# Spatial Query Optimization With Learning

Xin Zhang
University of California, Riverside
Riverside, USA
xzhan261@ucr.edu

Ahmed Eldawy
University of California, Riverside
Riverside, USA
eldawy@ucr.edu

## ABSTRACT

Query optimization is a key component in database management systems (DBMS) and distributed data processing platforms. Recent research in the database community incorporated techniques from artificial intelligence to enhance query optimization. Various learning models have been extended and applied to the query optimization tasks, including query execution plan, query rewriting, and cost estimation. The tasks involved in query optimization differ based on the type of data being processed, such as relational data or spatial geometries. This tutorial reviews recent learning-based approaches for spatial query optimization tasks. We go over methods designed specifically for spatial data, as well as solutions proposed for high-dimensional data. Additionally, we present learning-based spatial indexing and spatial partitioning methods, which are also vital components in spatial data processing. We also identify several open research problems in these fields.

**PVLDB Reference Format:**
Xin Zhang and Ahmed Eldawy. Spatial Query Optimization With Learning. PVLDB, 17(12): 4245 - 4248, 2024.
doi:10.14778/3685800.3685846

## 1 INTRODUCTION

In database management systems (DBMSs) and data processing platforms, query optimization involves two main steps: logical transformation and cost estimation. Modern DBMSs and data processing platforms can process heterogeneous data. Each type of data requires specific considerations in its query optimization process. However, the main steps of query optimizations are always the same, regardless of the type of data being processed. The query optimizer returns a query plan with the lowest cost. A query plan is represented by a set of operators that form a query tree. The query transformer maps a query tree into equivalent query trees by following a set of rules inherited from relational algebra. The query optimizer evaluates the overall cost of each candidate plan to determine the optimal plan. The cost function considers access cost, storage cost, computation cost, and communication cost [31]. To estimate the computation cost, the query estimators compute the selectivity and cardinality through data statistics. The data statistics are extracted and compacted into data synopses, such as histograms, samples, sketches, and wavelets [9]. The I/O costs of the query execution are affected by data storage, such as partitioning and indexing. Over the past decades, extensive efforts have been made to incorporate different query optimization tasks and various query operation requirements.

AI4DB [18] becomes a very hot direction in the database community. Researchers spend efforts to improve query optimization by Artificial Intelligence (AI) techniques. Learned query optimizers beat the traditional heuristic solutions on generating more efficient query plans among the large search space [8, 21, 23, 40, 44, 49, 50]. Traditional synopses-based estimators rely on the Attribute Value Independence (AVI) assumption and cannot capture the attribute correlations [11, 32]. To overcome this limitation, researchers propose learned-based estimators including data-driven and query-driven models. The data-driven models [2, 17, 26, 30, 34, 36, 41, 42, 46, 52] learn the joint data distribution over all attributes. The query-driven models [16, 20, 25, 28, 38, 41, 48] learn a mapping function from queries to the corresponding cardinalities.

Existing learned-based query optimization techniques for high-dimensional data cannot be directly applied to spatial data because the spatial query operators are more complicated than high-dimensional vector data. Spatial data includes raster and vector formats. Raster data is represented by a grid of regularly sized pixels, while vector data uses geometry, such as points, linestrings, and polygons, defined by a set of numerical values. Due to the complexity of spatial attributes, expressing a spatial query in SQL often requires multiple inequality predicates in the WHERE clause. The classification of spatial queries is not unified. One group categorizes spatial queries into five types: basic, join, computational geometry, data mining, and raster operations [13]. The other group classifies spatial queries on vector data into five types: topology-based, metric-based, and direction-based [7, 24]. Techniques for spatial query optimization differ significantly from those used for relational data query optimization. For query rewriting, due to the complex data types and user-defined functions, several heuristic rules by relational query rewriter are no longer unconditional for spatial query rewriter [31]. For cost estimators, spatial query estimations are also more complex than relational query estimations. Spatial queries involve both spatial operators and nonspatial operators. Estimating spatial operators is often associated with inequality relations, such as OVERLAP, DISTANCE, etc.

