How Good Are Learned Cost Models, Really? Insights From Query Optimization Tasks

ACM SIGMOD 2025

Roman Heinrich, Manisha Luthra, Johannes Wehrstein,
Harald Kornmayer, Carsten Binnig

Cost Estimation is Crucial for Databases

Query Optimization | **Resource Allocation** | **Query Scheduling** | **Index & MV Advisors**

Hand-Crafted Cost Functions

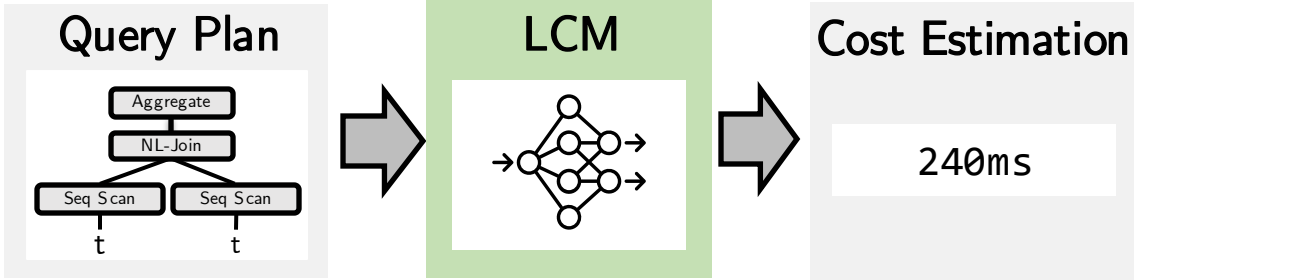
$costs = c1 * p_{seq} + c2 * p_{random} + \ldots$

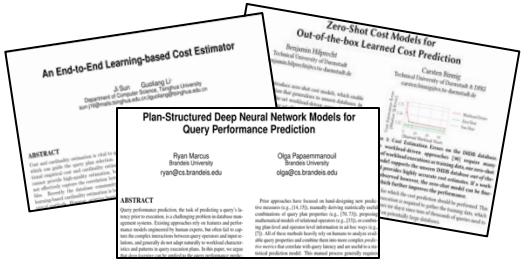
❌ **Often misestimate costs → loosing optimization potential**

⚠ **We need more accurate cost models**

The Rise of Learned Cost Models

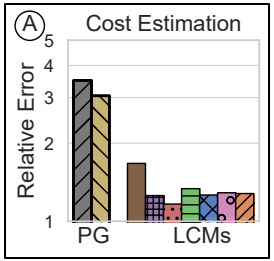
Query Optimization | **Resource Allocation** | **Index & MV Advisors** | **Query Scheduling**





