Learned Query Optimizers

Other titles in Foundations and Trends® in Databases

More Modern B-Tree Techniques
Goetz Graefe
ISBN: 978-1-63828-372-0

Data Structures for Data-Intensive Applications: Tradeoffs and Design Guidelines
Manos Athanassoulis, Stratos Idreos and Dennis Shasha
ISBN: 978-1-63828-184-9

Consensus in Data Management: From Distributed Commit to Blockchain
Faisal Nawab and Mohammad Sadoghi
ISBN: 978-1-63828-160-3

Multidimensional Array Data Management
Florin Rusu
ISBN: 978-1-63828-148-1

Learned Query Optimizers

Bolin Ding
Alibaba Group
bolin.ding@alibaba-inc.com

Rong Zhu
Alibaba Group
red.zr@alibaba-inc.com

Jingren Zhou
Alibaba Group
jingren.zhou@alibaba-inc.com

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Databases

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

B. Ding, *et al.*. *Learned Query Optimizers*. Foundations and Trends® in Databases, vol. 13, no. 4, pp. 250–310, 2024.

ISBN: 978-1-63828-383-6

Full text available at: http://dx.doi.org/10.1561/1900000082

Foundations and Trends® in Databases
Volume 13, Issue 4, 2024

Editorial Board

Editorial Scope

Foundations and Trends® in Databases publishes survey and tutorial articles in the following topics:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians

Foundations and Trends® in Databases, 2024, Volume 13, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.

Contents

Learned Query Optimizers

Bolin Ding, Rong Zhu and Jingren Zhou

Alibaba Group, China; bolin.ding@alibaba-inc.com, red.zr@alibaba-inc.com, jingren.zhou@alibaba-inc.com

ABSTRACT

This survey presents recent progress on using machine learning techniques to improve query optimizers in database systems. Centering around a generic paradigm of *learned query optimizers*, this survey covers several lines of effort on rebuilding or aiding important components in query optimizers (*i.e.*, *cardinality estimators*, *cost models*, and *plan enumerators*) with machine learning. We introduce some important machine learning tools developed recently, which are useful for query optimization, and how they are adapted for sub-tasks of query optimization. This survey is for readers who are already familiar with query optimization and are eager to understand what machine learning techniques can be helpful and how to apply them with examples and necessary details, or for machine learning researchers who want to expand their research agendas to helping database systems with machine learning techniques. Some open research challenges are also discussed with the goal of making learned query optimizers truly applicable in production.

Bolin Ding, Rong Zhu and Jingren Zhou (2024), "Learned Query Optimizers", Foundations and Trends® in Databases: Vol. 13, No. 4, pp 250–310. DOI: 10.1561/1900000082.

1

Introduction

1.1 Basics of Query Optimization

Query optimizers play one of the most important roles in database systems. It aims to select an efficient execution plan for a query written in a declarative language, *e.g.*, SQL. Traditional cost-based query optimizers (Selinger *et al.*, 1979; Graefe and McKenna, 1993; Graefe, 1995) find the plan with the minimum estimated *cost* for the given query.

Let's start with some notations that will be used throughout this survey. A relational database $\mathbb{D}$ consists of a set of base relations (tables), $\{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{|\mathbb{D}|}\}$. A query q accesses and manipulates data in the database via relational operations, *e.g.*, *select*, *project*, *join*, and *aggregate*. There are usually a large number of ways to process a query q, called *physical query execution plans* (denoted as P) or *plans* for short in the rest of this survey, with different choices of *join ordering* (which relations are joined first), *join operators* (*e.g.*, hash join $\bowtie_{\texttt{H}}$ and indexed nested loop join $\bowtie_{\texttt{INL}}$), and *access paths* (different ways to retrieve tuples from relations, *e.g.*, index seek `IdxSeek` and sequential scan `SeqScan`). For example, to process the query $q = \mathbf{R} \bowtie \mathbf{S} \bowtie \mathbf{T}$,

$$P = (\texttt{IdxSeek}(\mathbf{R}) \bowtie_{\texttt{INL}} \texttt{SeqScan}(\mathbf{S})) \bowtie_{\texttt{H}} \texttt{SeqScan}(\mathbf{T}) \qquad (1.1)$$

is a plan which joins relations **R** and **S** first with an indexed nested loop join and then joins the result of $\mathbf{R} \bowtie \mathbf{S}$ with **T** with a hash join.

For a query q, let $\mathbb{P}(q)$ be the set of all valid plans. The goal of query optimization is to select the most "efficient" plan P^* from $\mathbb{P}(q)$.

A *cost model* in a cost-based query optimizer (*e.g.*, Selinger *et al.*, 1979) measures the "efficiency" of a plan in terms of the execution latency or other user-specified metrics about resource consumption for the plan to be executed. The cost estimates derived from cost models are in forms of formulas with cardinalities of sub-queries as variables as well as some magic constant numbers to approximate the actual execution latency of the plan. These formulas and magic constant numbers depend on the algorithmic complexities and implementations of physical operators (*e.g.*, various join algorithms). The cardinalities of sub-queries are the sizes of inputs to these physical operators and are unknown before a query is executed. Thus, their estimates are obtained with *cardinality estimators* and fed into the cost model.

A *plan enumerator* is a cost-based search algorithm that explores the plan space and aims to find the one with the minimal (estimated) cost based on, *e.g.*, transformation rules or dynamic programming.

Figure 1.1 (excluding the shaded parts) gives an overview about how the three components, cost model, cardinality estimator, and plan enumerator, work together in a query optimizer.

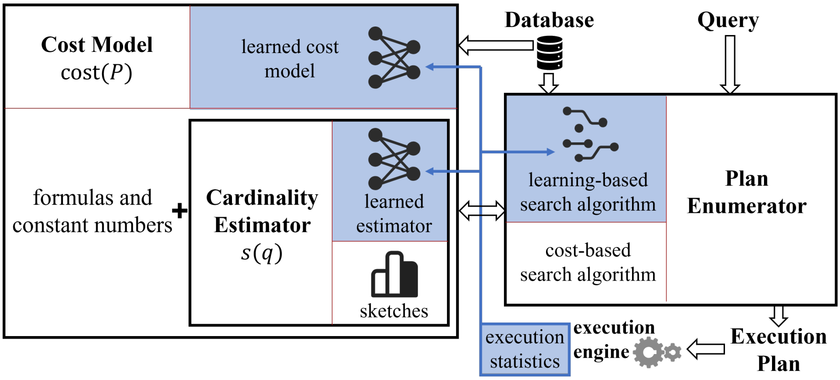

Figure 1.1: Overview of (learned) query optimizers.

