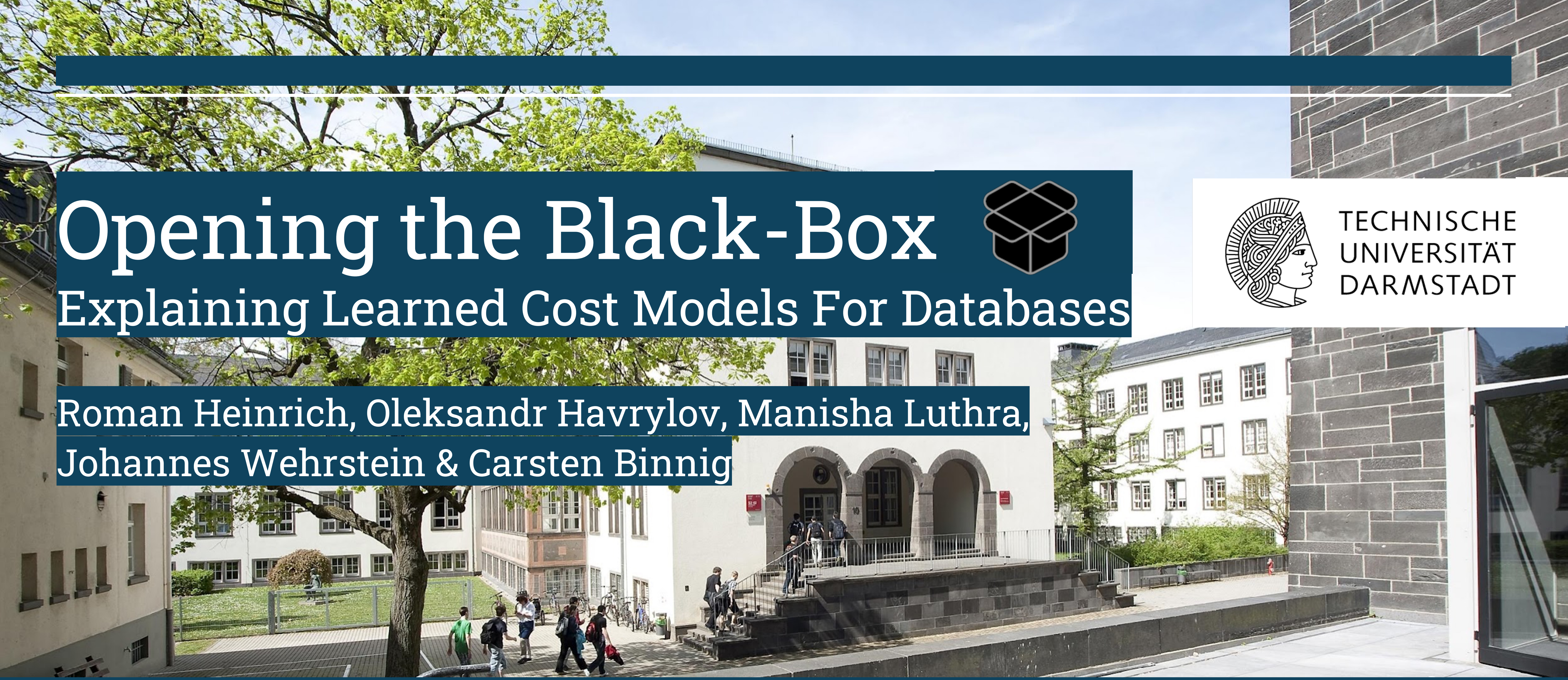


Opening the Black-Box

Explaining Learned Cost Models For Databases

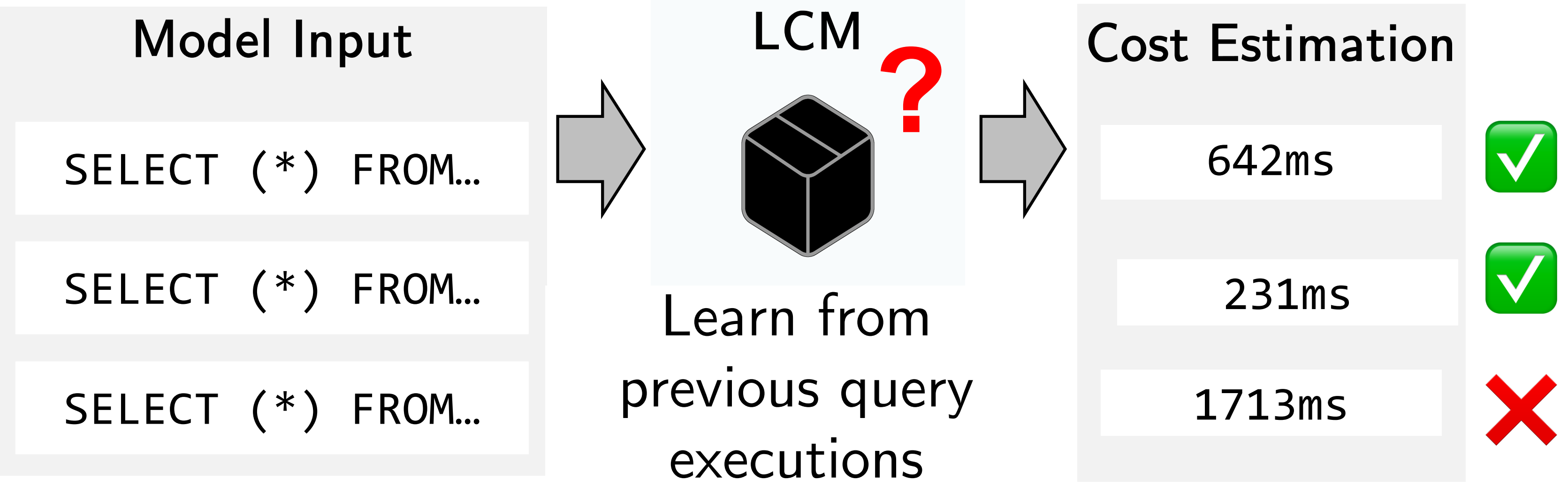
Roman Heinrich, Oleksandr Havrylov, Manisha Luthra,
Johannes Wehrstein & Carsten Binnig

TECHNISCHE UNIVERSITÄT DARMSTADT

What has a Learned Cost Model (LCM) really learned?

Motivation: LCMs Behave as a Black-Box

- **Learned Cost Models (LCMs)** predict query execution costs.
- LCMs show highly **promising results**, often outperforming classical models.
- However, LCMs are trained with deep learning and thus behave as **black box** in their predictions
- The black-box nature of LCMs makes it difficult to understand **why** and **how** they came up with given results



Model Input

```
SELECT (*) FROM…
SELECT (*) FROM…
SELECT (*) FROM…
```

LCM ?