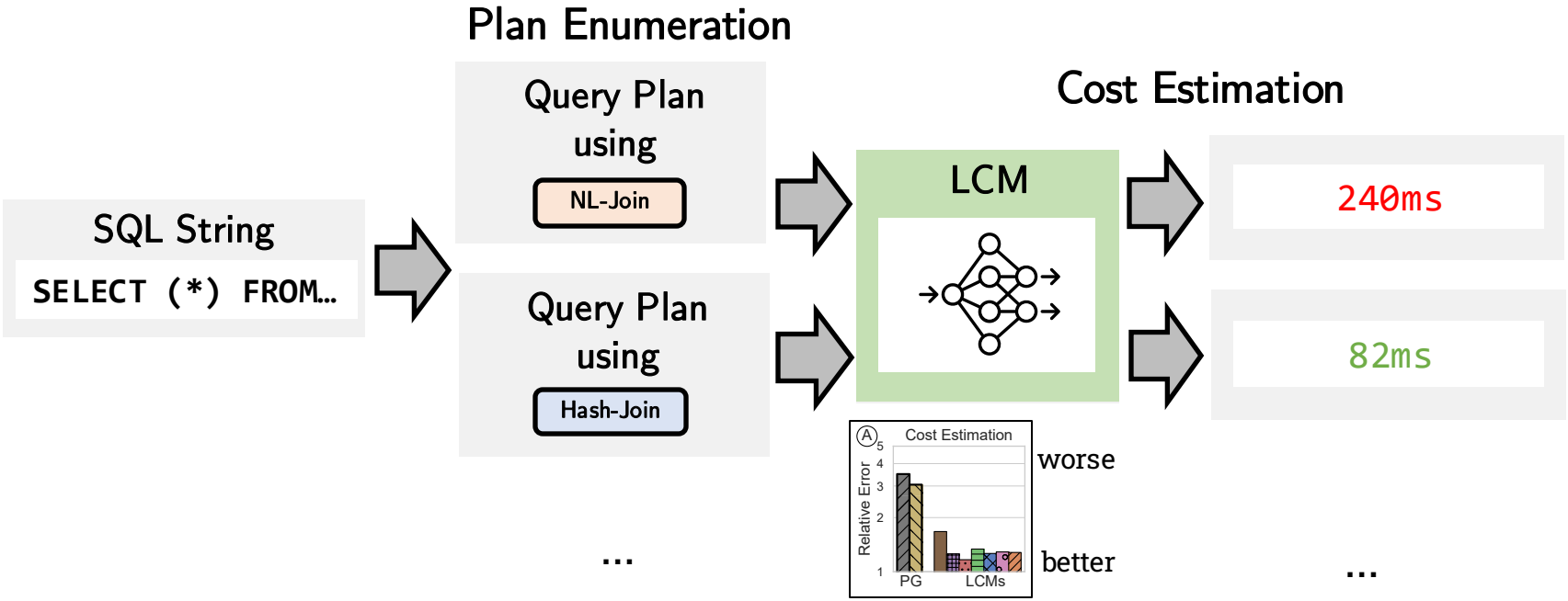
Hot research topic

Learn from previous query executions using ML



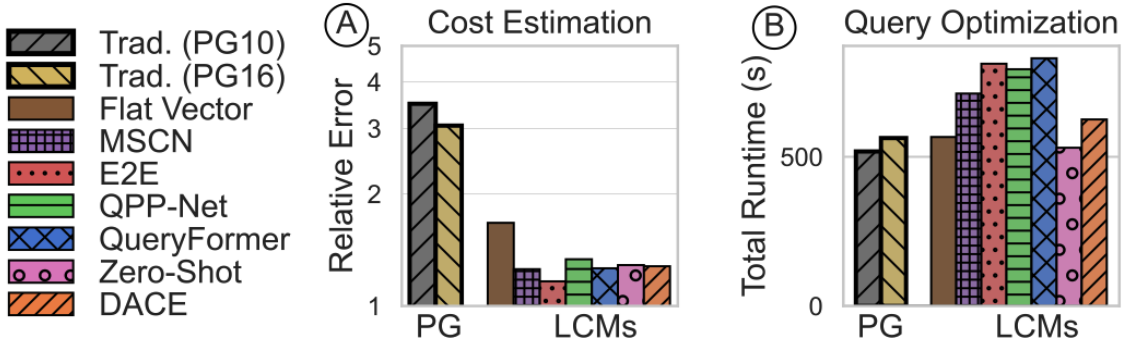
✅ **Highly precise cost estimates!**

Cost Models in Query Optimization



Precise Cost Estimates are crucial!
How do LCMs help in Query Optimization?

How Good Are LCMs for Query Optimization?



See Paper

Task 1: Join Ordering

Task 2: Access Path Selection

Task 3: Join Operator Selection

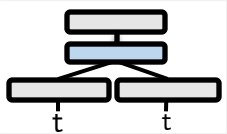
Setting the Stage

Comparing 7 State-Of-The-Art LCMs

Flat

```
["num_joins": 1, …]
```

Graph



Model		Plan Representation	Database-Dependency	Model Architecture
Flat Vector	[10]	Flat	DB-agnostic	Regression Tree
MSCN	[18]	Flat	DB-specific	Deep Sets
End-To-End	[31]	Graph	DB-specific	Tree Structured NN
QPP-Net	[27]	Graph	DB-specific	Neural Units
QueryFormer	[44]	Graph	DB-specific	Transformer
Zero-Shot	[13]	Graph	DB-agnostic	Graph Neural Networks
DACE	[23]	Graph	DB-agnostic	Transformer

