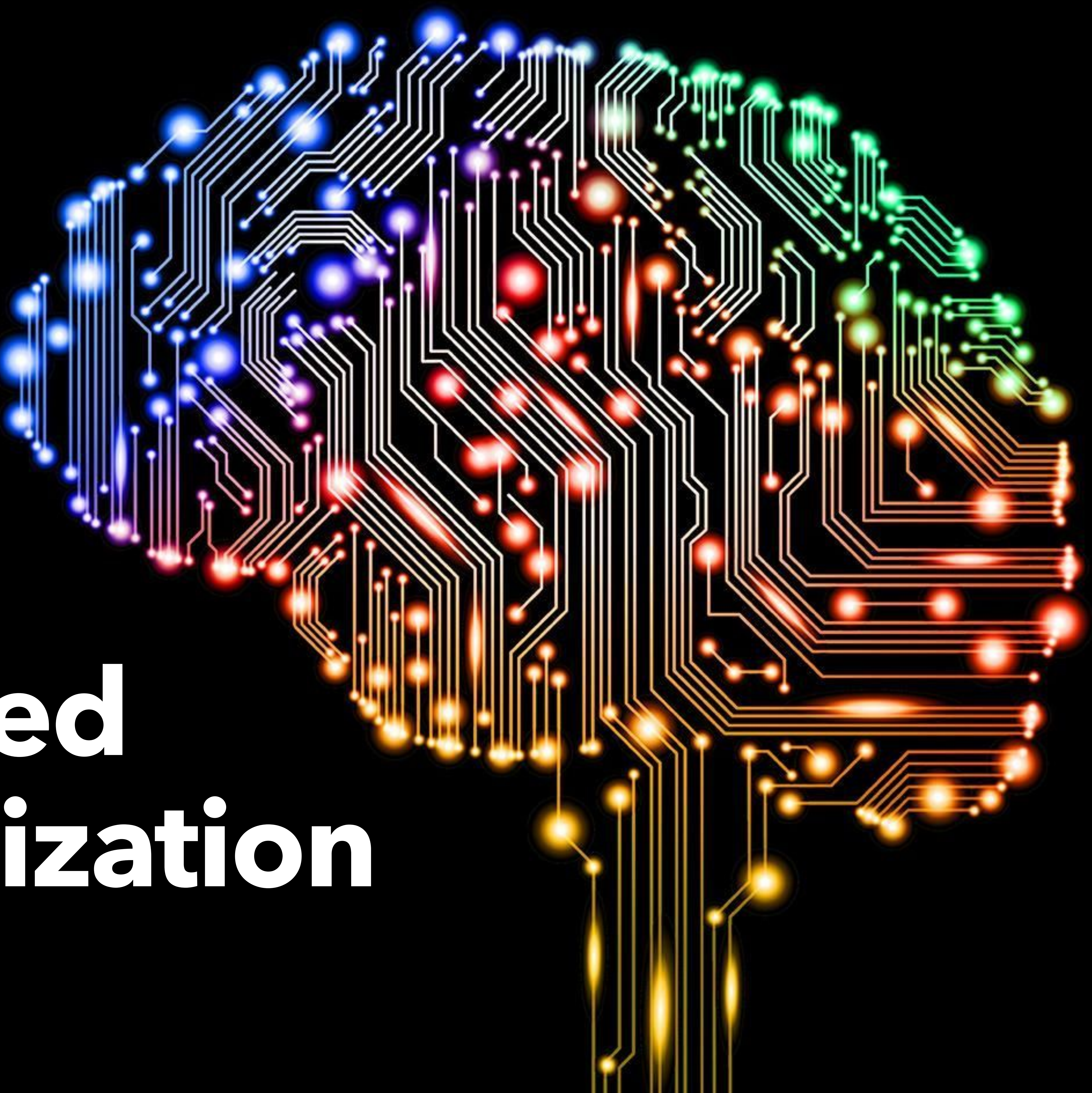
Zoi Kaoudi - 13.02.2023



Learning-based Query Optimization

What are we still missing?

IT UNIVERSITY OF COPENHAGEN

Why query optimization?

