatial datasets generated by QGIS. Analele stiintifice ale Universitatii" Alexandru Ioan Cuza" din Iasi-seria Geografie, 2020, 1: 13-32.

[87] Hagedorn S, Gotze P, Sattler K U. The STARK framework for spatio-temporal data analytics on spark. 2017.

[88] Yu J, Wu J, Sarwat M. Geospark: A cluster computing framework for processing large-scale spatial data. Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. 2015: 1-4.

[89] Li R, He H, Wang R, et al. Just: Jd urban spatio-temporal data engine. 2020 IEEE 36th International Conference on Data Engineering (ICDE). IEEE, 2020: 1558-1569.

[90] Dua I K, Patel P. Monitoring flood using Amazon SageMaker geospatial capabilities. Authorea Preprints, 2024.

[91] Fernandes S, Bernardino J. What is bigquery?. Proceedings of the 19th International Database Engineering & Applications Symposium. 2015: 202-203.

[92] Chowdhury, Kanchan, and Mohamed Sarwat. Deep Learning with Spatiotemporal Data: A Deep Dive into GeotorchAI. 2024 IEEE 40th International Conference on Data Engineering (ICDE). IEEE, 2024.

[93] Ding Guangyao, Xu Chen, Qian Weining, Zhou Aoying. Advances in Visual Database Management Systems for Deep Learning. *Journal of Software*, 2024, 35(3): 1207-1230.

[94] Yadav P, Salwala D, Pontes F A, et al. Query-driven video event processing for the internet of multimedia things. Proceedings of the VLDB Endowment, 2021, 14(12): 2847-2850.

[95] OpenStreetMap. OSM data download service. Available online: http:. osm.db.in.tum.de/.

[96] Vu T, Migliorini S, Eldawy A, et al. Spatial data generators[M]. Spatial Gems, Volume 1. 2022: 13-24.

[97] AustinCode [n.d.]. Austin Open Data. Austin Code Complaint Cases. https://data.austintexas.gov/Public-Safety/Austin-Code-Complaint-Cases/6wtj-zbtb.

[98] “National agriculture line]. Available: imagery program (naip).” [On https://www.fsa.usda.gov/programs-and-services/ aerial-photography/imagery-programs/naip-imagery/.

[99] Chorochronos Project. IMIS 3-months Dataset. Available at: http://chorochronos.datastories.org/?q=content/imis-3months/.

[100] Eldawy A, Mokbel M F. Spatialhadoop: A mapreduce framework for spatial data. 2015 IEEE 31st international conference on Data Engineering. IEEE, 2015: 1352-1363.

[101] Ghosh S, Vu T, Eskandari M A, et al. UCR-STAR: The UCR spatio-temporal active repository. SIGSPATIAL Special, 2019, 11(2): 34-40.

[102] Zheng Y, Li Q, Chen Y, et al. Understanding mobility based on GPS data. Proceedings of the 10th international conference on Ubiquitous computing. 2008: 312-321.

[103] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, “T-drive trajectory data sample,” Microsoft Research, [Online]. Available: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

[104] DiDi Chuxing, “Chengdu taxi trajectory dataset,” [Online]. Available: https://drive.google.com/file/d/1onzDFpbD9OOfvOK7jHJ6Tpi2V4oKfxXR/view?usp=sharing.

[105] Xu J, Zheng B, Lee W C, et al. The D-tree: an index structure for planar point queries in location-based