While an obvious challenge in building a query optimizer is that the size of $\mathbb{P}(q)$ is exponential in the number of relations involved in q and the number of operator types, more uncertainty comes from the traditional cost model which depends on cardinality estimates for sub-queries, and quantitative models for costing query processing operators. Various heuristics and assumptions are essential in deriving these cardinality/cost estimates. For example, independence between attributes across relations is assumed and utilized for estimating cardinalities of joins of multiple relations (Tzoumas *et al.*, 2011; Leis *et al.*, 2015); magic constant numbers are prevalent in cost models, and they are often calibrated and tuned over years to ensure that the estimated cost matches the plan's performance well empirically, under certain system and hardware configurations though. It has been realized that such heuristics and assumptions are not always reliable for varying data distributions (especially on skewed and correlated data) or system configurations. As a result, cost models may produce significant errors and the plan generated by the traditional query optimizer may have poor quality (Doraiswamy *et al.*, 2008; Han *et al.*, 2021).

1.2 Why a Learned Optimizer is Possible

There are a recent line of efforts to assist or rebuild these components in query optimizers with machine learning models, which are trained on a specific dataset and "previous experience" collected from executing queries in the same or historical workloads. Such attempts date back to 2000s, *e.g.*, DB2's LEarning Optimizer Leo (Stillger *et al.*, 2001).

From the perspective of machine learning and optimization, the tasks tackled by cardinality estimator and cost model are *regression problems* (predicting cardinalities and costs of sub-queries and plans, respectively) and the one by plan enumerator is a *decision-making problem* (finding the best execution plan). With the recent progresses on deep models (*e.g.*, Mou *et al.*, 2016; Vaswani *et al.*, 2017) and deep reinforcement learning (*e.g.*, Sutton and Barto, 2018), we have more powerful tools for these two types of tasks.

For example, an execution plan for a SQL query is a tree structure representing the join order with each node in the tree specifying a physical operator and its two children specifying the two input relations. From the perspective of machine learning, it is non-trivial to map the plans with varying sizes into a regularized feature space while encoding both the plans' structural and node-wise information. The *tree convolution network* (Mou *et al.*, 2016) and the *attention mechanism* (Vaswani *et al.*, 2017) are two tools (though invented for different purposes) that are able to featurize such complex objects and judiciously utilize their structural information for the prediction task.

Specifically, two types of distributions are important for selecting efficient execution plans: i) data distributions over single and multiple relations (*e.g.*, deciding the join sizes), and ii) joint distribution over relations and query workloads (*e.g.*, deciding the selectivity of predicates). Traditional query optimizers rely on histograms and samples to approximate distributions in i) and ii) (refer to, *e.g.*, the survey by Cormode *et al.*, 2012) for the purposes of cardinality and cost estimation. Machine learning models trained on the targeting datasets and workloads may serve as their replacements, and indeed, the models need to be continuously updated when datasets and workloads are dynamic.

1.3 A Generic Paradigm of Learned Query Optimizers

Figure 1.1 illustrates how the three major components (*i.e.*, cost model, cardinality estimator, and plan enumerator) in a query optimizer can be replaced or enhanced with machine learning models (the shaded parts). Modeling more complex and high-dimensional data-query distributions and utilizing feedback/statistics from query executions are where the opportunities lie for these machine-learned counterparts to further improve the performance of query optimizers. To this end, we need to collect training data for these models, from both the databases and the execution engine that processes the query workloads, and organize the training data according to the goals of different models (in *learned cardinality estimator*, *learned cost model*, and *learning-based search algorithm*). Most previous works on learning to optimize queries do not rebuild the whole optimizer. Instead, they focus on one or multiple of these machine-learned counterparts, without a clear separation between different components (especially in reinforcement learning), and integrate them into a traditional query optimizer in a holistic way.

From sketches to learned cardinality estimator. For the task of cardinality estimation for (sub-)queries, there are two types of machine-learning based approaches, *data driven estimator* and *data-query jointly driven estimator*, both of which can be plugged into traditional cost models.

The former uses statistical and deep models (*e.g.*, deep autoregressive model) to approximate high-dimensional data distributions over database attributes and relations. The training and usage of such models can be analogous to how the traditional sketches (*e.g.*, histograms and samples) are constructed and used. They are trained on samples drawn from relations with the goal of minimizing the gap between the predicted data distribution and the seen distribution. Query workloads are assumed to be unknown when fitting these models. For a given query, these models are "invoked" to estimate its cardinality.

The latter trains models for a specific query workload for better accuracy. Queries are featurized as parts of the inputs to the model, and the model is trained to minimize the gap between the estimated cardinalities and the true cardinalities. Indeed, the model needs to be updated when the distribution of query workload shifts.

From traditional cost model to learned cost model. The cost of a plan is the sum of costs of all operators in it. For each operator, a traditional cost model typically takes cardinality estimates of immediate sub-queries under the operator as the inputs in a formula to estimate its cost, since they are the numbers of tuples to be processed by this operator. The concrete form of this formula and the magic constants in it, depend on the operator's type and implementation, and are tuned with years of engineering efforts to ensure that the estimated cost matches the plan's performance well empirically. In this sense, traditional cost models are "human learning" models. It is thus a natural idea to develop machine learning models with execution statistics (for specific performance metrics) on different datasets and query workloads as the training data, to enable finer-grained characterization of various data distributions and system configurations, thus providing instance-level optimization of each query. The learned cost model can be plugged into the traditional cost-based search algorithm to cost (sub-)plans in the search procedure, and updated when more queries are processed.

Learning-based search algorithm. Traditional query optimizers treat the task of finding the best execution plan as a combinatorial optimization problem. Thus, dynamic programming algorithms as well as heuristics (*e.g.*, based on transformation rules) are developed to find the best plan under certain cost models. If we treat query optimization as a machine learning task, we unlock other possibilities of designing the search algorithm. For example, we can model it as a *multi-armed bandit* problem, where each arm corresponds to a candidate plan and we want to select the best arm (execution plan) with more and more observations of their performance. We can also model it as a *deep reinforcement learning* problem, with learned cost models as value networks to guide the generative search for the best plan. Moreover, since what we essentially need for query optimization is an oracle that compares two plans and ranks a set of candidate query plans with respect to their execution efficiency, we can model the task as a *learning-to-rank* problem. These possibilities will be introduced and formalized later in this survey.