PostgreSQL

MySQL

ORACLE

IBM DB2

mongoDB

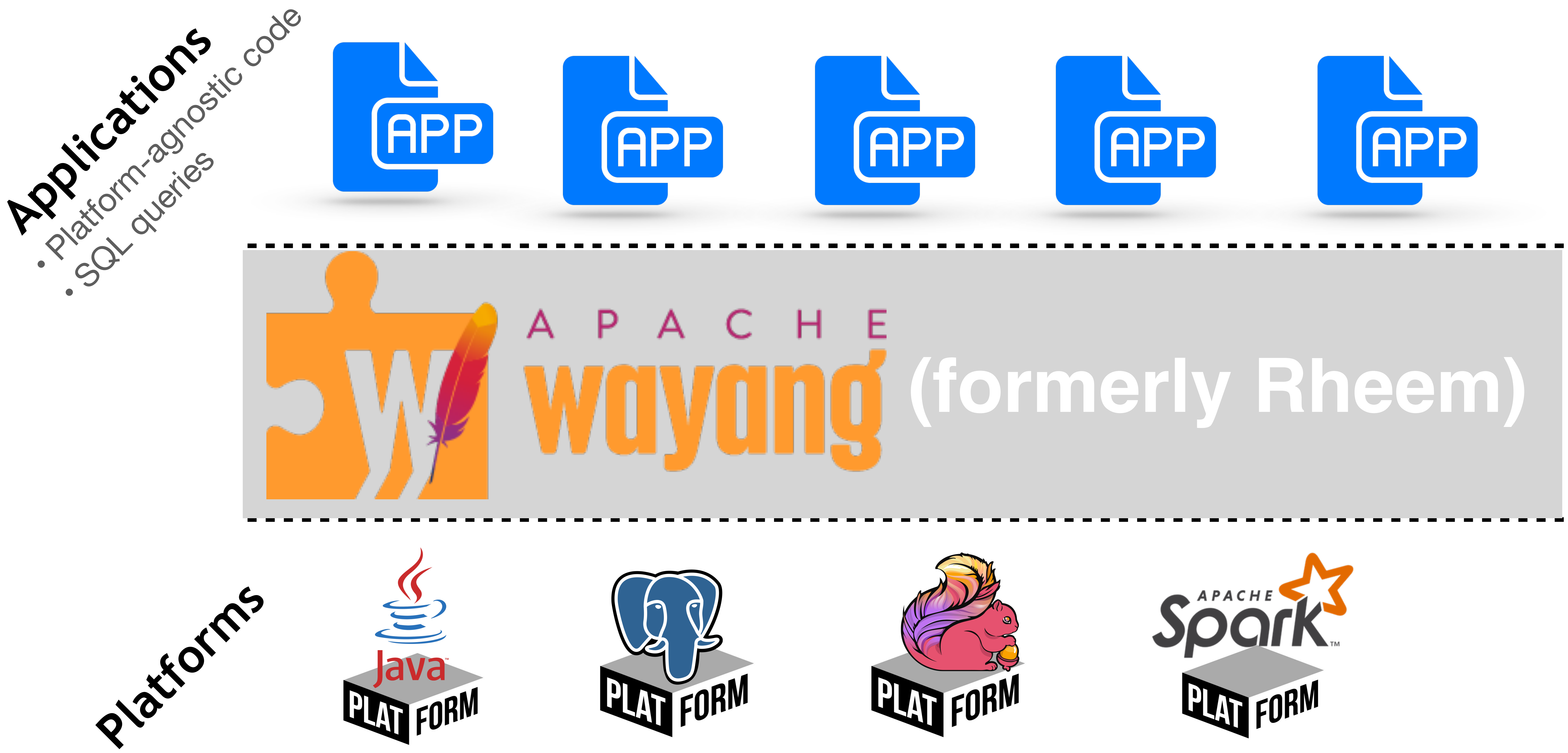
APACHE Spark

Flink

APACHE wayang

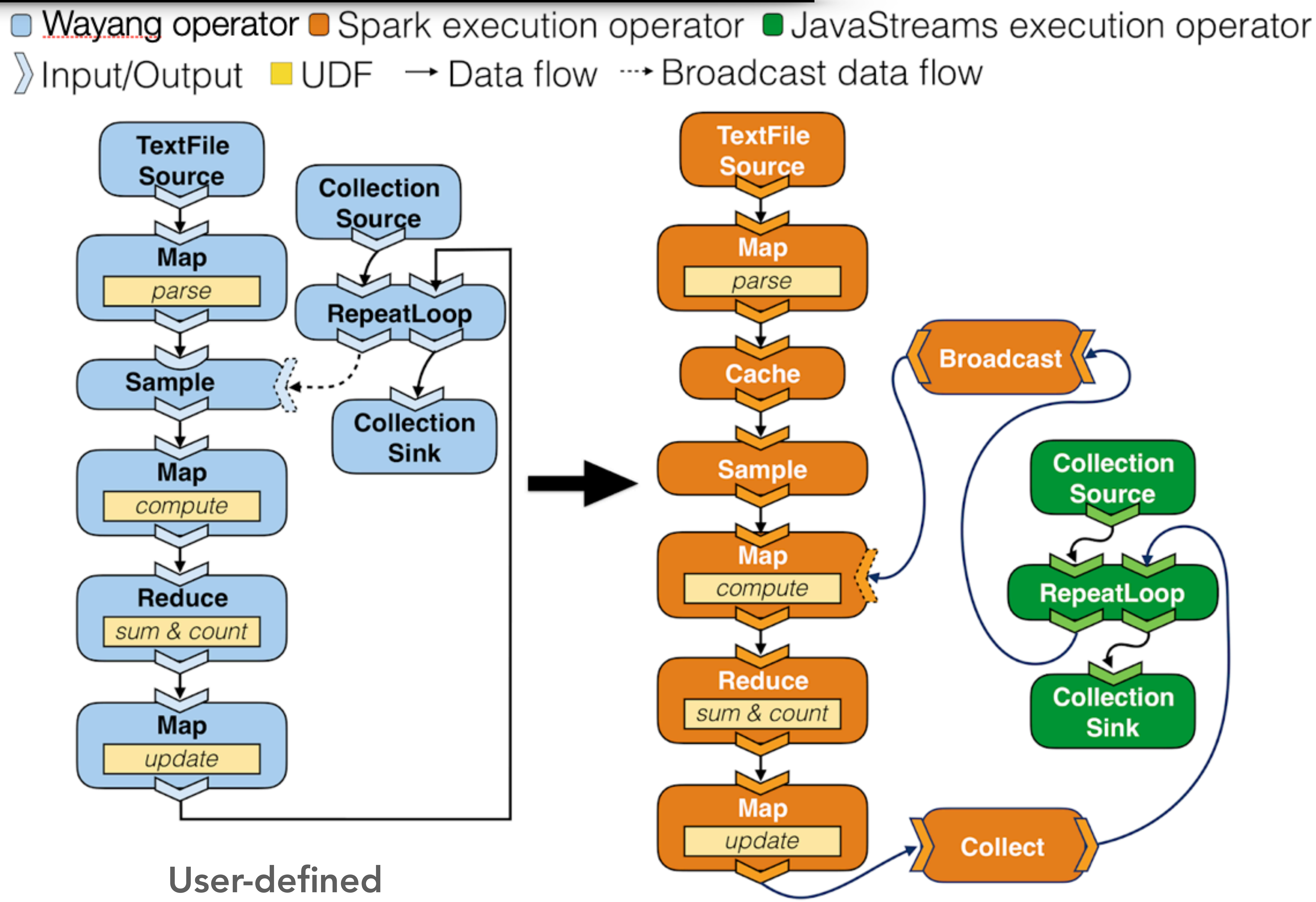
Apache Wayang

Decoupling applications from underlying processing platforms