Learn from previous query executions

Cost Estimation

642ms ✓

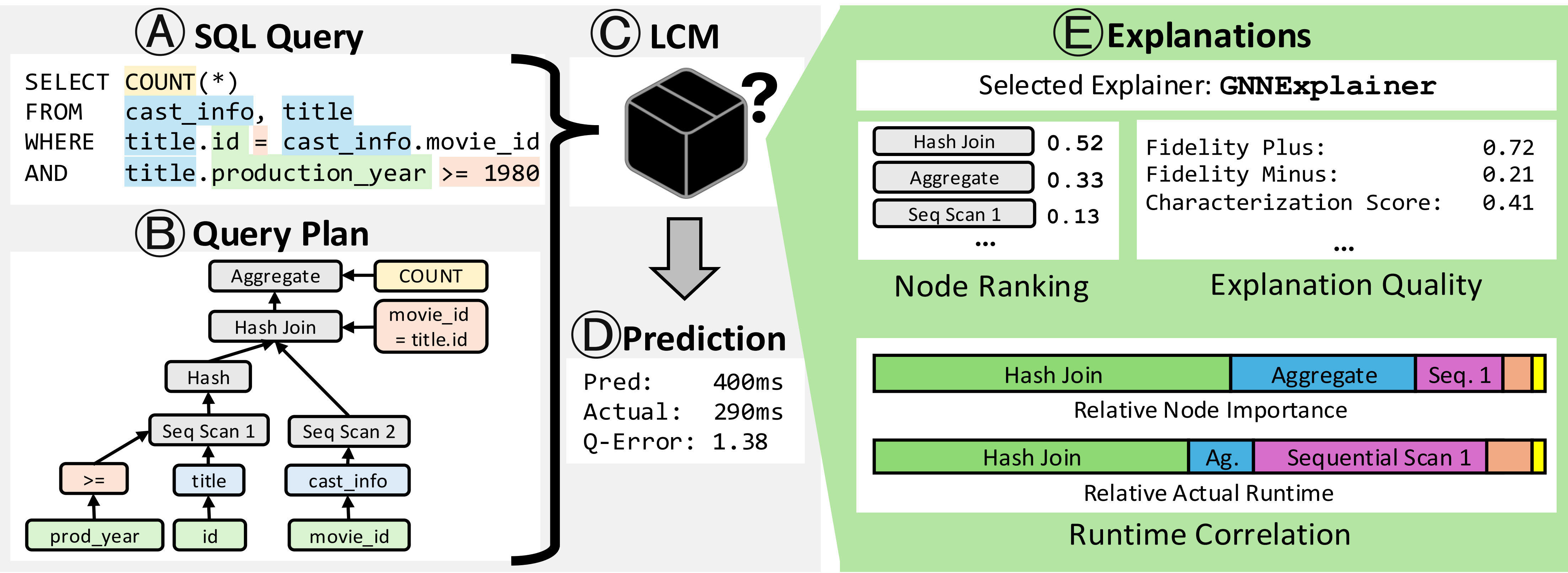
231ms ✓

1713ms ✗

It is unclear why LCMs mispredict query execution costs – because they act as a black-box during prediction

Contribution: Making LCMs Explainable

- Highlighting what the LCM has learned by showing the **importance** of the **nodes** (=query operators) during GNN inference.
- Analyzing how well existing explainers (e.g. GNNExplainer) can help.
- Using **gradient-based explainers** and **feature-based explainers** by adapting node masking for Zero-Shot cost models.



Ⓐ SQL Query

```
SELECT  COUNT(*)
FROM    cast_info, title
WHERE   title.id = cast_info.movie_id