DB-specific

DB-agnostic

Baselines

- Postgres 10
- Postgres 16

Training Data

- 200.000 SPAJ Queries
- 20 different databases
- All models trained on same data per dependency class

Model Inputs

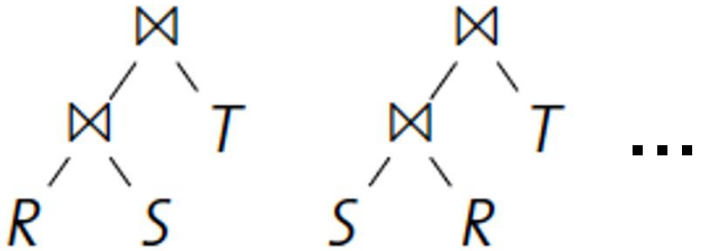
- Query Plans
- Estimated Cardinalities
- Table Samples
- Database Statistics
- PostgreSQL Costs

Task 1: Join Ordering

Which order of joins is optimal for a given query?

Requirement for Cost Models:
Identify Fastest Join Order

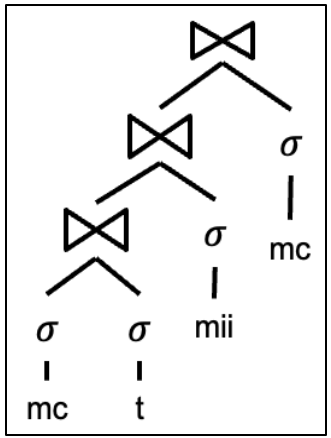
Example: Join tables R, S and T



2sec 28sec

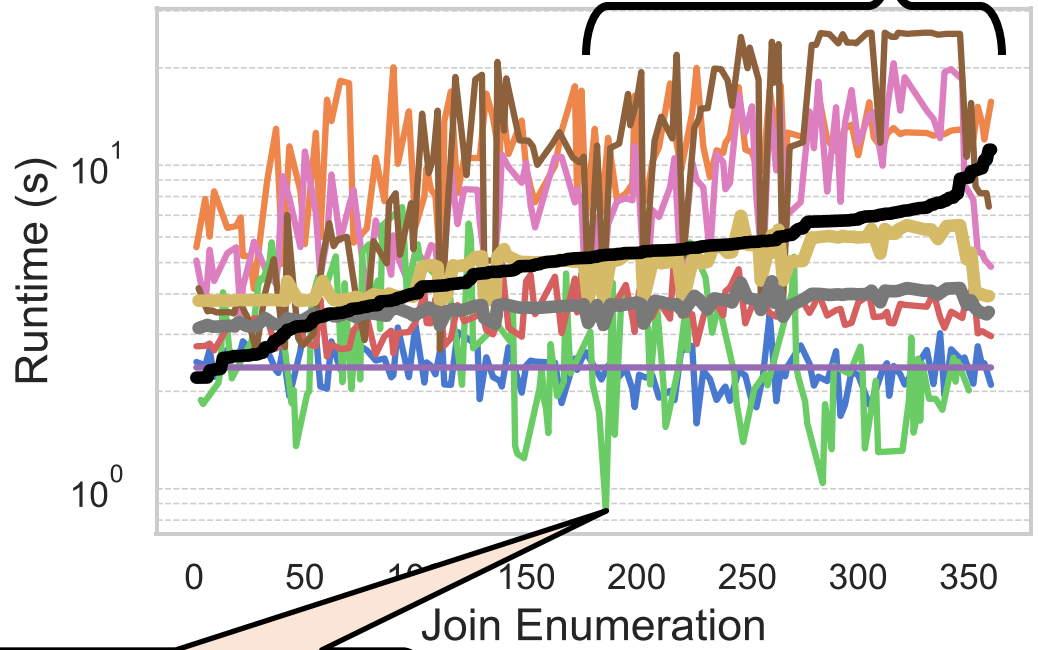
Join Ordering – Example Query

Query Nr. 33 from JOB-Light



- Real Runtime
- Sc. PG10
- Sc. PG16
- Flat Vector
- MSCN
- E2E
- QPP-Net
- QueryFormer
- Zero-Shot
- DACE