Apache Wayang

Decoupling applications from underlying processing platforms



User-defined

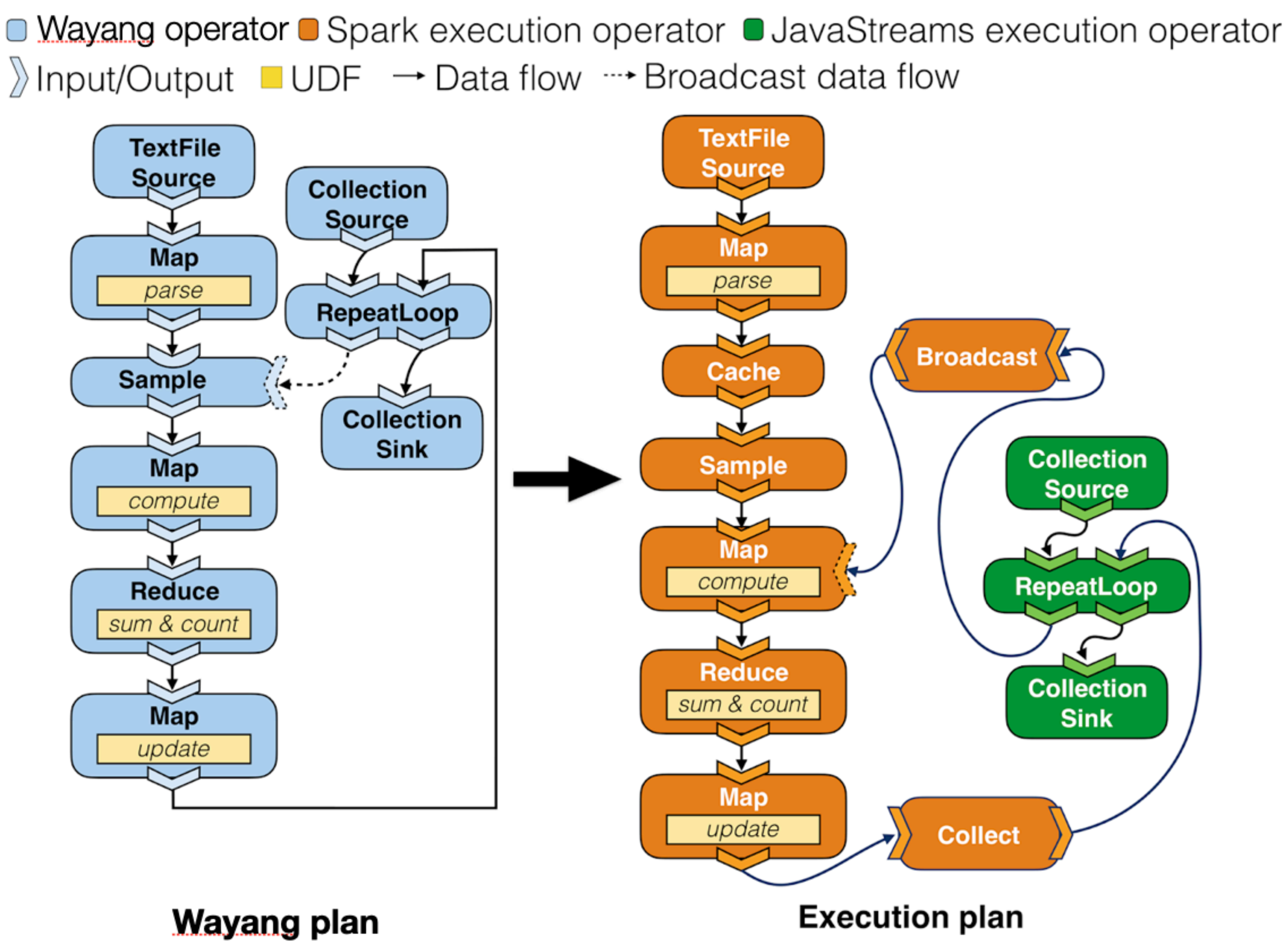
Logical plan

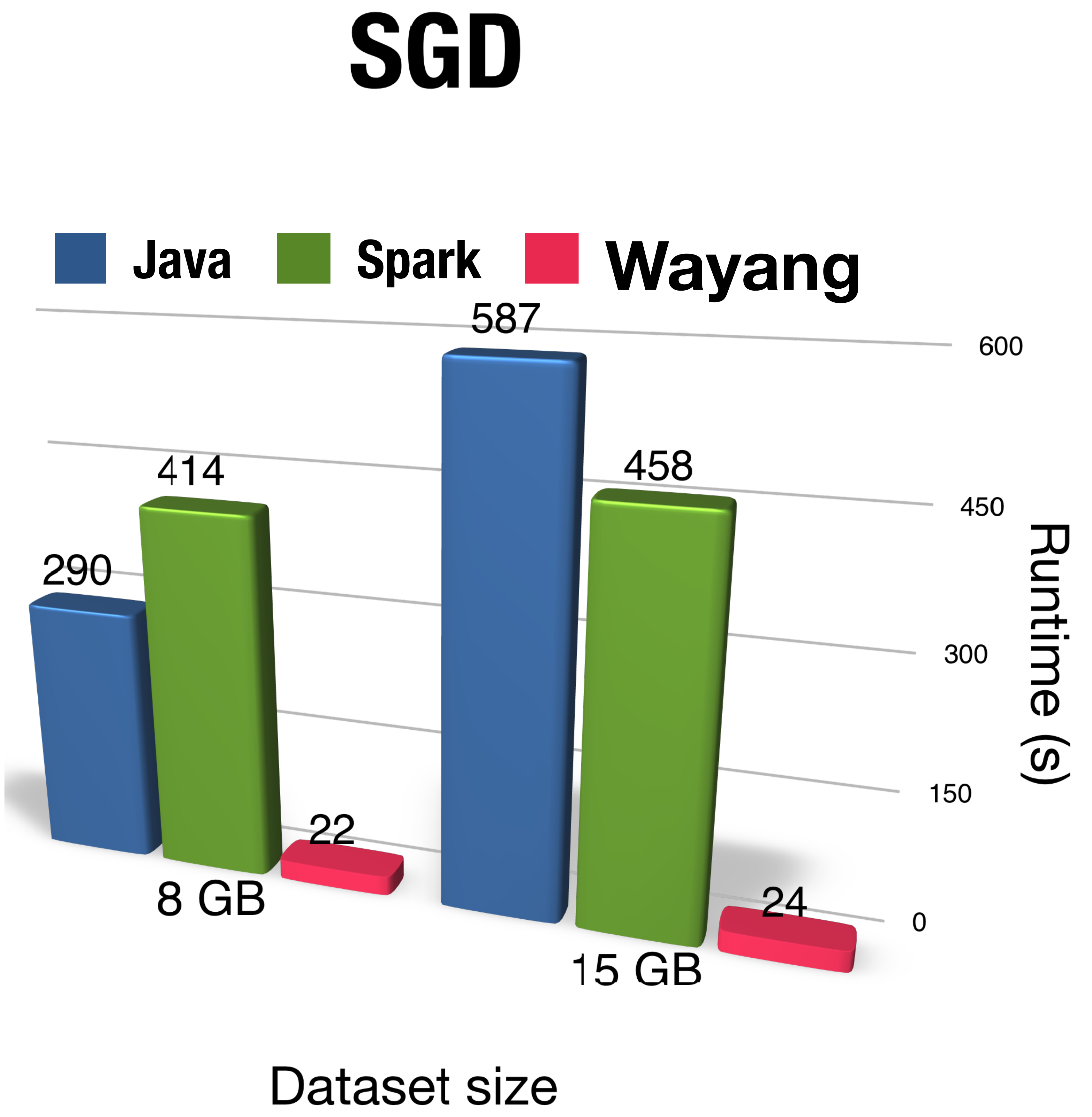
