JOB-Complex: A Challenging Benchmark for Traditional & Learned Query Optimization

Johannes Wehrstein [*], Timo Eckmann [*], Roman Heinrich [*‡], Carsten Binnig [*‡]

[*] Technical University of Darmstadt,
[‡] DFKI (German Research Center for AI)

SYSTEMS

Best Paper Award

TECHNISCHE
UNIVERSITÄT
DARMSTADT

How Good Are Query Optimizers?

Query Optimization is fundamental to DBMS performance

- Benchmarking QO is important: How good are query Optimizers?

JOB Benchmark:

☑ Many Joins (5-14)

☑ Complex Predicates (IN, LIKE, …)

☑ Real data distributions (IMDB)

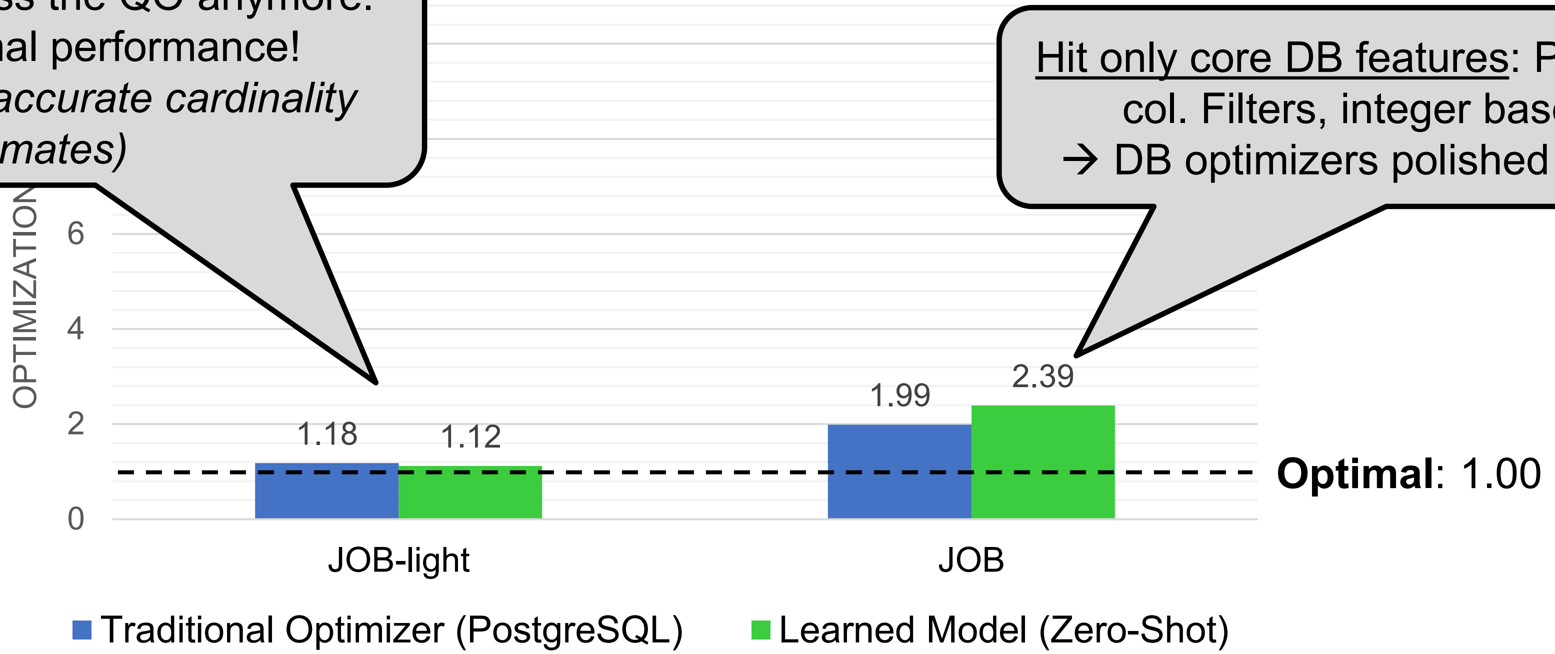
☑ Complex correlation between columns & tables

Phenomenal!

BUT: hit only core DB features!

How good are Query Optimizers Really?