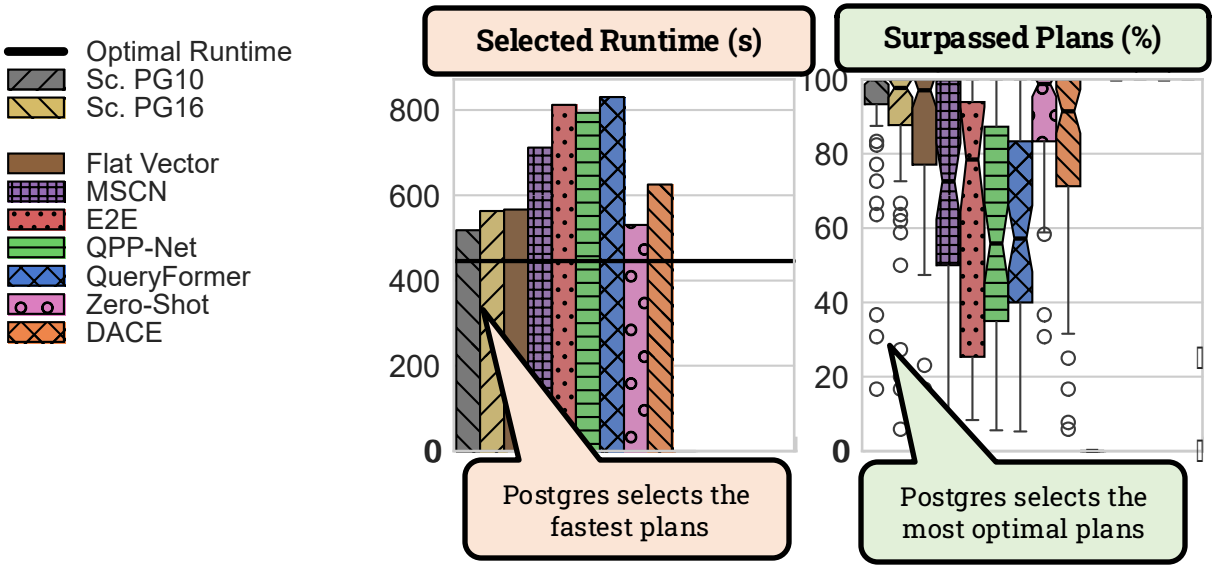
Surpassed Plans (%)
How optimal is the selected plan?



Selected Runtime (s)
How fast is the selected plan?

Join Ordering – Full Evaluation

Benchmark: JOB-Light
Exhaustive Join Enumeration
70 Queries with up to 4 joins → ~23.000 plans



Traditional models are still outperforming LCMs for join ordering!

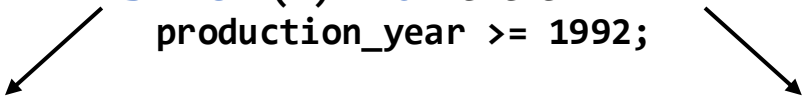
Task 2 – Access Path Selection

How to optimally access a given table?

Requirement for Cost Models:
Decide between Sequential Scan and Index Scan

Example: Find Optimal Access Path

```
SELECT (*) FROM title WHERE
production_year >= 1992;
```