Optimised

Execution plan

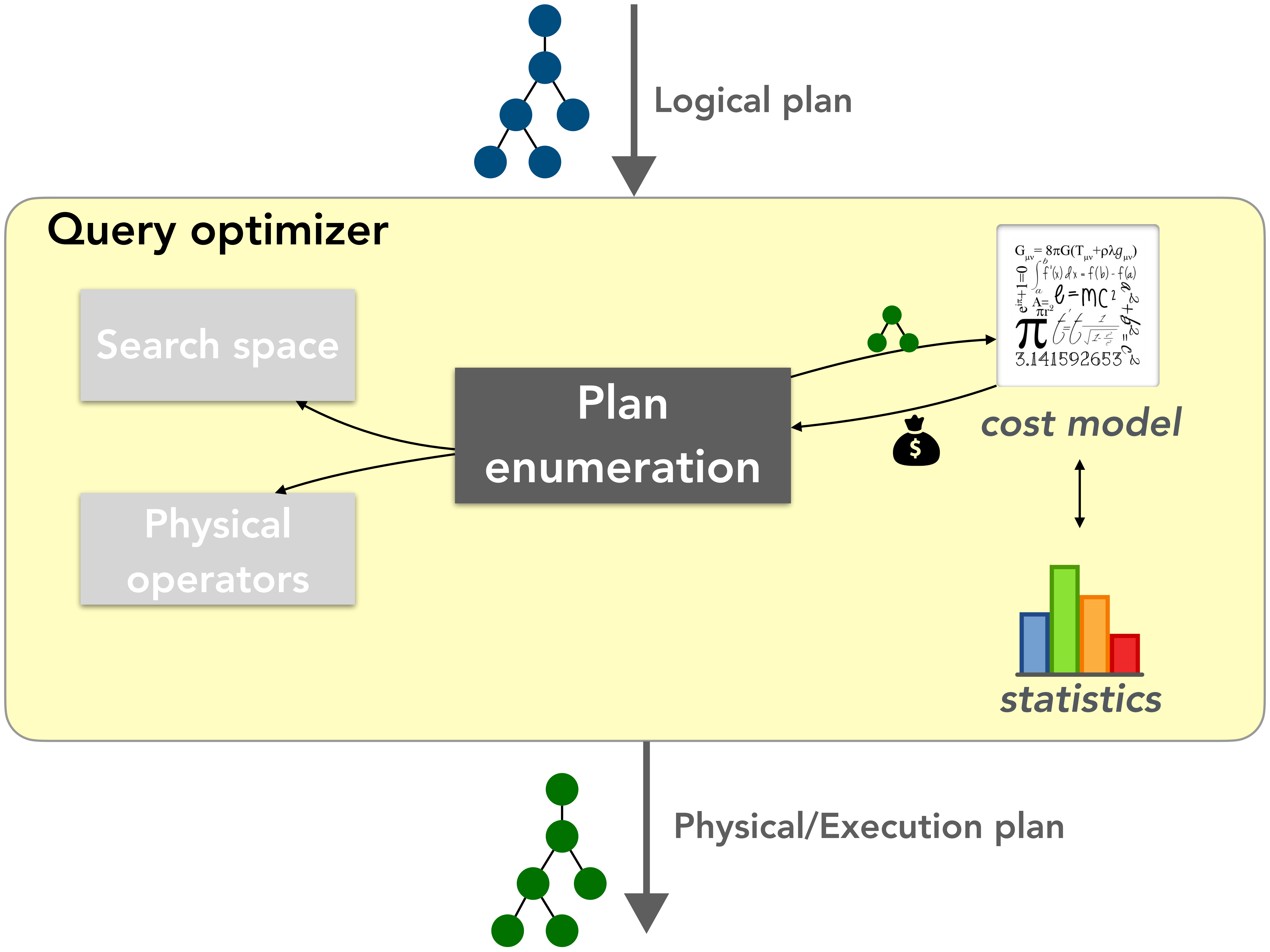
Apache Wayang

Decoupling applications from underlying processing platforms

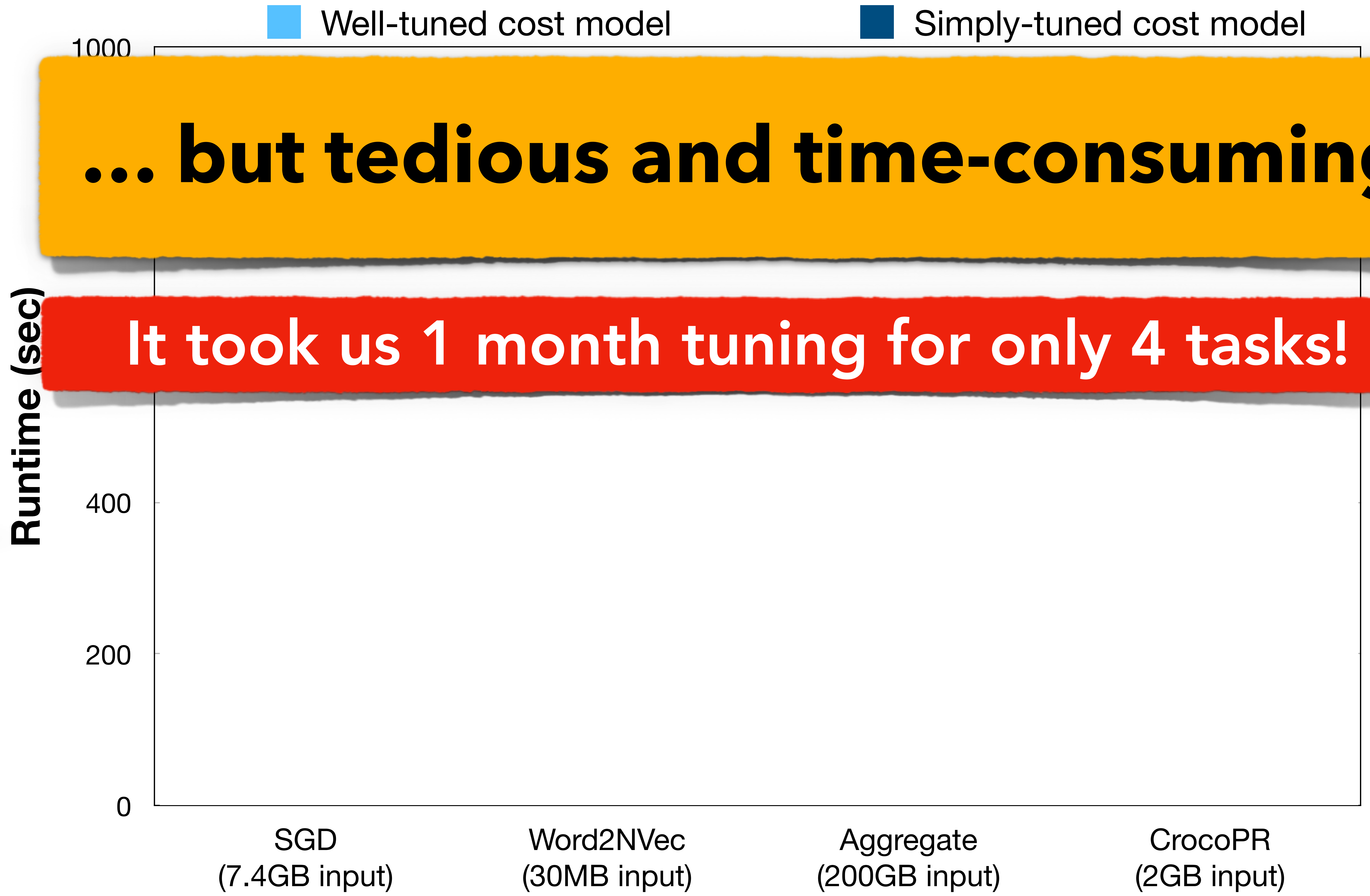




Cost-based query optimization process