They do not stress the QO anymore. Almost optimal performance! *(Considering inaccurate cardinality estimates)*



<u>Hit only core DB features</u>: PK/FK joins, single col. Filters, integer based join-conds
→ DB optimizers polished for these queries

Need benchmarks that stress the optimizer: Complex queries in the “wild”

Introducing JOB-Complex

JOB-Complex: Novel Benchmark for QO & Cost-Estimation

- 👍 Builds on IMDB dataset (like JOB)
- 👍 30 queries, 5-14 joins
- 👍 Builds on JOB queries (i.e. preserves join-paths & correlation from JOB)
- 👍 Adds real-world complexity
  - Joins on Strings
  - Joins on non-PK/FK
  - Complex Filters (LIKE, IN)
  - Intra-table comparisons

46% of join-keys in BI queries are strings in Snowflake
(Szlang et al., VLDB'25 – Industry 2)

Keep queries similar to JOB
BUT: introduce realistic conditions

Example Query (No. 12)

```
SELECT MIN(chn.name), ...
FROM complete_cast cc, comp_cast_type cct1,
    comp_cast_type cct2, ... -- (11 other tables)

WHERE cct1.kind = 'cast’
AND cct2.kind LIKE '%complete%'
AND chn.name IS NOT NULL
AND (chn.name LIKE '%man%’
     OR chn.name LIKE '%Man%’
AND k.keyword IN (...)
AND ... -- (other filters)

AND chn.id = ci.person_role_id
AND ak.name_pcode_cf=n.name_pcode_cf -- on strings
AND ak.name_pcode_nf=chn.name_pcode_nf -- on strings
AND ...
```

Up to 15 tables involved in the queries

Complex predicates on String & Numeric columns

Join Conditions on Strings & Non-PK/FK columns

How well do QO perform on JOB-Complex?

Evaluation Setup:

Query: SELECT * FROM … WHERE …

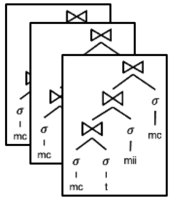
For fair comparison of approaches:
do candidate enumeration up-front (offline)

1. Query Plan Enumeration (Offline)

Randomized Plan Generation

+

Oracle-guided (cardinalities)