wireless services. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1526-1542.
[106]TIGER/Line Shapefiles. https://www.census.gov/geo/maps-data/data/tiger-line.html, 2006. Accessed: 2020-06-10.
[107]Hong Kong 40 Index 2018. https://www.dukascopy.com/swiss/english/ marketwatch/historical/. [Online; accessed 20-Dec-2019].
[108]Foursquare check-in dataset. Available: https://foursquare.com/.
[109]SLIPO project. Available: http://slipo.eu/.
[110]Moreira-Matias L, Gama J, Ferreira M, et al. Predicting taxi–passenger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(3): 1393-1402.
[111]Brščić D, Kanda T, Ikeda T, et al. Person tracking in large public spaces using 3-D range sensors. I EEE Transactions on Human-Machine Systems, 2013, 43(6): 522-534.
[112]Chan T N, Cheng R, Yiu M L. QUAD: Quadratic-bound-based kernel density visualization. Proceedin gs of the 2020 ACM SIGMOD International Conference on Management of Data. 2020: 35-50.
[113]Le-Phuoc D, Quoc H N M, Quoc H N, et al. The graph of things: A step towards the live knowledg e graph of connected things. Journal of Web Semantics, 2016, 37: 25-35.
[114]NYC Taxi Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data. page.
[115]Boston311 [n.d.]. Analyze Boston. Boston 311 Service Requests. https://data.boston.gov/dataset/311-servi ce-requests.
[116]ChicagoBuilding [n.d.]. Chicago Data Portal. Building Permits. https://data. cityofchicago.org/Buildings/ Building-Permits/ydr8-5enu.
[117]NYCOpenData [n.d.]. NYC Open Data. Motor Vehicle Collisions - Crashes. https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95.
[118]Cinnos Mission Critical, “Data center readings,” https://www.cas. mcmaster.ca/ ~ zhengz13/Dataset/Sen sor.rar, 2018.
[119]iTrust Labs [n.d.]. iTrust Labs Dataset Info. Secure Water Treatment (SWaT). https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#swat.
[120]Zenodo [2017]. Brest AIS Dataset – Maritime Traffic Data from Brest Region.https://zenodo.org/record/ 2563256.
[121]MarineTraffic [n.d.]. MarineTraffic Historical AIS Dataset – Greece Region.https://www.marinetraffic.co m.
[122]Zheng Y, Yi X, Li M, et al. Forecasting fine-grained air quality based on big data. Proceedings of th e 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015: 2267-2276.
[123]Guoliang Li, Xuanhe Zhou, Ji Sun, Xiang Yu, Yue Han, Lianyuan Jin, Wenbo Li, Tianqing Wang, an d Shifu Li. 2021. OpenGauss: an autonomous database system. Proc. VLDB Endow. 14, 12 (July 202 1), 3028–3042.
[124]Vu T, Belussi A, Migliorini S, et al. A learning-based framework for spatial join processing: estimatio n, optimization and tuning. The VLDB Journal, 2024: 1-23.
[125]Liu K. Towards advanced distributed data processing: framework, optimization, and application. 2024.
[126]Zhao X, Zhou X, Li G. Automatic database knob tuning: A survey. IEEE Transactions on Knowledg and Data Engineering, 2023, 35(12): 12470-12490.

附中文参考文献:

[5] 曹布阳,冯华森,梁峻浩,等.利用 Hilbert 曲线与 Cassandra 技术实现时空大数据存储与索引.武汉大学学报(信息科学版),2021,46(05):620-629.

[7] 柴茗珂, 范举, 杜小勇. 学习式数据库系统: 挑战与机遇. 软件学报, 2020, 31(3): 806-830.

[8] 李国良, 周煊赫. 面向 AI 的数据管理技术综述. 软件学报, 2020, 32(1): 21-40.

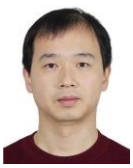
[9] 李国良, 周煊赫. 轩辕: AI 原生数据库系统. 软件学报, 2020, 31(3): 831-844.

[13] 曾梦熊, 华一新, 张江水, 等. 多粒度时空对象动态行为表达模型与方法研究. 地球信息科学学报, 2021, 23(1): 104-112.

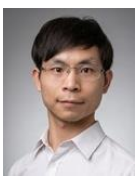
[18] 刘孟怡, 许建秋, 童咏昕. 基于大语言模型的空间数据库自然语言查询转换方法. 软件学报, 2026, 37(3): 1121－1142.

[93] 丁光耀,徐辰,钱卫宁,周傲英.支持深度学习的视觉数据库管理系统研究进展.软件学报,2024,35(3):1207-1230.

苏赛男(2001－),女,硕士生,主要研究领域为 AI4DB,时空数据管理与挖掘.



杨广超(1979－),男,博士,副教授,主要研究领域为数据库技术,模式识别,图像处理.

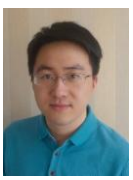


龙程(1988－),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为向量数据库,时空数据的管理和挖掘.

郑宇(1979－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为时空数据挖掘,城市计算.

李瑞远(1990－),男,博士,副教授,CCF 高级会员,主要研究领域为时空数据管理与挖掘,城市计算.

但静培(1978－),女,博士,副教授,硕士生导师,主要研究领域为数据挖掘,计算智能,大数据处理与分析.



张钧波(1986－),男,博士,CCF 杰出会员,主要研究领域为时空数据管理与挖掘,城市计算.