Cost model tuning is important …

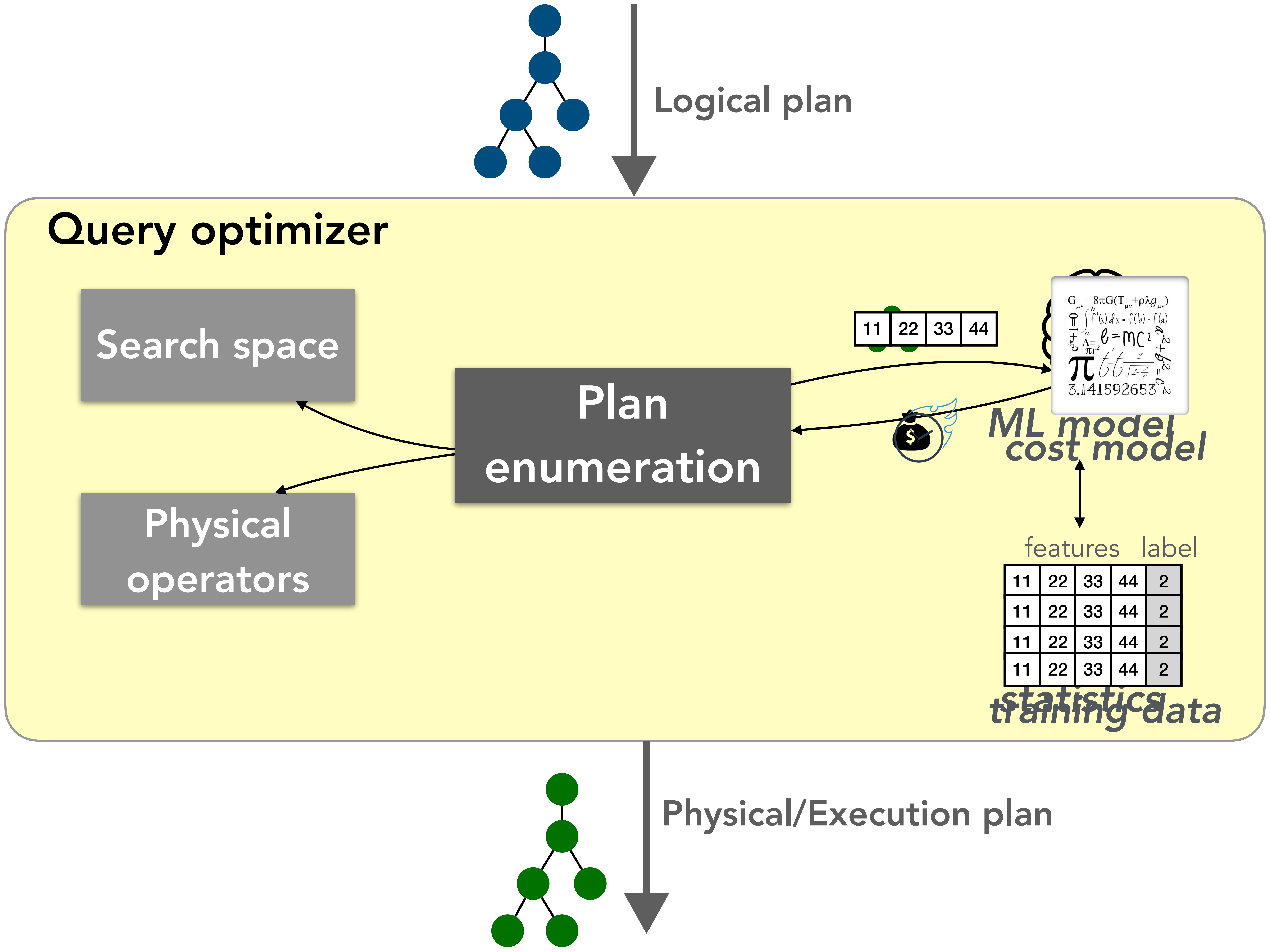


… but tedious and time-consuming!

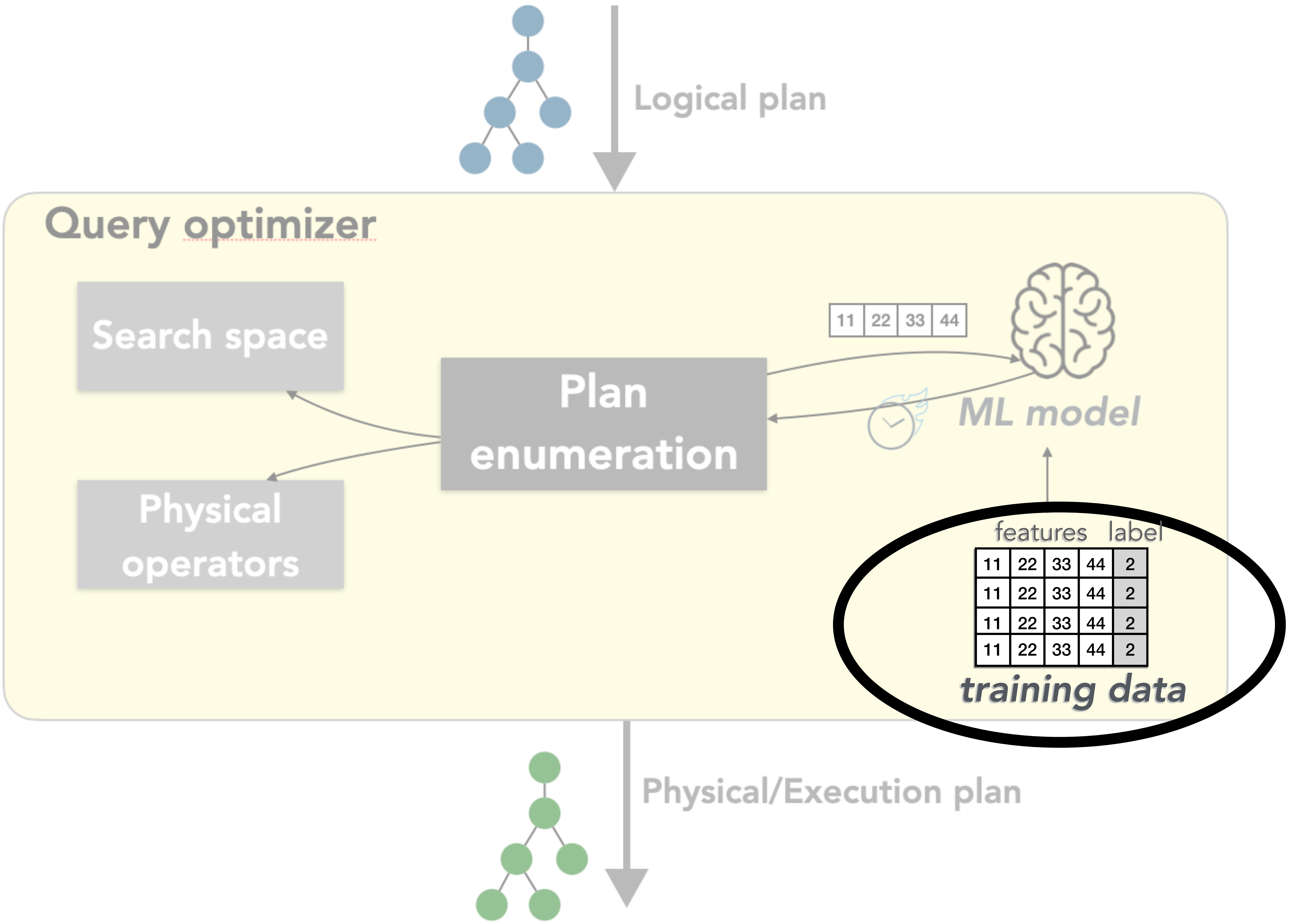
It took us 1 month tuning for only 4 tasks!