Diverse set of query plans

Runtime of **Best Plan**

3.5 s

Apply Cost-Models on the Query Plan

2. Costing

Apply Traditional & Learned Cost Models

↓

Select Plan with lowest cost

Runtime of **Selected Plan**

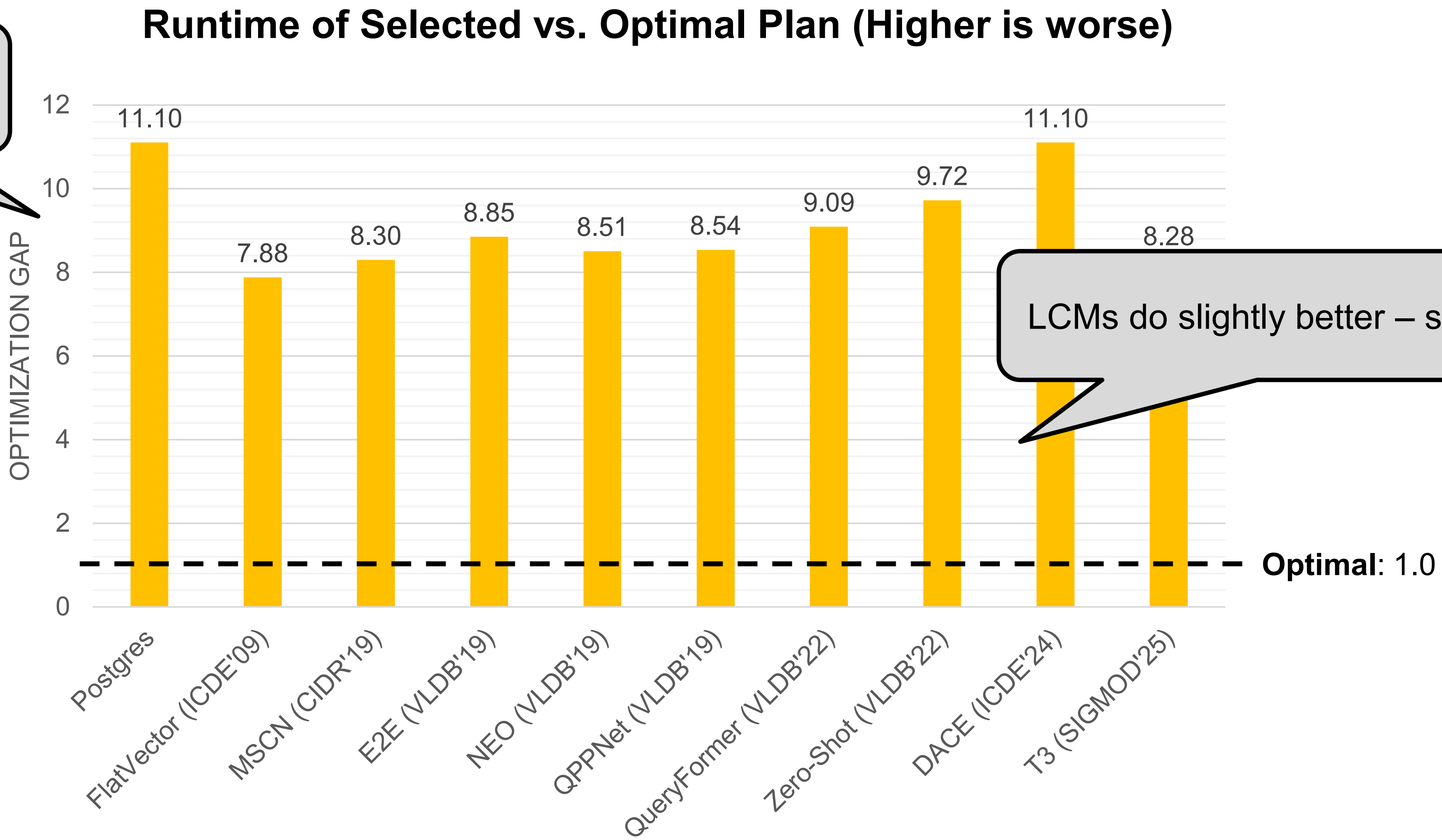
24.3 s

Compare with runtime of the best plan

*Focus on cost-models since most Query Optimizers are cost-based

JOB-Complex

11x Optimization Gap for PostgreSQL



LCMs do slightly better – still 8x gap

LCM’s help – BUT: we are far from solving the problem

Why Optimizers Struggle

Cost Estimation Accuracy:

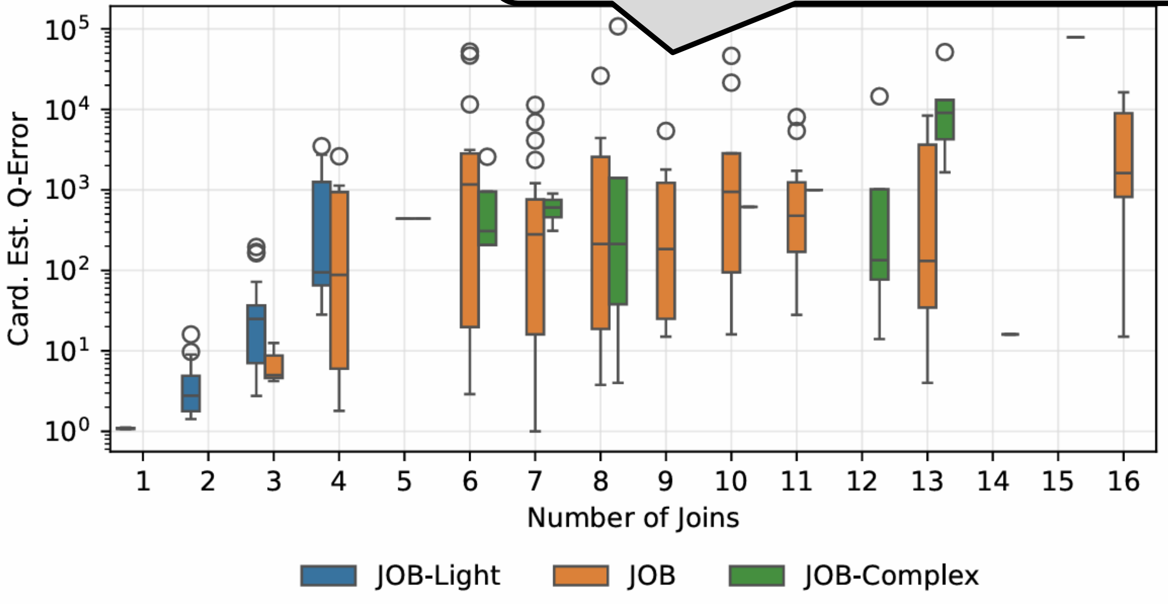
PG Q-Error increases by 2x

Median Q-Error	JOB	JOB-Complex
Postgres (v16)	1015.67	2669.22
DACE	1.91	1.81
T3	4.07	2.30
Zero-Shot	1.58	1.60
Median of all 10 Cost Models	**10.75**	**6.99**

Learned Models are more robust

Cardinality Accuracy:

Cardinalities for JOB-Complex are slightly worse than JOB – but not catastrophic



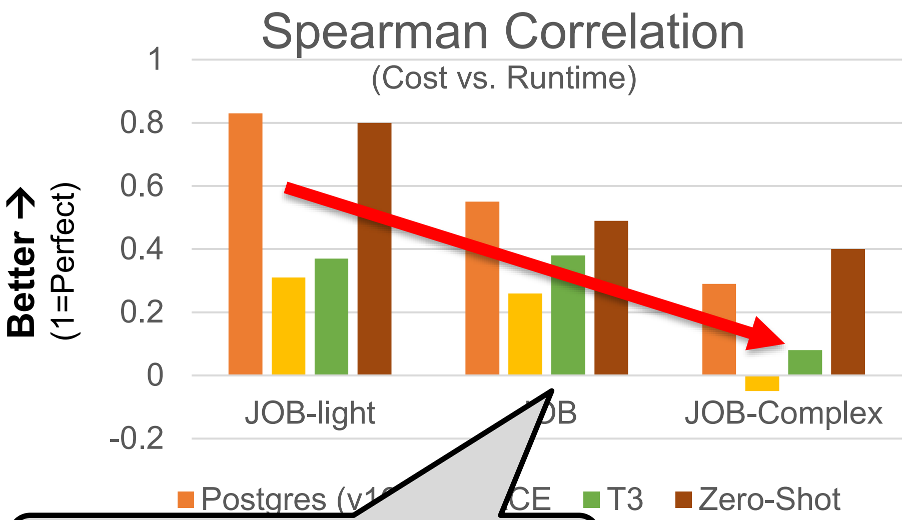
~~"it's all about cardinalities" – JOB CIDR'17~~

It's more than that!