Previous tutorials present comprehensive studies about learned-based DBMS [18], query optimizer [15], and learned-based query optimizers [33, 43]; however, they missed the spatial query operators and requirements. Several tutorials discuss spatial data management [12] and spatial data applications [6, 29], but none of them link to spatial query optimization in DBMS and data processing platforms. In this tutorial, we aim to review the existing methods in learning-based query plan generators and cost estimators and discuss the open problems in spatial query optimization. Additionally,


Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685846

- **Part 1: Spatial Data Management and Spatial Data Processing (25 minutes)**
  - Brief overview of the query optimization in DBMS and distributed data processing framework (Spark)
  - Highlight the specific features related to manage the spatial data and spatial query processing
- **Part 2: Learned Solutions for Query Optimizer (15 minutes)**
  - Query Rewriter
  - Query Optimizer
- **Part 3: Learned Solutions Cost Estimator (25 minutes)**
  - Brief overview of spatial query cost estimation tasks
  - Learned cardinality estimation
  - Learned join selectivity estimation
  - Other learned spatial queries
- **Part 4: Learned Solutions for Spatial Data Management (20 minutes)**
  - Brief overview of spatial indexes and spatial partitioning strategies
  - Learned Indexes
  - Learned Partitioning solutions
- **Part 5: Open Problems for Future Research (5 minutes)**

**Figure 1: Tutorial Outline (90 minutes)**

we will also cover learned solutions for spatial data management, including learning spatial index and spatial partitioning, to highlight the I/O cost of spatial query execution. A tutorial [1] reviews learned multi-dimensional indexes. As a supplement to [1], we discuss newly learned spatial indexes from recent years and learned spatial partitioning techniques in this tutorial.

## 2 TUTORIAL OUTLINE

Figure 1 shows the outline of this tutorial. We plan to spend 90 minutes discussing techniques for spatial query optimization with learning. This tutorial targets students, researchers, and practitioners who are interested in exploring problems in spatial query optimization. In the first part, we will introduce the background of query optimization components and spatial data processing. No prior knowledge of spatial data is required for the audience. We aim to inspire the audience with the following three parts: (1) Key features of spatial data management and spatial query optimizations; (2) Existing works on spatial query optimization; (3) Why techniques for relational query optimization cannot be directly applied to spatial query optimization; (4) Gaps for future research in spatial query optimization with learning. Figure 2 summarizes the works that will be discussed during the presentation and categorizes them based on query optimization goals.

### 2.1 Learned Solutions for Query Optimizer

We cover two topics about learned solutions for query optimizers: query plan optimizer [5, 8, 23, 34, 40, 44, 49, 50] and query rewriter [4, 24, 39, 51].

SJML [34] designs a spatial join framework based on several learning models. The proposed models predict the best spatial join algorithm and features, such as the plane-sweep direction (along the x- or y-axis). SpatialEmbedding [5] proposes a framework based on three learning models, which include an unsupervised model to capture the features of spatial datasets and two supervised models for the cost estimation of spatial operations. Optimizing the query execution plan is an important step in the DBMS and distributed data processing platforms. There are also several works [8, 23, 40, 44, 49, 50] that use learned models to evaluate the query plan and select the optimal query execution plan. These works are marked with grey color in Figure 1 because they are proposed for relational data and queries. Existing spatial optimizers focus on query execution and lack of supporting in considering query plans.

Maliva [4] applies the Markov Decision Process model to rewrite the queries. The proposed model can be applied to spatial aggregation queries. SemanticQueryOpt [24] proposes a strategy for the semantic query optimization of spatial join queries. This technique aims to eliminate unnecessary spatial joins or replace expensive spatial joins with cheaper thematic joins. Since SemanticQueryOpt is not a learning solution, it is marked with a dashed border in Figure 1. We mark LearnedRewriter [51] and WeTune [39] with grey color because they are rule-based learned query rewriters for relational data. Although these works in grey are not learning-based spatial query optimization techniques, we highlight them to guide future research on rule-based spatial query rewriters.