Using real cardinalities

ML to the rescue



Learning-based query optimization

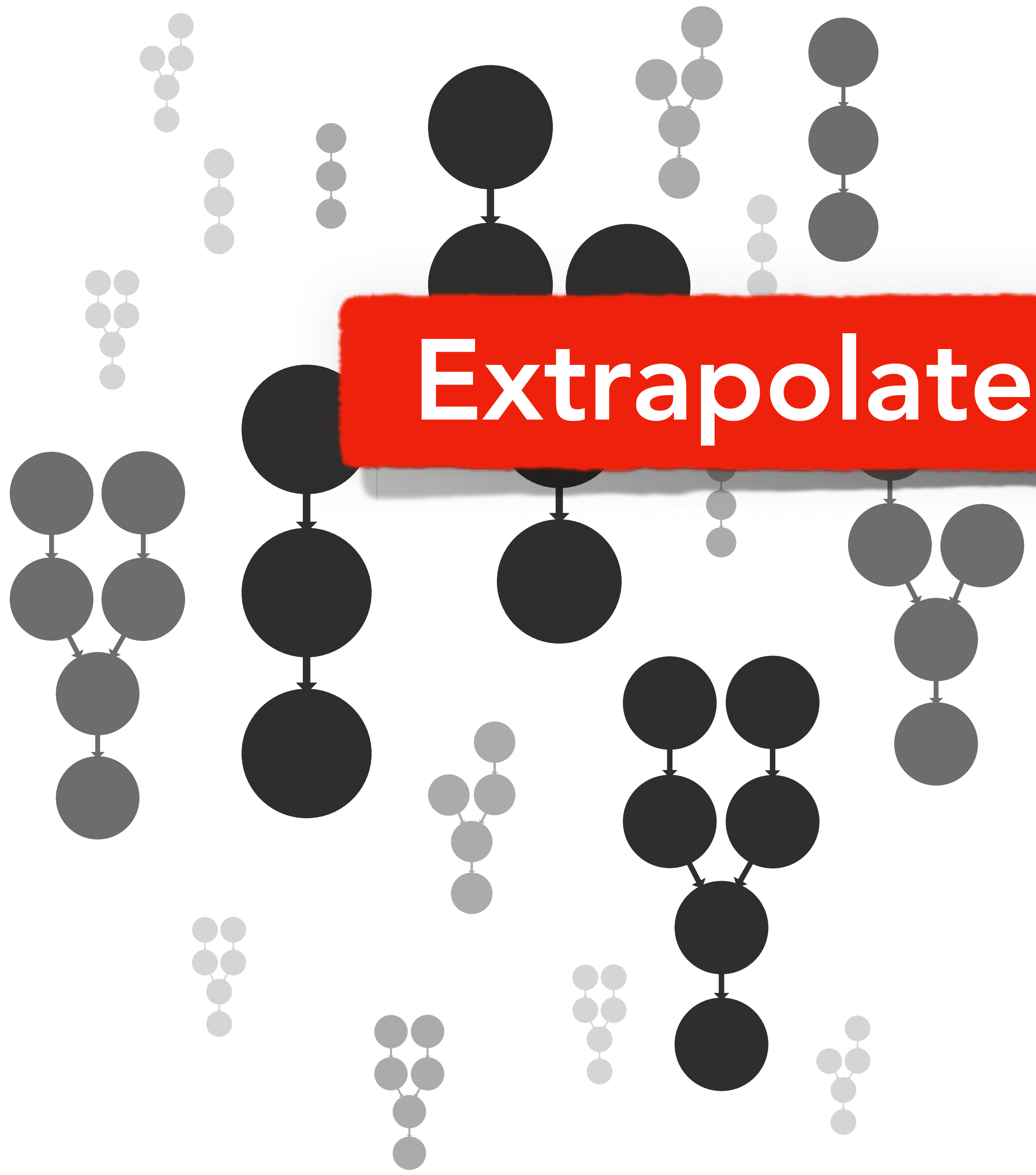


ML models are data-hungry

Naive solution to generate training data

Time-consuming

Extrapolated cost of 10,000 plans with 1TB input data > 6 months*



P1	…	…	…
P2	…	…	…
P3	…	…	…
…	…	…	…

P1	…	…	…	8s
P2	…	…	…	3s
P3	…	…	…	36s
…	…	…	…	…

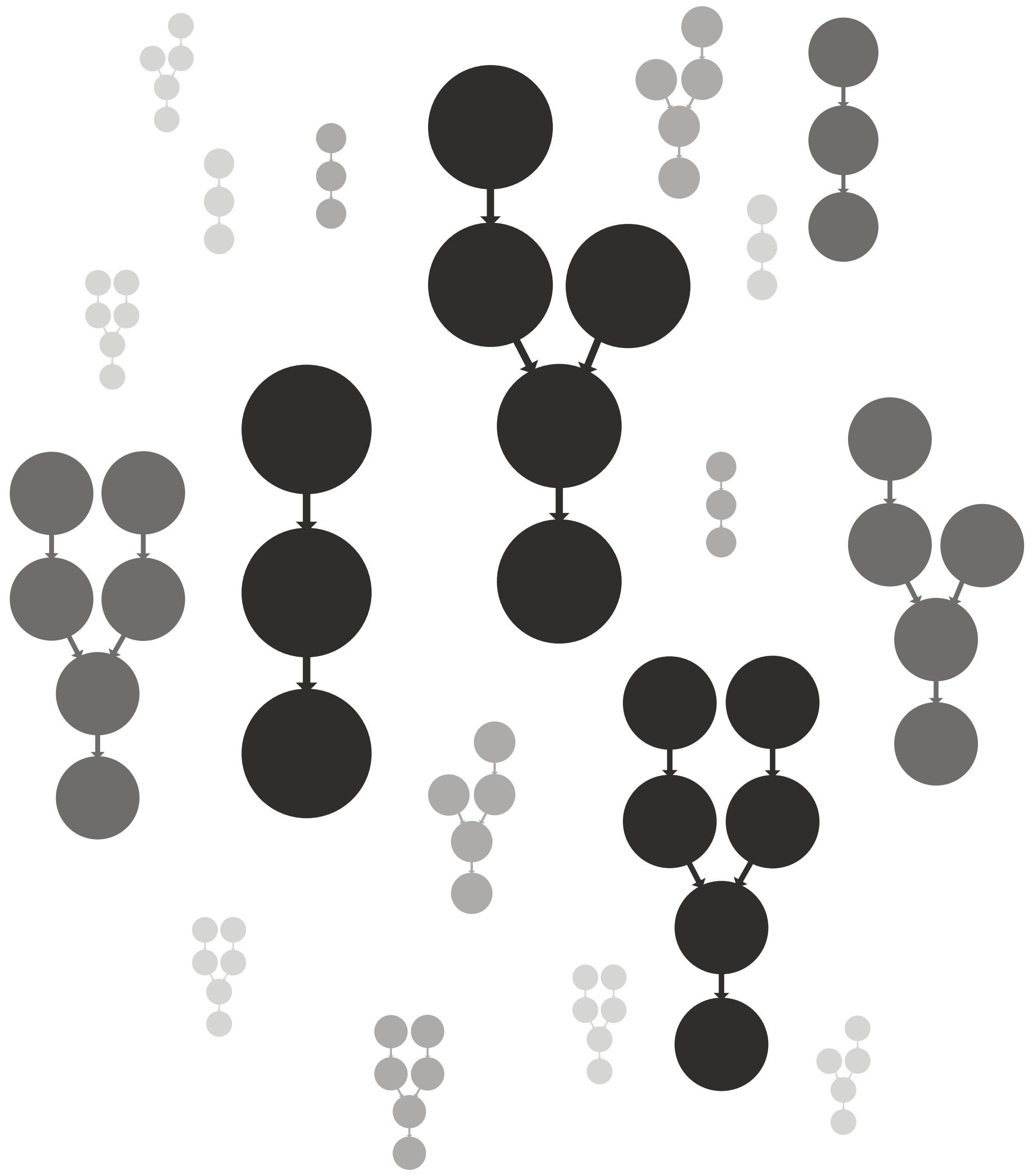
(1) Manually create THOUSANDS of (good and BAD) plans