AND     title.production_year >= 1980
```

Ⓑ Query Plan

Ⓒ LCM ?

Ⓓ Prediction

```
Pred:     400ms
Actual:   290ms
Q-Error:  1.38
```

Ⓔ Explanations

Selected Explainer: **GNNExplainer**

Hash Join	0.52
Aggregate	0.33
Seq Scan 1	0.13
…	

Node Ranking

```
Fidelity Plus:             0.72
Fidelity Minus:            0.21
Characterization Score:    0.41
…
```

Explanation Quality

Relative Node Importance

Relative Actual Runtime

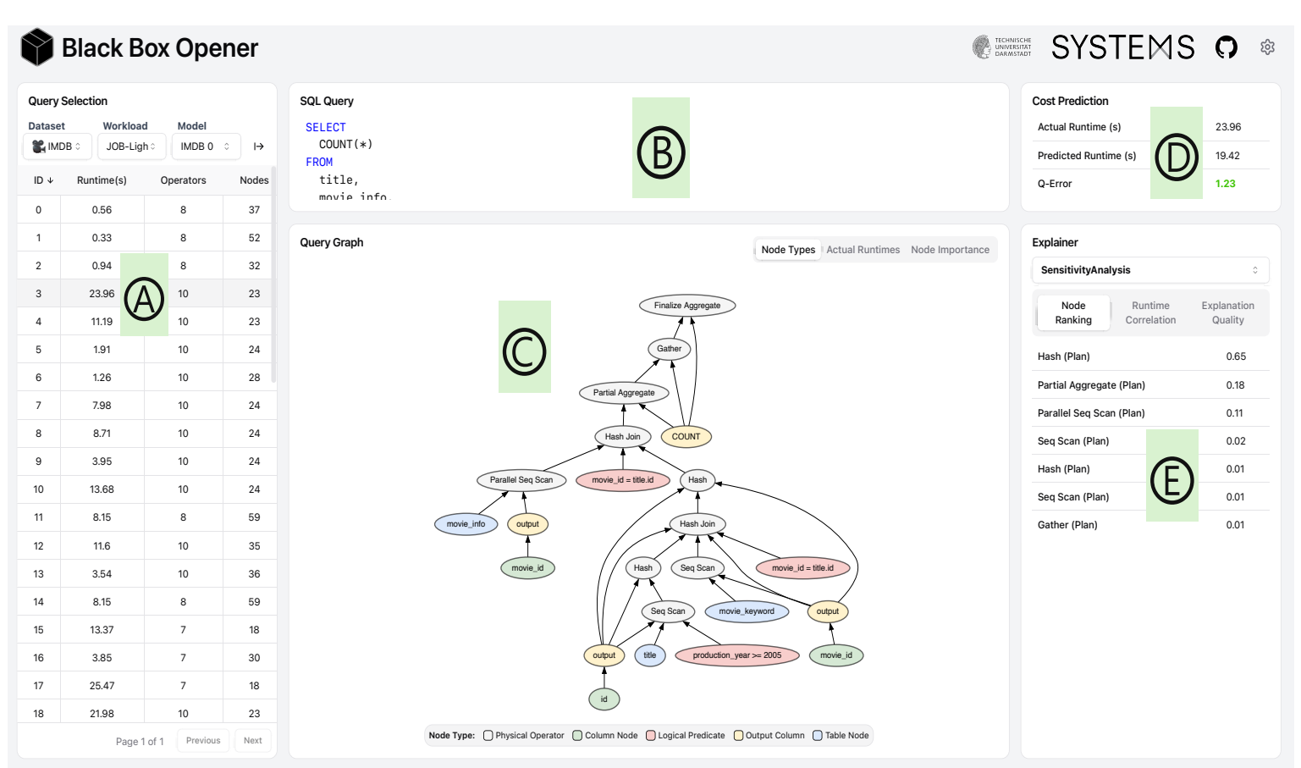
Runtime Correlation

Introducing **novel metrics**:

- *Node Ranking*: What node was most important?
- *Runtime Correlation:* How much do the importance scores correspond to the actual runtimes?
- *Explanation Quality*: How good is the explanation?