### 2.2 Learned Solutions for Cost Estimator

Extensive efforts have been dedicated to learned-based cost estimation. In this subsection, we cover works related to computation cost estimation. The computation cost of the query plan relates to selectivity and cardinality. Selectivity refers to the percentage of tuples among the whole dataset that satisfies the query predicates [3]. Cardinality refers to the number of results returned by each operation [3].

Researchers usually summarize computation cost estimation learning models into two categories: query-driven and data-driven. Query-driven models treat cardinality estimation as a regression problem and learn a mapping function between the query and its cardinality on a database [16, 20, 25, 28, 38, 41, 48]. They use query workload as labeled training data to learn supervised query models. Data-driven models learn the join data distribution of attributes directly from the dataset [2, 17, 26, 30, 34, 36, 42, 46, 52]. Some models take data as unsupervised information to learn unsupervised data models [2, 17, 26, 30, 41, 42, 46, 52] Some models are unsupervised data models that are directly learned from the data. Some models are supervised data models. LearningToSample [36] learns a probabilistic classifier by queries. SJML [34] learns data statistics similar to data synopses.

In spatial data processing, the statistics used to complete cardinality estimation tasks vary for different types of data. For example, estimating the cardinality of a spatial overlap-join for two sets of polygon datasets requires knowledge of the average volume and other features of the polygons. On the other hand, the cardinality of a distance-join for two sets of point datasets can be estimated using data distribution. In this part, we will highlight the types of cardinality estimation tasks each work addresses and the types of data each model can handle. SJML [34] is designed for polygon datasets and spatial join selectivity estimation. Some works are designed for high-dimensional datasets and can be applied to range query cardinality estimation for spatial point data [2, 16, 17, 20, 25, 26, 28, 32, 36, 38, 41, 42, 46, 48, 52]. LearningToSample [36] leans a classifier based on the sample of the data and can be applied to spatial point datasets. TurboReg [30] proposes a regression model for predicting the presence or absence of spatial