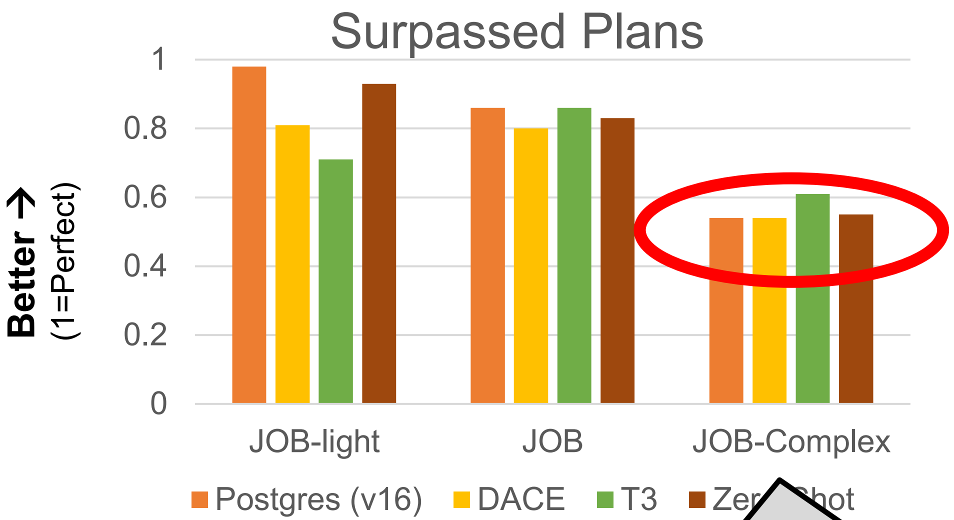
The missing piece: The Rank

Cost models have to be good sorting functions

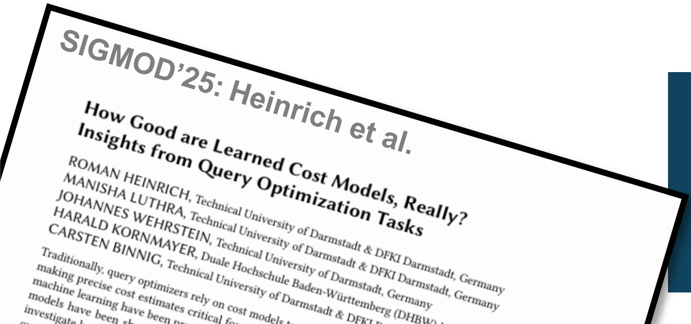
Percentile of the plan picked
*(0.5 = 50% of plans are worse,
50% are better*



Correlation drastically decreases



Not much better than random plan selection (~0.5)!



SIGMOD'25: Heinrich et al.

**How Good are Learned Cost Models, Really?
Insights from Query Optimization Tasks**

ROMAN HEINRICH, Technical University of Darmstadt & DFKI Darmstadt, Germany
MANISHA LUTHRA, Technical University of Darmstadt & DFKI Darmstadt, Germany
JOHANNES WEHRSTEIN, Technical University of Darmstadt, Germany
HARALD KORNMAYER, Duale Hochschule Baden-Württemberg (DHBW)
CARSTEN BINNIG, Technical University of Darmstadt & DFKI

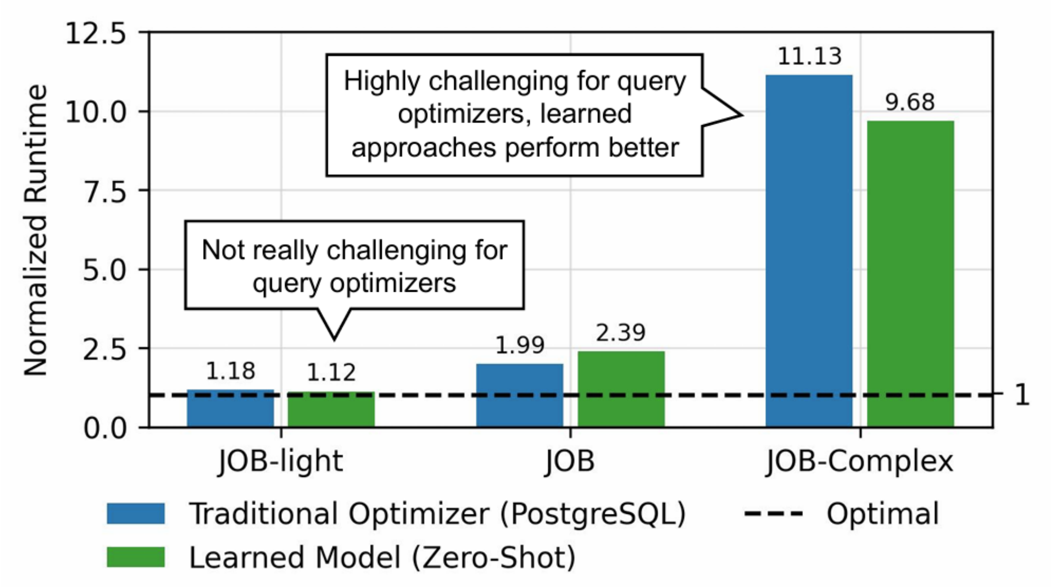
Traditionally, query optimizers rely on cost models
making precise cost estimates critical
machine learning have been
models have been
investigate

**It's about the sorting of the plans:
Monotonicity is key!**

Takeaways

Query Optimization is not solved – Especially under real-world complexity

- Existing Benchmarks underestimate the problem
- JOB-Complex can effectively benchmark the QO capabilities



JOB-Complex

30 Challenging Queries on the IMDB dataset

Try it out!
→ Directly available on Github
github.com/DataManagementLab/JOB-Complex

Paper

Github