(2) Extract features for each plan

(3) Execute ALL plans to collect labels (e.g., exec time)

*On four-node quad-core cluster

Proposed solution to generate training data



(1) Generate diverse synthetic plans based on initial small workload

(1) Manually create THOUSANDS of (good and BAD) plans

	F1	F2	F3
P1	…	…	…
P2	…	…	…
P3	…	…	…
…	…	…	…

(2) Extract features for each plan

	F1	F2	F3	L
P1	…	…	…	8s
P2	…	…	…	3s
P3	…	…	…	36s
…	…	…	…	…

real

estimated

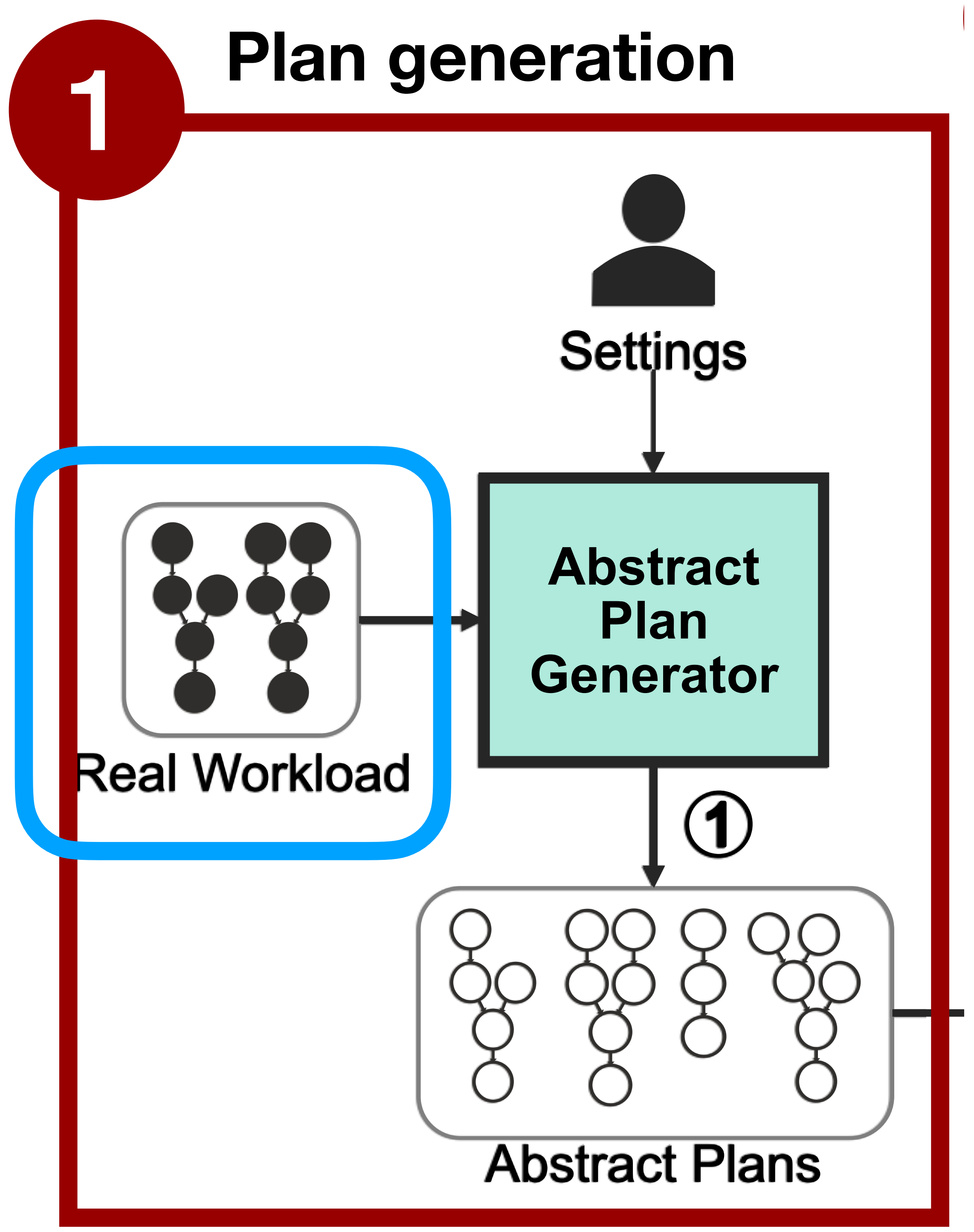
(3) Execute a small sample of plans and forecast the rest

(3) Execute ALL plans to collect labels (e.g., exec time)