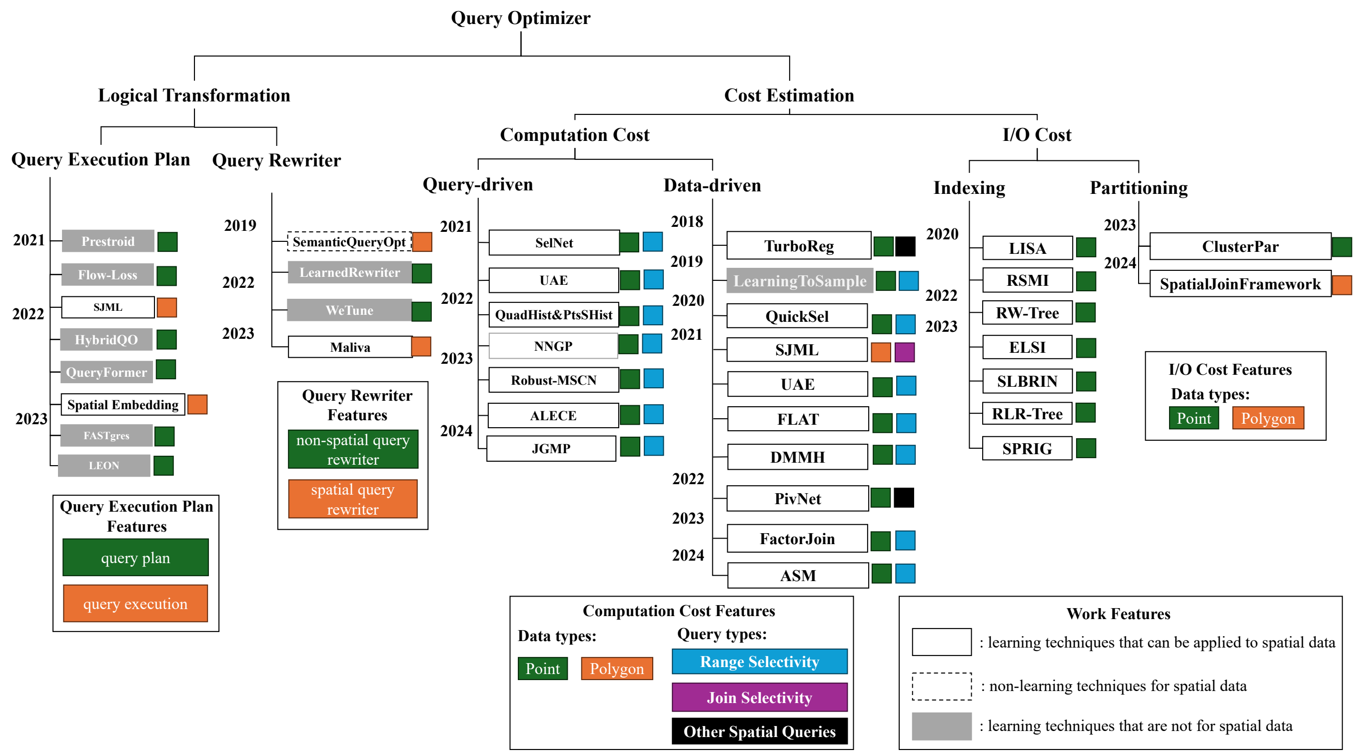


**Figure 2: Taxonomy of Query Optimization Techniques.**

phenomena. PivNet [2] uses a regression-based solution to estimate the k-NN queries.

## 2.3 Learned Solutions for Spatial Management

Data indexing is the most important factor that affects the I/O cost of spatial data processing in DBMSs. In distributed DBMSs and platforms, the I/O cost of spatial data processing is also related to data partitioning. Most recently proposed learned spatial indexes are designed for spatial point data [10, 14, 19, 22, 27, 37, 47]. RW-Tree [10] and RLR-Tree [14] leverage learning-based methods from query workload to help with R-tree construction. LISA [19], SLBRIN [37], and SPRIG [47] learn spatial index based on grid cells partitioning. RSMI [27] and ELSI [22] learn spatial index based on the z-order model. There are two recent works on learned spatial partitioning. ClusterPar [45] uses K-Means clustering to partition the spatial point datasets. SpatialJoinFramework [35] uses regression models to learn the features of spatial join. The models can decide the number of partitions and choose the partitioning strategy for spatial polygon datasets.

## 2.4 Open Problems for Future Research

After reviewing recently learned solutions for query optimization, we want to highlight several gaps in spatial query optimization: (1) Query rewriting rules for optimizing the spatial operations; (2) Models and frameworks that also consider the spatial query execution; (3) More focus on types of spatial data beyond point datasets, such as linestrings, polygons, and spatial raster data, along with their cost estimation tasks; (4) How to apply current learned partitioning models to existing distributed platforms and integrate these learned solutions into the query optimization components of distributed platforms.

# 3 PRESENTERS

Xin Zhang is a Ph.D. candidate in the Department of Computer Science and Engineering at the University of California, Riverside. She received her B.S. in Software Engineering from Shandong University in 2016 and her M.S. in Computer Science from Washington State University in 2018. She has interned at IBM Research, Almaden, and at Amazon AWS Redshift. Her research interests include big data management and approximate query processing.

Ahmed Eldawy is an Associate Professor in Computer Science at UC Riverside. His research focuses big data management and spatial data processing. Ahmed led the research and development in many open-source projects for big spatial data exploration and visualization including UCR-Star, an interactive repository with nearly four terabytes of publicly available geospatial data. He delivered a series of tutorials on big spatial data in VLDB, ICDE, IEEE BigData, and MDM. He is a recipient of the NSF CAREER award as well as the best demo award in SIGSPATIAL 2020.