Technical questions. There are some key technical questions to be resolved in the above paradigm. First, the data-query-workload joint distribution is complex. We need to carefully featurize data and queries in such a way that we can effectively model their correlation and the marginal distributions via, *e.g.*, statistical or deep models. Second, we need to collect "training data" for these models. Cold start is always a challenge, especially when we train models to estimate and optimize the execution latency. Third, learning-based search algorithms need to be co-designed with the estimation models, so that they have consistent learning goals; meanwhile, when a search algorithm invokes learned estimation models with non-trivial inference costs (possibly many times), it needs to be designed to avoid prohibitive optimization cost.

Other possibilities and tasks. The generic paradigm in Figure 1.1 rules out some other possible ways to find better plans by learning from experience. For example, one can execute plans on samples of relations and use such experience to refine cardinality estimates and thus improve the final execution plans (Krauthgamer *et al.*, 2008; Wu *et al.*, 2016). Even during the processing of a specific query, one can use early-stage experience (*e.g.*, try different operator types and join orders on samples from intermediate results) to revise the remaining execution plan (Kabra and DeWitt, 1998; Markl *et al.*, 2004; Kader *et al.*, 2009). Detailed discussion about these works is beyond the scope of this survey, but one can refer to a recent benchmark paper by Zhang *et al.* (2023) on such adaptive query processing algorithms.

Worst-case optimal join algorithms (refer to, *e.g.*, Ngo *et al.*, 2018) are set apart from traditional query processing algorithms with theoretical guarantees on their processing costs. Their practical performance also depends heavily on the order in which join attributes are processed, which is not reflected in the definition of worst-case optimality (w.r.t. worst-case assumptions about the database content) and the formal analysis by Ngo *et al.* (2018). Wang *et al.* (2023b) introduces a query engine which selects the attribute orders via reinforcement learning.

While the paradigm in Figure 1.1 matters primarily for join ordering, access path, and operator selection in query optimization, there are other tasks that can effectively improve the execution performance of a SQL query. For example, *query rewriting* is to transform a poorly-written SQL query into one that executes more efficiently while maintaining the result set. Approaches for this task are based on, *e.g.*, rules (Begoli *et al.*, 2018; Wang *et al.*, 2022), program synthesis (Dong *et al.*, 2023), Monte Carlo tree search with deep estimation models (Zhou *et al.*, 2021; Zhou *et al.*, 2023), or, more recently, large language models (Liu and Mozafari, 2024). These query rewriting approaches are orthogonal to the majority of techniques discussed in this survey.

There are some specific scenarios of query optimization that can be aided by machine learning but are not covered by this survey. For example, *multi-query optimization* aims to select plans for a group of queries, considering opportunities to reduce the total execution cost by sharing redundant work to be done by an identical sub-query across plans of different queries. This problem can be tackled with, *e.g.*, reinforcement learning by Sioulas and Ailamaki (2021). *Parametric query optimization*, addressed by, *e.g.*, Doshi *et al.* (2023), is to generate a set of candidate plans for a single query template and decide which plan to use for each query instance. Learned query optimization for specialized types of data such as spatial data is also studied in Vu *et al.* (2021).

1.4 Summary of the Survey

In a learned query optimizer, one or multiple core components are aided or rebuilt with machine learning techniques. Most state-of-the-art learned query optimizers can be regarded as concrete implementations of the aforementioned paradigm (Figure 1.1) or its variants. Section 2 will focus on representative techniques for the costing components (cardinality estimator and cost model). These two components are closely related, as in traditional query optimizers, cost models invoke cardinality estimators to cost plans. We will first discuss their relationship and how estimation error transfer from cardinality estimators to cost models. We will then introduce, purely data-driven as well as data-query jointly driven, machine learning techniques for cardinality estimation, followed by how to train machine learning models to cost plans directly. Section 3 will focus on plan enumerators. Several new types of search algorithms, empowered by machine learning models, are proposed recently. Section 3.1 introduces a multi-armed bandit modeling of the plan enumeration procedure. Section 3.2 introduces how to apply generative search in reinforcement learning for (bottom-up) plan construction, with the help of value networks which is adapted from learned cost models. Section 3.3 introduces a learning-to-rank scheme for plan enumeration and selection. We will also discuss interesting future research directions inspired by some more recent efforts in Section 4.

References

Akdere, M., U. Çetintemel, M. Riondato, E. Upfal, and S. B. Zdonik. (2012). "Learning-based Query Performance Modeling and Prediction". In: *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*. IEEE Computer Society. 390–401. DOI: 10.1109/ICDE.2012.64.

Anneser, C., N. Tatbul, D. E. Cohen, Z. Xu, P. Pandian, N. Laptev, and R. Marcus. (2023). "AutoSteer: Learned Query Optimization for Any SQL Database". *Proc. VLDB Endow.* 16(12): 3515–3527. DOI: 10.14778/3611540.3611544.

Atserias, A., M. Grohe, and D. Marx. (2013). "Size Bounds and Query Plans for Relational Joins". *SIAM J. Comput.* 42(4): 1737–1767. DOI: 10.1137/110859440.

Begoli, E., J. Camacho-Rodríguez, J. Hyde, M. J. Mior, and D. Lemire. (2018). "Apache Calcite: A Foundational Framework for Optimized Query Processing Over Heterogeneous Data Sources". In: *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. ACM. 221–230. DOI: 10.1145/3183713.3190662.

Cai, W., M. Balazinska, and D. Suciu. (2019). “Pessimistic Cardinality Estimation: Tighter Upper Bounds for Intermediate Join Cardinalities”. In: *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. ACM. 18–35. DOI: 10.1145/3299869.3319894.

Charikar, M., S. Chaudhuri, R. Motwani, and V. R. Narasayya. (2000). “Towards Estimation Error Guarantees for Distinct Values”. In: *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA*. ACM. 268–279. DOI: 10.1145/335168.335230.