Our Idea: Show what LCMs have really learned by visualizing how important is a given node for the final prediction

Interactive Demo



Black Box Opener

Query Selection

Dataset: IMDB · Workload: JOB-Ligh · Model: IMDB 0

ID ↓	Runtime(s)	Operators	Nodes
0	0.56	8	37
1	0.33	8	52
2	0.94	8	32
3	23.96	10	23
4	11.19	10	23
5	1.91	10	24
6	1.26	10	28
7	7.98	10	24
8	8.71	10	24
9	3.95	10	24
10	13.68	10	24
11	8.15	8	59
12	11.6	10	35
13	3.54	10	36
14	8.15	8	59
15	13.37	7	18
16	3.85	7	30
17	25.47	7	18
18	21.98	10	23

Page 1 of 1 · Previous · Next

SQL Query

```
SELECT
  COUNT(*)
FROM
  title,
  movie_info,
```

Cost Prediction

Actual Runtime (s)	23.96
Predicted Runtime (s)	19.42
Q-Error	1.23

Query Graph

Explainer: SensitivityAnalysis

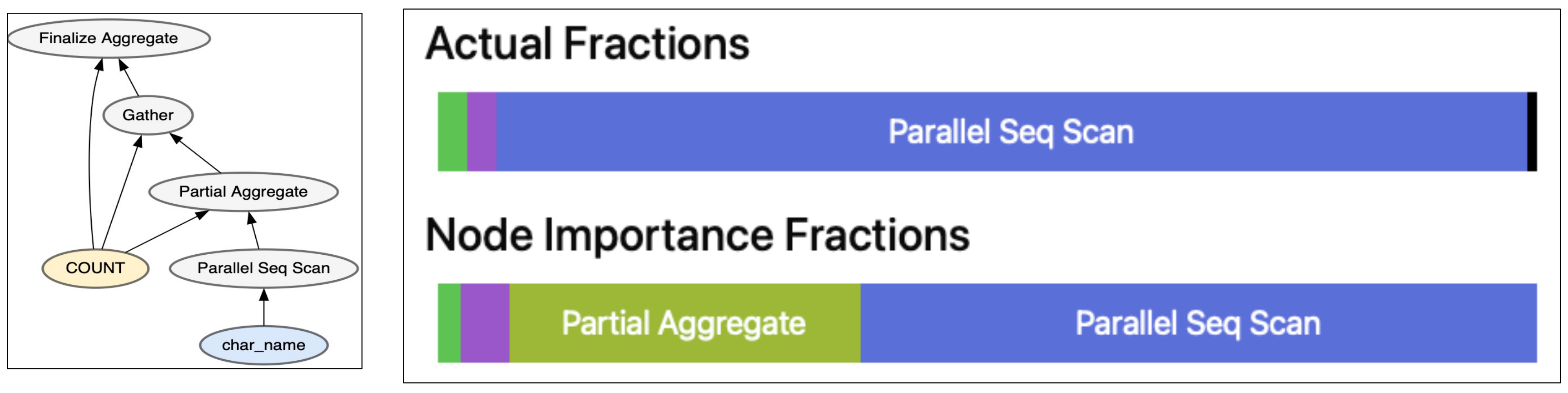
Node Ranking · Runtime Correlation · Explanation Quality

Hash (Plan)	0.65
Partial Aggregate (Plan)	0.18
Parallel Seq Scan (Plan)	0.11
Seq Scan (Plan)	0.02
Hash (Plan)	0.01
Seq Scan (Plan)	0.01
Gather (Plan)	0.01

Visualizing node importance scores of Learned Cost Models

Initial Results

- Often, the node importance scores match with the actual operator runtimes.



Actual Fractions

Node Importance Fractions

- Sometimes, LCMs tends to model puts too much **emphasis on aggregation** nodes!
- Still, LCMs often achieve **good predictions** despite **incorrect assumptions**

Outlook

- Exploring more explainer algorithms and LCM architectures
- Take feature importance and subgraph importance into account as well

Paper

Code