# ACKNOWLEDGMENTS


This work is supported in part by the National Science Foundation (NSF) under grants IIS-1838222, CNS-1924694, IIS-2046236, and IIS-1954644.

## REFERENCES


[1] Abdullah Al-Mamun, Hao Wu, and Walid G Aref. 2020. A tutorial on learned multi-dimensional indexes. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 1–4.

[2] Daichi Amagata, Yusuke Arai, Sumio Fujita, and Takahiro Hara. 2022. Learned k-nn distance estimation. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–4.

[3] Lance Ashdown et al. 2024. Oracle Database SQL Tuning Guide, 19c. https://docs.oracle.com/en/database/oracle/oracle-database/19/tgsql/index.html

[4] Qiushi Bai, Sadeem Alsudais, Chen Li, and Shuang Zhao. 2023. Maliva: Using Machine Learning to Rewrite Visualization Queries Under Time Constraints.. In *EDBT*. 157–170.

[5] Alberto Belussi et al. 2022. Spatial embedding: a generic machine learning model for spatial query optimization. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–4.

[6] Tsz Nam Chan, Leong Hou U, Byron Choi, Jianliang Xu, and Reynold Cheng. 2023. Large-scale geospatial analytics: Problems, challenges, and opportunities. In *Companion of the 2023 International Conference on Management of Data*. 21–29.

[7] Anderson Chaves Carniel. 2023. Defining and designing spatial queries: the role of spatial relationships. *Geo-spatial Information Science* (2023), 1–25.

[8] Xu Chen, Haitian Chen, Zibo Liang, Shuncheng Liu, Jinghong Wang, Kai Zeng, Han Su, and Kai Zheng. 2023. Leon: A new framework for ml-aided query optimization. *Proceedings of the VLDB Endowment* 16, 9 (2023), 2261–2273.

[9] Graham Cormode, Minos Garofalakis, Peter J Haas, Chris Jermaine, et al. 2011. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends® in Databases* 4, 1–3 (2011), 1–294.

[10] Haowen Dong, Chengliang Chai, Yuyu Luo, Jiabin Liu, Jianhua Feng, and Chaoqun Zhan. 2022. Rw-tree: A learned workload-aware framework for r-tree construction. In *ICDE*. IEEE, 2073–2085.

[11] Anshuman Dutt, Chi Wang, Azade Nazi, Srikanth Kandula, Vivek Narasayya, and Surajit Chaudhuri. 2019. Selectivity estimation for range predicates using lightweight models. *Proceedings of the VLDB Endowment* 12, 9 (2019), 1044–1057.

[12] Ahmed Eldawy and Mohamed F Mokbel. 2017. The era of big spatial data. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1992–1995.

[13] Ahmed Eldawy, Mohamed F Mokbel, et al. 2016. The era of big spatial data: A survey. *Foundations and Trends® in Databases* 6, 3-4 (2016), 163–273.

[14] Tu Gu, Kaiyu Feng, Gao Cong, Cheng Long, Zheng Wang, and Sheng Wang. 2023. The rlr-tree: A reinforcement learning based r-tree for spatial data. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–26.

[15] Jayant R Haritsa. 2019. Robust query processing: Mission possible. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2072–2075.

[16] Xiao Hu, Yuxi Liu, Haibo Xiu, Pankaj K Agarwal, Debmalya Panigrahi, Sudeepa Roy, and Jun Yang. 2022. Selectivity functions of range queries are learnable. In *Proceedings of the 2022 International Conference on Management of Data*. 959–972.

[17] Kyoungmin Kim, Sangoh Lee, Injung Kim, and Wook-Shin Han. 2024. ASM: Harmonizing Autoregressive Model, Sampling, and Multi-dimensional Statistics Merging for Cardinality Estimation. *PACMOD* 2, 1 (2024).