Chen, T., J. Gao, H. Chen, and Y. Tu. (2023a). “LOGER: A Learned Optimizer towards Generating Efficient and Robust Query Execution Plans”. *Proc. VLDB Endow.* 16(7): 1777–1789. URL: https://www.vldb.org/pvldb/vol16/p1777-gao.pdf.

Chen, X., Z. Wang, S. Liu, Y. Li, K. Zeng, B. Ding, J. Zhou, H. Su, and K. Zheng. (2023b). “BASE: Bridging the Gap between Cost and Latency for Query Optimization”. *Proc. VLDB Endow.* 16(8): 1958–1966. URL: https://www.vldb.org/pvldb/vol16/p1958-chen.pdf.

Cormode, G., M. N. Garofalakis, P. J. Haas, and C. Jermaine. (2012). “Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches”. *Found. Trends Databases.* 4(1-3): 1–294. DOI: 10.1561/1900000004.

Davitkova, A., D. Gjurovski, and S. Michel. (2022). “LMKG: Learned Models for Cardinality Estimation in Knowledge Graphs”. In: *Proceedings of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 - April 1, 2022*. OpenProceedings.org. 2:169–2:182. DOI: 10.48786/edbt.2022.07.

Dey, A., S. Bhaumik, H. Doraiswamy, and J. R. Haritsa. (2008). “Efficiently approximating query optimizer plan diagrams”. *Proc. VLDB Endow.* 1(2): 1325–1336. DOI: 10.14778/1454159.1454173.

Ding, B., S. Das, R. Marcus, W. Wu, S. Chaudhuri, and V. R. Narasayya. (2019). "AI Meets AI: Leveraging Query Executions to Improve Index Recommendations". In: *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. ACM. 1241–1258. DOI: 10.1145/3299869.3324957.

Dong, R., J. Liu, Y. Zhu, C. Yan, B. Mozafari, and X. Wang. (2023). "SlabCity: Whole-Query Optimization using Program Synthesis". *Proc. VLDB Endow.* 16(11): 3151–3164. DOI: 10.14778/3611479.3611515.

Doraiswamy, H., P. N. Darera, and J. R. Haritsa. (2008). "Identifying robust plans through plan diagram reduction". *Proc. VLDB Endow.* 1(1): 1124–1140. DOI: 10.14778/1453856.1453976.

Doshi, L., V. Zhuang, G. Jain, R. Marcus, H. Huang, D. Altinbüken, E. Brevdo, and C. Fraser. (2023). "Kepler: Robust Learning for Parametric Query Optimization". *Proc. ACM Manag. Data.* 1(1): 109:1–109:25. DOI: 10.1145/3588963.

Duggan, J., U. Çetintemel, O. Papaemmanouil, and E. Upfal. (2011). "Performance prediction for concurrent database workloads". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*. ACM. 337–348. DOI: 10.1145/1989323.1989359.

Durkan, C. and C. Nash. (2019). "Autoregressive Energy Machines". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Vol. 97. *Proceedings of Machine Learning Research*. PMLR. 1735–1744. URL: http://proceedings.mlr.press/v97/durkan19a.html.

Dutt, A., C. Wang, V. R. Narasayya, and S. Chaudhuri. (2020). "Efficiently Approximating Selectivity Functions using Low Overhead Regression Models". *Proc. VLDB Endow.* 13(11): 2215–2228. URL: http://www.vldb.org/pvldb/vol13/p2215-dutt.pdf.

Dutt, A., C. Wang, A. Nazi, S. Kandula, V. R. Narasayya, and S. Chaudhuri. (2019). "Selectivity Estimation for Range Predicates using Lightweight Models". *Proc. VLDB Endow.* 12(9): 1044–1057. DOI: 10.14778/3329772.3329780.

Germain, M., K. Gregor, I. Murray, and H. Larochelle. (2015). "MADE: Masked Autoencoder for Distribution Estimation". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. Vol. 37. *JMLR Workshop and Conference Proceedings*. JMLR.org. 881–889. URL: http://proceedings.mlr.press/v37/germain15.html.

Getoor, L., B. Taskar, and D. Koller. (2001). "Selectivity Estimation using Probabilistic Models". In: *Proceedings of the 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, CA, USA, May 21-24, 2001*. ACM. 461–472. DOI: 10.1145/375663.375727.

Goodrich, M. T., R. Tamassia, and M. H. Goldwasser. (2013). *Data Structures and Algorithms in Python*. Wiley.

Graefe, G. (1995). "The Cascades Framework for Query Optimization". *IEEE Data Eng. Bull.* 18(3): 19–29. URL: http://sites.computer.org/debull/95SEP-CD.pdf.

Graefe, G. and W. J. McKenna. (1993). "The Volcano Optimizer Generator: Extensibility and Efficient Search". In: *Proceedings of the Ninth International Conference on Data Engineering, April 19-23, 1993, Vienna, Austria*. IEEE Computer Society. 209–218. DOI: 10.1109/ICDE.1993.344061.

Gregor, K., I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra. (2014). "Deep AutoRegressive Networks". In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. *JMLR Workshop and Conference Proceedings*. JMLR.org. 1242–1250. URL: http://proceedings.mlr.press/v32/gregor14.html.

Grimmett, G. R. and D. R. Stirzaker. (2001). *Probability and Random Processes*. Oxford University Press.

Haas, L. M., M. J. Carey, M. Livny, and A. Shukla. (1997). "Seeking the Truth About ad hoc Join Costs". *VLDB J.* 6(3): 241–256. DOI: 10.1007/S007780050043.

Han, Y., Z. Wu, P. Wu, R. Zhu, J. Yang, L. W. Tan, K. Zeng, G. Cong, Y. Qin, A. Pfadler, Z. Qian, J. Zhou, J. Li, and B. Cui. (2021). "Cardinality Estimation in DBMS: A Comprehensive Benchmark Evaluation". *Proc. VLDB Endow.* 15(4): 752–765. DOI: 10.14778/3503585.3503586.

Hayek, R. and O. Shmueli. (2020). "Improved Cardinality Estimation by Learning Queries Containment Rates". In: *Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020.* OpenProceedings.org. 157–168. DOI: 10.5441/002/edbt.2020.15.

Heimel, M., M. Kiefer, and V. Markl. (2015). "Self-Tuning, GPU-Accelerated Kernel Density Models for Multidimensional Selectivity Estimation". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015.* ACM. 1477–1492. DOI: 10.1145/2723372.2749438.