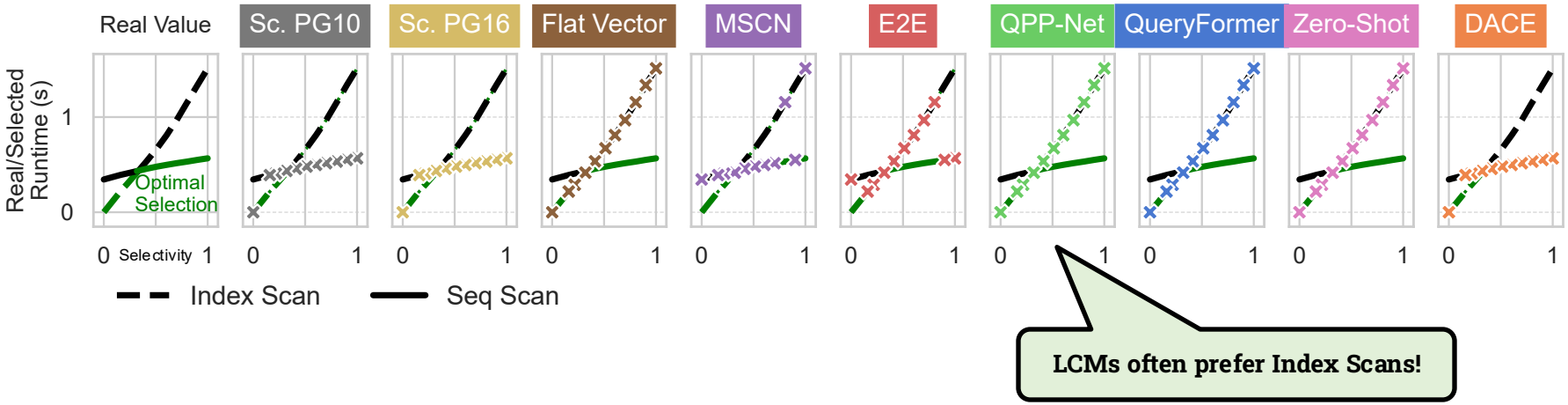


Sequential Scan
fast for many qualifying tuples

Index Scan
fast for few qualifying tuples

Access Path Selection – Over Single Column

```
SELECT (*) FROM title WHERE production_year >= ??
```



Postgres outperforms LCMs for access path selection

How Good Are LCMs for Query Optimization?

Task 1:
Join Ordering

Task 2:
Access Path Selection

Task 3:
Join Operator Selection

See Paper

Classical Approaches typically still outperform LCMs in all tasks!
In some cases, they are on par.

One Reason: The Training Data Bias
LCMs prefer Indexes - because they are learned to be always fast.
LCMs learn from pre-optimized plans only

Is this the end?

Why use LCMs at all?

Use LCMs at all!

- Traditional Models are still off and come with limitations
- LCMs are able to provide highly accurate estimates
- LCMs are not yet optimized for Query Optimization

But How Can We Fix LCMs?

Look at the Downstream Task – which is Plan Selection!

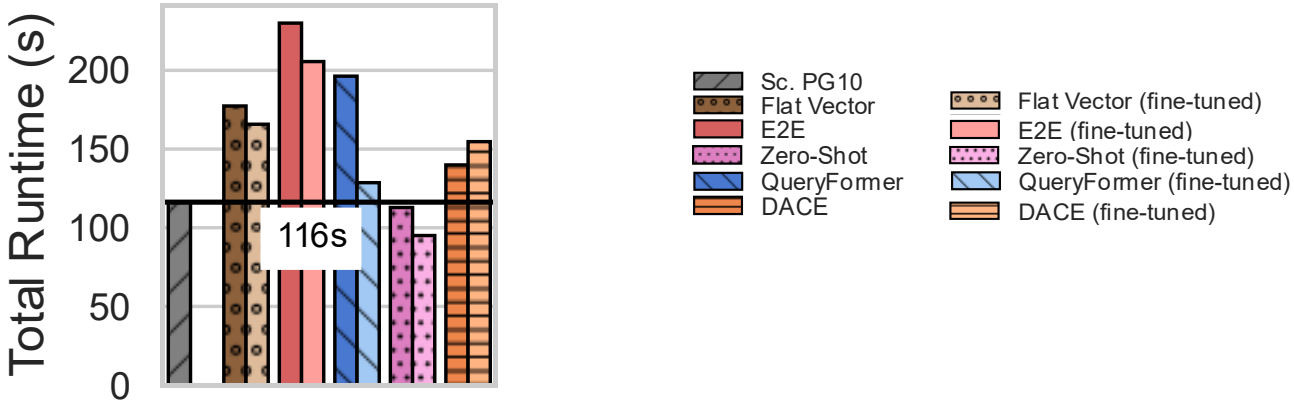
❌ LCMs focus only on accuracy

✅ LCMs need to address both ranking and accuracy

Overcome Training Data Bias

❌ Existing works learn from pre-optimized plans

✅ Learn also from sub-optimal plans



Making LCMs Practical For Query Optimization

Questions?

How Good Are Learned Cost Models, Really?
Insights From Query Optimization Tasks

roman.heinrich@dfki.de

Paper **Code** **Data**