[18] Guoliang Li, Xuanhe Zhou, and Lei Cao. 2021. AI meets database: AI4DB and DB4AI. In *Proceedings of the 2021 International Conference on Management of Data*. 2859–2866.

[19] Pengfei Li, Hua Lu, Qian Zheng, Long Yang, and Gang Pan. 2020. LISA: A learned index structure for spatial data. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. 2119–2133.

[20] Pengfei Li, Wenqing Wei, Rong Zhu, Bolin Ding, Jingren Zhou, and Hua Lu. 2023. ALECE: An Attention-based Learned Cardinality Estimator for SPJ Queries on Dynamic Workloads. *Proceedings of the VLDB Endowment* 17, 2 (2023), 197–210.

[21] Feng Liang, Francis CM Lau, Heming Cui, Yupeng Li, Bing Lin, Chengming Li, and Xiping Hu. 2024. RelJoin: Relative-cost-based selection of distributed join methods for query plan optimization. *Information Sciences* 658 (2024), 120022.

[22] Guanli Liu, Jianzhong Qi, Christian S Jensen, James Bailey, and Lars Kulik. 2023. Efficiently learning spatial indices. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1572–1584.

[23] Ryan Marcus et al. 2021. Flow-loss: Learning cardinality estimates that matter. *Proceedings of the VLDB Endowment* 14, 11 (2021).

[24] Eduardo Mella, M Andrea Rodríguez, Loreto Bravo, and Diego Gatica. 2019. Query rewriting for semantic query optimization in spatial databases. *GeoInformatica* 23 (2019), 79–104.

[25] Parimarjan Negi, Ziniu Wu, Andreas Kipf, Nesime Tatbul, Ryan Marcus, Sam Madden, Tim Kraska, and Mohammad Alizadeh. 2023. Robust query driven cardinality estimation under changing workloads. *Proceedings of the VLDB Endowment* 16, 6 (2023), 1520–1533.

[26] Yongjoo Park, Shucheng Zhong, and Barzan Mozafari. 2020. Quicksel: Quick selectivity learning with mixture models. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1017–1033.

[27] Jianzhong Qi, Guanli Liu, Christian S Jensen, and Lars Kulik. 2020. Effectively learning spatial indices. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2341–2354.

[28] Silvan Reiner and Michael Grossniklaus. 2024. Sample-Efficient Cardinality Estimation Using Geometric Deep Learning. *Proc. VLDB Endow.* 17, 4 (mar 2024), 740–752.

[29] Ibrahim Sabek and Mohamed F Mokbel. 2020. Machine learning meets big spatial data. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1782–1785.

[30] Ibrahim Sabek, Mashaal Musleh, and Mohamed F Mokbel. 2018. TurboReg: a framework for scaling up spatial logistic regression models. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*. 129–138.

[31] Shashi Shekhar Shekhar and Sanjay Chawla. 2003. *Spatial Databases: A Tour*. Prentice Hall.

[32] Ji Sun, Jintao Zhang, Zhaoyan Sun, Guoliang Li, and Nan Tang. 2021. Learned cardinality estimation: A design space exploration and a comparative evaluation. *Proceedings of the VLDB Endowment* 15, 1 (2021), 85–97.

[33] Dimitris Tsesmelis and Alkis Simitsis. 2022. Database optimizers in the era of learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 3213–3216.

[34] Tin Vu, Alberto Belussi, Sara Migliorini, and Ahmed Eldawy. 2021. A Learned Query Optimizer for Spatial Join. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 458–467.

[35] Tin Vu, Alberto Belussi, Sara Migliorini, and Ahmed Eldawy. 2024. A learning-based framework for spatial join processing: estimation, optimization and tuning. *The VLDB Journal* (2024), 1–23.

[36] Brett Walenz, Stavros Sintos, Sudeepa Roy, and Jun Yang. 2019. Learning to sample: counting with complex queries. *Proceedings of the VLDB Endowment* 13, 3 (2019), 390–402.