Hertzschuch, A., C. Hartmann, D. Habich, and W. Lehner. (2021). "Simplicity Done Right for Join Ordering". In: *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings.* www.cidrdb.org. URL: http://cidrdb.org/cidr2021/papers/cidr2021%5C_paper01.pdf.

Hilprecht, B. and C. Binnig. (2022). "Zero-Shot Cost Models for Out-of-the-box Learned Cost Prediction". *Proc. VLDB Endow.* 15(11): 2361–2374. URL: https://www.vldb.org/pvldb/vol15/p2361-hilprecht.pdf.

Hilprecht, B., A. Schmidt, M. Kulessa, A. Molina, K. Kersting, and C. Binnig. (2020). "DeepDB: Learn from Data, not from Queries!" *Proc. VLDB Endow.* 13(7): 992–1005. DOI: 10.14778/3384345.3384349.

Izenov, Y., A. Datta, F. Rusu, and J. H. Shin. (2021). "COMPASS: Online Sketch-based Query Optimization for In-Memory Databases". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021.* ACM. 804–816. DOI: 10.1145/3448016.3452840.

Johannes, F. and H. Eyke. (2011). *Preference Learning.* Springer.

Jung, J., H. Hu, J. Arulraj, T. Kim, and W. Kang. (2019). "APOLLO: Automatic Detection and Diagnosis of Performance Regressions in Database Systems". *Proc. VLDB Endow.* 13(1): 57–70. DOI: 10.14778/3357377.3357382.

Kabra, N. and D. J. DeWitt. (1998). "Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans". In: *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*. ACM Press. 106–117. DOI: 10.1145/276304.276315.

Kader, R. A., P. A. Boncz, S. Manegold, and M. van Keulen. (2009). "ROX: run-time optimization of XQueries". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*. ACM. 615–626. DOI: 10.1145/1559845.1559910.

Kang, J. K. Z., Gaurav, S. Y. Tan, F. Cheng, S. Sun, and B. He. (2021). "Efficient Deep Learning Pipelines for Accurate Cost Estimations Over Large Scale Query Workload". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. Ed. by G. Li, Z. Li, S. Idreos, and D. Srivastava. ACM. 1014–1022. DOI: 10.1145/3448016.3457546.

Kiefer, M., M. Heimel, S. Breß, and V. Markl. (2017). "Estimating Join Selectivities using Bandwidth-Optimized Kernel Density Models". *Proc. VLDB Endow.* 10(13): 2085–2096. DOI: 10.14778/3151106.3151112.

Kim, K., J. Jung, I. Seo, W. Han, K. Choi, and J. Chong. (2022). "Learned Cardinality Estimation: An In-depth Study". In: *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. ACM. 1214–1227. DOI: 10.1145/3514221.3526154.

Kipf, A., T. Kipf, B. Radke, V. Leis, P. A. Boncz, and A. Kemper. (2019). "Learned Cardinalities: Estimating Correlated Joins with Deep Learning". In: *9th Biennial Conference on Innovative Data Systems Research, CIDR 2019, Asilomar, CA, USA, January 13-16, 2019, Online Proceedings*. www.cidrdb.org. URL: http://cidrdb.org/cidr2019/papers/p101-kipf-cidr19.pdf.

Krauthgamer, R., A. Mehta, V. Raman, and A. Rudra. (2008). "Greedy List Intersection". In: *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*. IEEE Computer Society. 1033–1042. DOI: 10.1109/ICDE.2008.4497512.

Krishnan, S., Z. Yang, K. Goldberg, J. M. Hellerstein, and I. Stoica. (2018). "Learning to Optimize Join Queries With Deep Reinforcement Learning". *CoRR*. abs/1808.03196. arXiv: 1808.03196. URL: http://arxiv.org/abs/1808.03196.

Kwon, S., W. Jung, and K. Shim. (2022). "Cardinality Estimation of Approximate Substring Queries using Deep Learning". *Proc. VLDB Endow.* 15(11): 3145–3157. URL: https://www.vldb.org/pvldb/vol15/p3145-jung.pdf.

Lan, H., Z. Bao, and Y. Peng. (2021). "A Survey on Advancing the DBMS Query Optimizer: Cardinality Estimation, Cost Model, and Plan Enumeration". *Data Sci. Eng.* 6(1): 86–101. DOI: 10.1007/S41019-020-00149-7.

Leis, V., A. Gubichev, A. Mirchev, P. A. Boncz, A. Kemper, and T. Neumann. (2015). "How Good Are Query Optimizers, Really?" *Proc. VLDB Endow.* 9(3): 204–215. DOI: 10.14778/2850583.2850594.

Leis, V., B. Radke, A. Gubichev, A. Kemper, and T. Neumann. (2017). "Cardinality Estimation Done Right: Index-Based Join Sampling". In: *8th Biennial Conference on Innovative Data Systems Research, CIDR 2017, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*. www.cidrdb.org. URL: http://cidrdb.org/cidr2017/papers/p9-leis-cidr17.pdf.

Li, F., B. Wu, K. Yi, and Z. Zhao. (2016). "Wander Join: Online Aggregation via Random Walks". In: *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*. ACM. 615–629. DOI: 10.1145/2882903.2915235.

Li, P., W. Wei, R. Zhu, B. Ding, J. Zhou, and H. Lu. (2023). "ALECE: An Attention-based Learned Cardinality Estimator for SPJ Queries on Dynamic Workloads". *Proc. VLDB Endow.* 17(2): 197–210. URL: https://www.vldb.org/pvldb/vol17/p197-li.pdf.

Li, Y., L. Wang, S. Wang, Y. Sun, and Z. Peng. (2022). "A Resource-Aware Deep Cost Model for Big Data Query Processing". In: *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE. 885–897. DOI: 10.1109/ICDE53745.2022.00071.

Lipton, R. J., J. F. Naughton, and D. A. Schneider. (1990). "Practical Selectivity Estimation through Adaptive Sampling". In: *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, USA, May 23-25, 1990*. ACM Press. 1–11. DOI: 10.1145/93597.93611.

Liu, H., M. Xu, Z. Yu, V. Corvinelli, and C. Zuzarte. (2015). "Cardinality estimation using neural networks". In: *Proceedings of 25th Annual International Conference on Computer Science and Software Engineering, CASCON 2015, Markham, Ontario, Canada, 2-4 November, 2015*. IBM / ACM. 53–59. URL: http://dl.acm.org/citation.cfm?id=2886453.

Liu, J., W. Dong, D. Li, and Q. Zhou. (2021a). "Fauce: Fast and Accurate Deep Ensembles with Uncertainty for Cardinality Estimation". *Proc. VLDB Endow.* 14(11): 1950–1963. DOI: 10.14778/3476249.3476254.

Liu, J. and B. Mozafari. (2024). "Query Rewriting via Large Language Models". *CoRR*. abs/2403.09060. DOI: 10.48550/ARXIV.2403.09060. arXiv: 2403.09060.

Liu, Q., Y. Shen, and L. Chen. (2021b). "LHist: Towards Learning Multi-dimensional Histogram for Massive Spatial Data". In: *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE. 1188–1199. DOI: 10.1109/ICDE51399.2021.00107.

Liu, S., X. Chen, Y. Zhao, J. Chen, R. Zhou, and K. Zheng. (2022). "Efficient Learning with Pseudo Labels for Query Cost Estimation". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. Ed. by M. A. Hasan and L. Xiong. ACM. 1309–1318. DOI: 10.1145/3511808.3557305.

Liu, T.-Y. (2009). "Learning to Rank for Information Retrieval". *Foundations and Trends in Information Retrieval*. 3(3): 225–331.

Loeliger, H. (2004). "An introduction to factor graphs". *IEEE Signal Process. Mag.* 21(1): 28–41. DOI: 10.1109/MSP.2004.1267047.

Lu, Y., S. Kandula, A. C. König, and S. Chaudhuri. (2021). "Pretraining Summarization Models of Structured Datasets for Cardinality Estimation". *Proc. VLDB Endow.* 15(3): 414–426. DOI: 10.14778/3494124.3494127.

Malik, T., R. C. Burns, and N. V. Chawla. (2007). "A Black-Box Approach to Query Cardinality Estimation". In: *Third Biennial Conference on Innovative Data Systems Research, CIDR 2007, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*. www.cidrdb.org. 56–67. URL: http://cidrdb.org/cidr2007/papers/cidr07p06.pdf.

Marcus, R., P. Negi, H. Mao, N. Tatbul, M. Alizadeh, and T. Kraska. (2021). "Bao: Making Learned Query Optimization Practical". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM. 1275–1288. DOI: 10.1145/3448016.3452838.

Marcus, R. and O. Papaemmanouil. (2018). "Deep Reinforcement Learning for Join Order Enumeration". In: *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, aiDM@SIGMOD 2018, Houston, TX, USA, June 10, 2018*. ACM. 3:1–3:4. DOI: 10.1145/3211954.3211957.

Marcus, R. C., P. Negi, H. Mao, C. Zhang, M. Alizadeh, T. Kraska, O. Papaemmanouil, and N. Tatbul. (2019). "Neo: A Learned Query Optimizer". *Proc. VLDB Endow.* 12(11): 1705–1718. DOI: 10.14778/3342263.3342644.

Marcus, R. C. and O. Papaemmanouil. (2019). "Plan-Structured Deep Neural Network Models for Query Performance Prediction". *Proc. VLDB Endow.* 12(11): 1733–1746. DOI: 10.14778/3342263.3342646.

Markl, V., V. Raman, D. E. Simmen, G. M. Lohman, and H. Pirahesh. (2004). "Robust Query Processing through Progressive Optimization". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*. ACM. 659–670. DOI: 10.1145/1007568.1007642.

Meng, Z., P. Wu, G. Cong, R. Zhu, and S. Ma. (2022). "Unsupervised Selectivity Estimation by Integrating Gaussian Mixture Models and an Autoregressive Model". In: *Proceedings of the 25th International Conference on Extending Database Technology, EDBT 2022, Edinburgh, UK, March 29 - April 1, 2022*. OpenProceedings.org. 2:247–2:259. DOI: 10.48786/edbt.2022.13.

Moerkotte, G. and T. Neumann. (2008). "Dynamic programming strikes back". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. ACM. 539–552. DOI: 10.1145/1376616.1376672.

Moerkotte, G., T. Neumann, and G. Steidl. (2009). "Preventing Bad Plans by Bounding the Impact of Cardinality Estimation Errors". *Proc. VLDB Endow.* 2(1): 982–993. DOI: 10.14778/1687627.1687738.

Mou, L., G. Li, L. Zhang, T. Wang, and Z. Jin. (2016). "Convolutional Neural Networks over Tree Structures for Programming Language Processing". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. AAAI Press. 1287–1293. DOI: 10.1609/AAAI.V30I1.10139.

Negi, P., M. Interlandi, R. Marcus, M. Alizadeh, T. Kraska, M. Friedman, and A. Jindal. (2021a). "Steering Query Optimizers: A Practical Take on Big Data Workloads". In: *SIGMOD*. 2557–2569.

Negi, P., M. Interlandi, R. Marcus, M. Alizadeh, T. Kraska, M. T. Friedman, and A. Jindal. (2021b). "Steering Query Optimizers: A Practical Take on Big Data Workloads". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM. 2557–2569. DOI: 10.1145/3448016.3457568.

Negi, P., R. C. Marcus, A. Kipf, H. Mao, N. Tatbul, T. Kraska, and M. Alizadeh. (2021c). "Flow-Loss: Learning Cardinality Estimates That Matter". *Proc. VLDB Endow.* 14(11): 2019–2032. DOI: 10.14778/3476249.3476259.

Negi, P., Z. Wu, A. Kipf, N. Tatbul, R. Marcus, S. Madden, T. Kraska, and M. Alizadeh. (2023). "Robust Query Driven Cardinality Estimation under Changing Workloads". *Proc. VLDB Endow.* 16(6): 1520–1533. URL: https://www.vldb.org/pvldb/vol16/p1520-negi.pdf.

Ngo, H. Q., E. Porat, C. Ré, and A. Rudra. (2018). "Worst-case Optimal Join Algorithms". *J. ACM*. 65(3): 16:1–16:40. DOI: 10.1145/3180143.