[37] Lijun Wang, Linshu Hu, Chenhua Fu, Yuhan Yu, Peng Tang, Feng Zhang, and Renyi Liu. 2023. SLBRIN: A Spatial Learned Index Based on BRIN. *ISPRS International Journal of Geo-Information* 12, 4 (2023), 171.

[38] Yaoshu Wang, Chuan Xiao, Jianbin Qin, Rui Mao, Makoto Onizuka, Wei Wang, Rui Zhang, and Yoshiharu Ishikawa. 2021. Consistent and flexible selectivity estimation for high-dimensional data. In *Proceedings of the 2021 International Conference on Management of Data*. 2319–2327.

[39] Zhaoguo Wang, Zhou Zhou, Yicun Yang, Haoran Ding, Gansen Hu, Ding Ding, Chuzhe Tang, Haibo Chen, and Jinyang Li. 2022. Wetune: Automatic discovery and verification of query rewrite rules. In *Proceedings of the 2022 International Conference on Management of Data*. 94–107.

[40] Lucas Woltmann, Jerome Thiessat, Claudio Hartmann, Dirk Habich, and Wolfgang Lehner. 2023. Fastgres: Making learned query optimizer hinting effective. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3310–3322.

[41] Peizhi Wu and Gao Cong. 2021. A unified deep model of learning from both data and queries for cardinality estimation. In *Proceedings of the 2021 International Conference on Management of Data*. 2009–2022.

[42] Ziniu Wu, Parimarjan Negi, Mohammad Alizadeh, Tim Kraska, and Samuel Madden. 2023. FactorJoin: a new cardinality estimation framework for join queries. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[43] Zhengtong Yan, Valter Uotila, and Jiaheng Lu. 2023. Join Order Selection with Deep Reinforcement Learning: Fundamentals, Techniques, and Challenges. *Proceedings of the VLDB Endowment* 16, 12 (2023), 3882–3885.

[44] Xiang Yu, Chengliang Chai, Guoliang Li, and Jiabin Liu. 2022. Cost-based or learning-based? A hybrid query optimizer for query plan selection. *Proceedings of the VLDB Endowment* 15, 13 (2022), 3924–3936.

[45] Alaa Aldin Zein, Salah Dowaji, and Mohamad Iyad Al-Khayatt. 2023. Clustering-based method for big spatial data partitioning. *Measurement: Sensors* 27 (2023), 100731.

[46] Meifan Zhang and Hongzhi Wang. 2021. Selectivity estimation with density-model-based multidimensional histogram. *Knowledge and Information Systems* 63 (2021), 971–992.

[47] Songnian Zhang, Suprio Ray, Rongxing Lu, and Yandong Zheng. 2022. Efficient learned spatial index with interpolation function based learned model. *IEEE Transactions on Big Data* 9, 2 (2022), 733–745.

[48] Kangfei Zhao, Jeffrey Xu Yu, Zongyan He, Rui Li, and Hao Zhang. 2022. Lightweight and accurate cardinality estimation by neural network gaussian process. In *Proceedings of the 2022 International Conference on Management of Data*. 973–987.

[49] Yue Zhao, Gao Cong, Jiachen Shi, and Chunyan Miao. 2022. Queryformer: A tree transformer model for query plan representation. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1658–1670.

[50] Johan Kok Zhi Kang et al. 2021. Efficient deep learning pipelines for accurate cost estimations over large scale query workload. In *SIGMOD*. 1014–1022.

[51] Xuanhe Zhou, Guoliang Li, Chengliang Chai, and Jianhua Feng. 2021. A learned query rewrite system using monte carlo tree search. *Proceedings of the VLDB Endowment* 15, 1 (2021), 46–58.

[52] Rong Zhu, Ziniu Wu, Yuxing Han, Kai Zeng, Andreas Pfadler, Zhengping Qian, Jingren Zhou, and Bin Cui. 2021. FLAT: fast, lightweight and accurate method for cardinality estimation. *Proceedings of the VLDB Endowment* 14, 9 (2021), 1489–1502.