Ortiz, J., M. Balazinska, J. Gehrke, and S. S. Keerthi. (2018). "Learning State Representations for Query Optimization with Deep Reinforcement Learning". In: *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning, DEEM@SIGMOD 2018, Houston, TX, USA, June 15, 2018*. ACM. 4:1–4:4. DOI: 10.1145/3209889.3209890.

Osband, I. and B. V. Roy. (2015). "Bootstrapped Thompson Sampling and Deep Exploration". *CoRR*. abs/1507.00300. arXiv: 1507.00300. URL: http://arxiv.org/abs/1507.00300.

Park, Y., S. Zhong, and B. Mozafari. (2020). "QuickSel: Quick Selectivity Learning with Mixture Models". In: *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*. ACM. 1017–1033. DOI: 10.1145/3318464.3389727.

Poon, H. and P. M. Domingos. (2011). "Sum-Product Networks: A New Deep Architecture". In: *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*. AUAI Press. 337–346.

Poosala, V. and Y. E. Ioannidis. (1997). "Selectivity Estimation Without the Attribute Value Independence Assumption". In: *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*. Morgan Kaufmann. 486–495. URL: http://www.vldb.org/conf/1997/P486.PDF.

Selinger, P. G., M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. (1979). "Access Path Selection in a Relational Database Management System". In: *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data, Boston, Massachusetts, USA, May 30 - June 1*. ACM. 23–34. DOI: 10.1145/582095.582099.

Shetiya, S., S. Thirumuruganathan, N. Koudas, and G. Das. (2020). "Astrid: Accurate Selectivity Estimation for String Predicates using Deep Learning". *Proc. VLDB Endow.* 14(4): 471–484. DOI: 10.14778/3436905.3436907.

Siddiqui, T., A. Jindal, S. Qiao, H. Patel, and W. Le. (2020). "Cost Models for Big Data Query Processing: Learning, Retrofitting, and Our Findings". In: *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*. ACM. 99–113. DOI: 10.1145/3318464.3380584.

Sioulas, P. and A. Ailamaki. (2021). "Scalable Multi-Query Execution using Reinforcement Learning". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM. 1651–1663. DOI: 10.1145/3448016.3452799.

Sonoda, S. and N. Murata. (2017). "Neural Network with Unbounded Activation Functions is Universal Approximator". *Applied and Computational Harmonic Analysis*. 43(2): 233–268.

Stillger, M., G. M. Lohman, V. Markl, and M. Kandil. (2001). "LEO - DB2's LEarning Optimizer". In: *VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy*. Morgan Kaufmann. 19–28. URL: http://www.vldb.org/conf/2001/P019.pdf.

Sun, J. and G. Li. (2019). "An End-to-End Learning-based Cost Estimator". *Proc. VLDB Endow.* 13(3): 307–319. DOI: 10.14778/3368289.3368296.

Sutton, R. S. and A. G. Barto. (2018). *Reinforcement Learning*. The MIT Press.

Tai, K. S., R. Socher, and C. D. Manning. (2015). "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics. 1556–1566. DOI: 10.3115/V1/P15-1150.

Thompson, W. R. (1933). "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples". *Biometrika*. 25: 285–294.

Tzoumas, K., A. Deshpande, and C. S. Jensen. (2011). "Lightweight Graphical Models for Selectivity Estimation Without Independence Assumptions". *Proc. VLDB Endow.* 4(11): 852–863. URL: http://www.vldb.org/pvldb/vol4/p852-tzoumas.pdf.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vu, T., A. Belussi, S. Migliorini, and A. Eldawy. (2021). "A Learned Query Optimizer for Spatial Join". In: *SIGSPATIAL '21: 29th International Conference on Advances in Geographic Information Systems, Virtual Event / Beijing, China, November 2-5, 2021*. ACM. 458–467. DOI: 10.1145/3474717.3484217.

Wang, F., X. Yan, M. L. Yiu, S. LI, Z. Mao, and B. Tang. (2023a). "Speeding Up End-to-end Query Execution via Learning-based Progressive Cardinality Estimation". *Proc. ACM Manag. Data.* 1(1): 28:1–28:25. DOI: 10.1145/3588708.

Wang, J., C. Chai, J. Liu, and G. Li. (2021a). "FACE: A Normalizing Flow based Cardinality Estimator". *Proc. VLDB Endow.* 15(1): 72–84. DOI: 10.14778/3485450.3485458.

Wang, J., I. Trummer, A. Kara, and D. Olteanu. (2023b). "ADOPT: Adaptively Optimizing Attribute Orders for Worst-Case Optimal Join Algorithms via Reinforcement Learning". *Proc. VLDB Endow.* 16(11): 2805–2817. DOI: 10.14778/3611479.3611489.

Wang, X., C. Qu, W. Wu, J. Wang, and Q. Zhou. (2021b). "Are We Ready For Learned Cardinality Estimation?" *Proc. VLDB Endow.* 14(9): 1640–1654. DOI: 10.14778/3461535.3461552.

Wang, Z., Z. Zhou, Y. Yang, H. Ding, G. Hu, D. Ding, C. Tang, H. Chen, and J. Li. (2022). "WeTune: Automatic Discovery and Verification of Query Rewrite Rules". In: *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. ACM. 94–107. DOI: 10.1145/3514221.3526125.

Watkins, C. (1989). "Learning From Delayed Rewards". May.

Weng, L., R. Zhu, D. Wu, B. Ding, B. Zheng, and J. Zhou. (2024). "Eraser: Eliminating Performance Regression on Learned Query Optimizer". *Proc. VLDB Endow.* 17(5): 926–938.

Woltmann, L., J. Thiessat, C. Hartmann, D. Habich, and W. Lehner. (2023). "FASTgres: Making Learned Query Optimizer Hinting Effective". *Proc. VLDB Endow.* 16(11): 3310–3322. DOI: 10.14778/3611479.3611528.

Wu, C., A. Jindal, S. Amizadeh, H. Patel, W. Le, S. Qiao, and S. Rao. (2018). "Towards a Learning Optimizer for Shared Clouds". *Proc. VLDB Endow.* 12(3): 210–222. DOI: 10.14778/3291264.3291267.

Wu, P. and G. Cong. (2021). "A Unified Deep Model of Learning from both Data and Queries for Cardinality Estimation". In: *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. ACM. 2009–2022. DOI: 10.1145/3448016.3452830.

Wu, R., B. Ding, X. Chu, Z. Wei, X. Dai, T. Guan, and J. Zhou. (2021). "Learning to be a Statistician: Learned Estimator for Number of Distinct Values". *Proc. VLDB Endow.* 15(2): 272–284. DOI: 10.14778/3489496.3489508.

Wu, W., J. F. Naughton, and H. Singh. (2016). "Sampling-Based Query Re-Optimization". In: *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*. ACM. 1721–1736. DOI: 10.1145/2882903.2882914.

Wu, Z., P. Negi, M. Alizadeh, T. Kraska, and S. Madden. (2023). "FactorJoin: A New Cardinality Estimation Framework for Join Queries". *Proc. ACM Manag. Data.* 1(1): 41:1–41:27. DOI: 10.1145/3588721.

Wu, Z. and A. Shaikhha. (2020). "BayesCard: A Unified Bayesian Framework for Cardinality Estimation". *CoRR*. abs/2012.14743. arXiv: 2012.14743. URL: https://arxiv.org/abs/2012.14743.

Yan, S., B. Ding, W. Guo, J. Zhou, Z. Wei, X. Jiang, and S. Xu. (2021). "FlashP: An Analytical Pipeline for Real-time Forecasting of Time-Series Relational Data". *Proc. VLDB Endow.* 14(5): 721–729. DOI: 10.14778/3446095.3446096.

Yang, Z., W. Chiang, S. Luan, G. Mittal, M. Luo, and I. Stoica. (2022). "Balsa: Learning a Query Optimizer Without Expert Demonstrations". In: *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. Ed. by Z. G. Ives, A. Bonifati, and A. E. Abbadi. ACM. 931–944. DOI: 10.1145/3514221.3517885.

Yang, Z., A. Kamsetty, S. Luan, E. Liang, Y. Duan, X. Chen, and I. Stoica. (2020). "NeuroCard: One Cardinality Estimator for All Tables". *Proc. VLDB Endow.* 14(1): 61–73. DOI: 10.14778/3421424.3421432.

Yang, Z., E. Liang, A. Kamsetty, C. Wu, Y. Duan, X. Chen, P. Abbeel, J. M. Hellerstein, S. Krishnan, and I. Stoica. (2019). "Deep Unsupervised Cardinality Estimation". *Proc. VLDB Endow.* 13(3): 279–292. DOI: 10.14778/3368289.3368294.

Yu, X., G. Li, C. Chai, and N. Tang. (2020). "Reinforcement Learning with Tree-LSTM for Join Order Selection". In: *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE. 1297–1308. DOI: 10.1109/ICDE48307.2020.00116.

Yuan, H., G. Li, L. Feng, J. Sun, and Y. Han. (2020). "Automatic View Generation with Deep Learning and Reinforcement Learning". In: *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE. 1501–1512. DOI: 10.1109/ICDE48307.2020.00133.

Zhang, W., M. Interlandi, P. Mineiro, S. Qiao, N. Ghazanfari, K. Lie, M. T. Friedman, R. Hosn, H. Patel, and A. Jindal. (2022). "Deploying a Steered Query Optimizer in Production at Microsoft". In: *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. ACM. 2299–2311. DOI: 10.1145/3514221.3526052.

Zhang, Y., Y. Chronis, J. M. Patel, and T. Rekatsinas. (2023). "Simple Adaptive Query Processing vs. Learned Query Optimizers: Observations and Analysis". *Proc. VLDB Endow.* 16(11): 2962–2975. DOI: 10.14778/3611479.3611501.

Zhao, K., J. X. Yu, Z. He, R. Li, and H. Zhang. (2022a). "Lightweight and Accurate Cardinality Estimation by Neural Network Gaussian Process". In: *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*. ACM. 973–987. DOI: 10.1145/3514221.3526156.

Zhao, Y., G. Cong, J. Shi, and C. Miao. (2022b). "QueryFormer: A Tree Transformer Model for Query Plan Representation". *Proc. VLDB Endow.* 15(8): 1658–1670. URL: https://www.vldb.org/pvldb/vol15/p1658-zhao.pdf.

Zhao, Z., R. Christensen, F. Li, X. Hu, and K. Yi. (2018). "Random Sampling over Joins Revisited". In: *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. ACM. 1525–1539. DOI: 10.1145/3183713.3183739.

Zhou, J., N. Bruno, M. Wu, P. Larson, R. Chaiken, and D. Shakib. (2012). "SCOPE: parallel databases meet MapReduce". *VLDB J.* 21(5): 611–636. DOI: 10.1007/S00778-012-0280-Z.

Zhou, X., G. Li, C. Chai, and J. Feng. (2021). "A Learned Query Rewrite System using Monte Carlo Tree Search". *Proc. VLDB Endow.* 15(1): 46–58. DOI: 10.14778/3485450.3485456.

Zhou, X., G. Li, J. Wu, J. Liu, Z. Sun, and X. Zhang. (2023). "A Learned Query Rewrite System". *Proc. VLDB Endow.* 16(12): 4110–4113. DOI: 10.14778/3611540.3611633.

Zhou, X., J. Sun, G. Li, and J. Feng. (2020). "Query Performance Prediction for Concurrent Queries using Graph Embedding". *Proc. VLDB Endow.* 13(9): 1416–1428. DOI: 10.14778/3397230.3397238.

Zhu, R., W. Chen, B. Ding, X. Chen, A. Pfadler, Z. Wu, and J. Zhou. (2023). "Lero: A Learning-to-Rank Query Optimizer". *Proc. VLDB Endow.* 16(6): 1466–1479. URL: https://www.vldb.org/pvldb/vol16/p1466-zhu.pdf.

Zhu, R., L. Weng, W. Wei, D. Wu, J. Peng, Y. Wang, B. Ding, D. Lian, B. Zheng, and J. Zhou. (2024). "PilotScope: Steering Databases with Machine Learning Drivers". *Proc. VLDB Endow.* 17(5): 980–993.

Zhu, R., Z. Wu, Y. Han, K. Zeng, A. Pfadler, Z. Qian, J. Zhou, and B. Cui. (2021). "FLAT: Fast, Lightweight and Accurate Method for Cardinality Estimation". *Proc. VLDB Endow.* 14(9): 1489–1502. DOI: 10.14778/3461535.3461539.