DataFarm: Training data generator for learning-based QO

White-Box

Data-driven



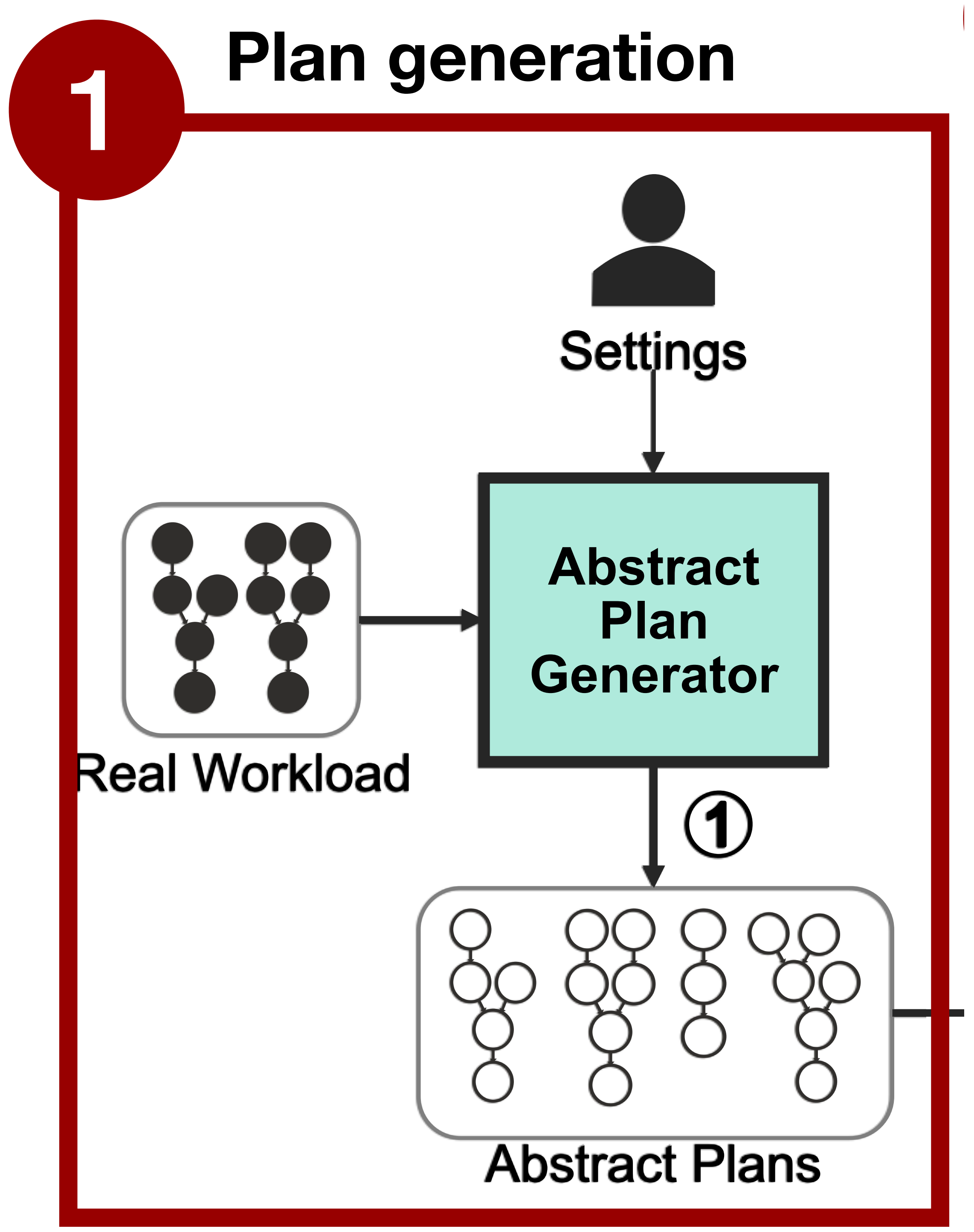
Explainable ML process

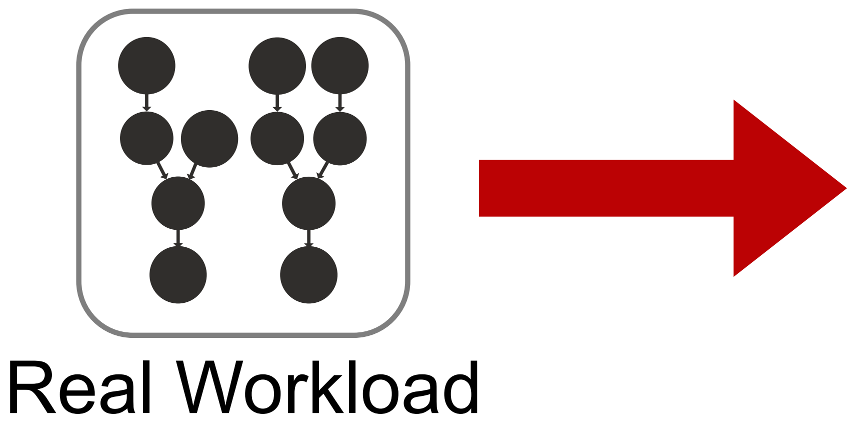
Easy to customise and debug

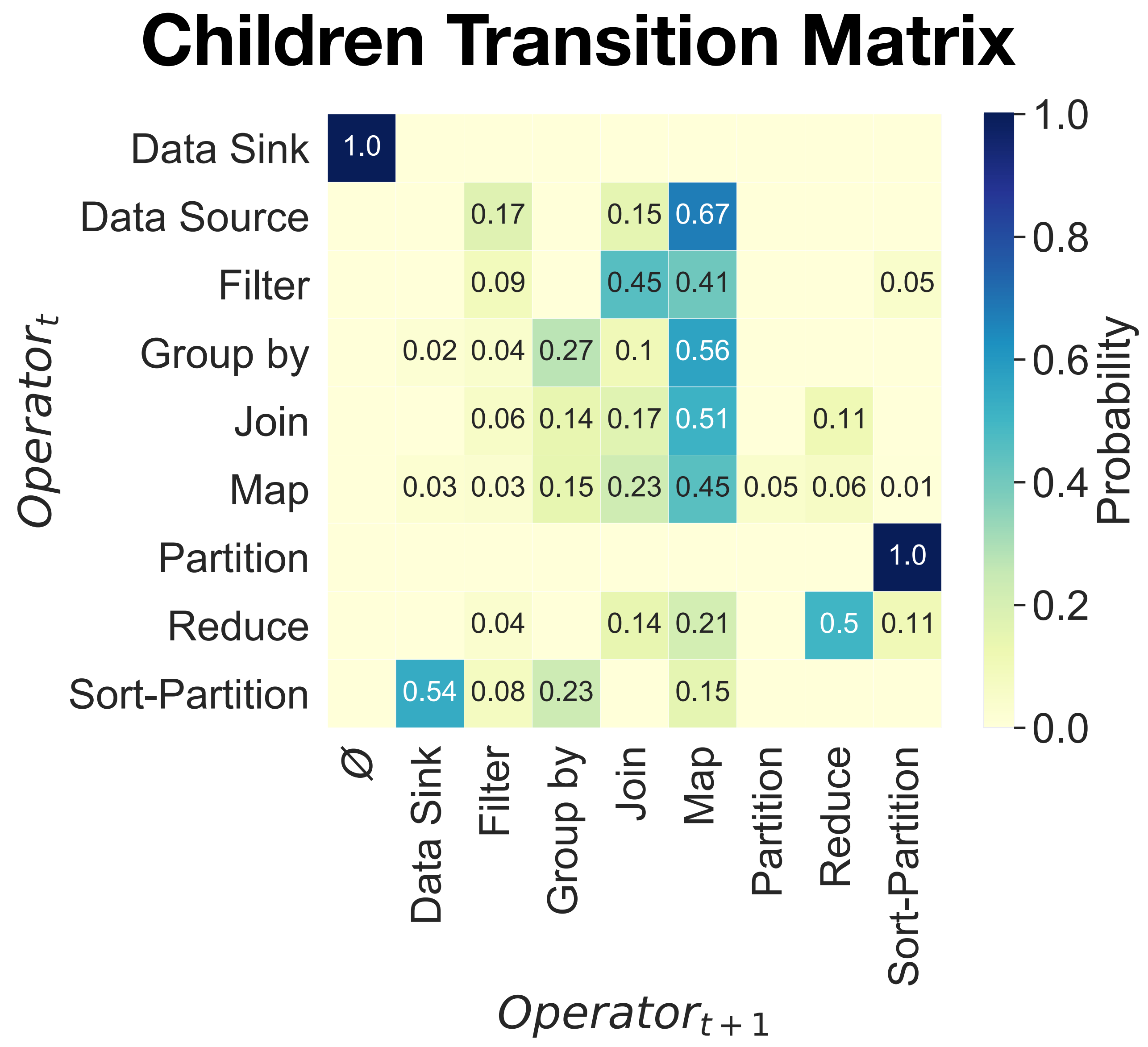
Expand your Training Limits! Generating Training Data for ML-based Data Management. ***SIGMOD 2021****: 1865-1878*
DataFarm: Farm Your ML-based Query Optimizer's Food! – Human-Guided Training Data Generation – ***CIDR 2022***
Farming Your ML-based Query Optimizer's Food. ***ICDE 2022 (best demo award)***

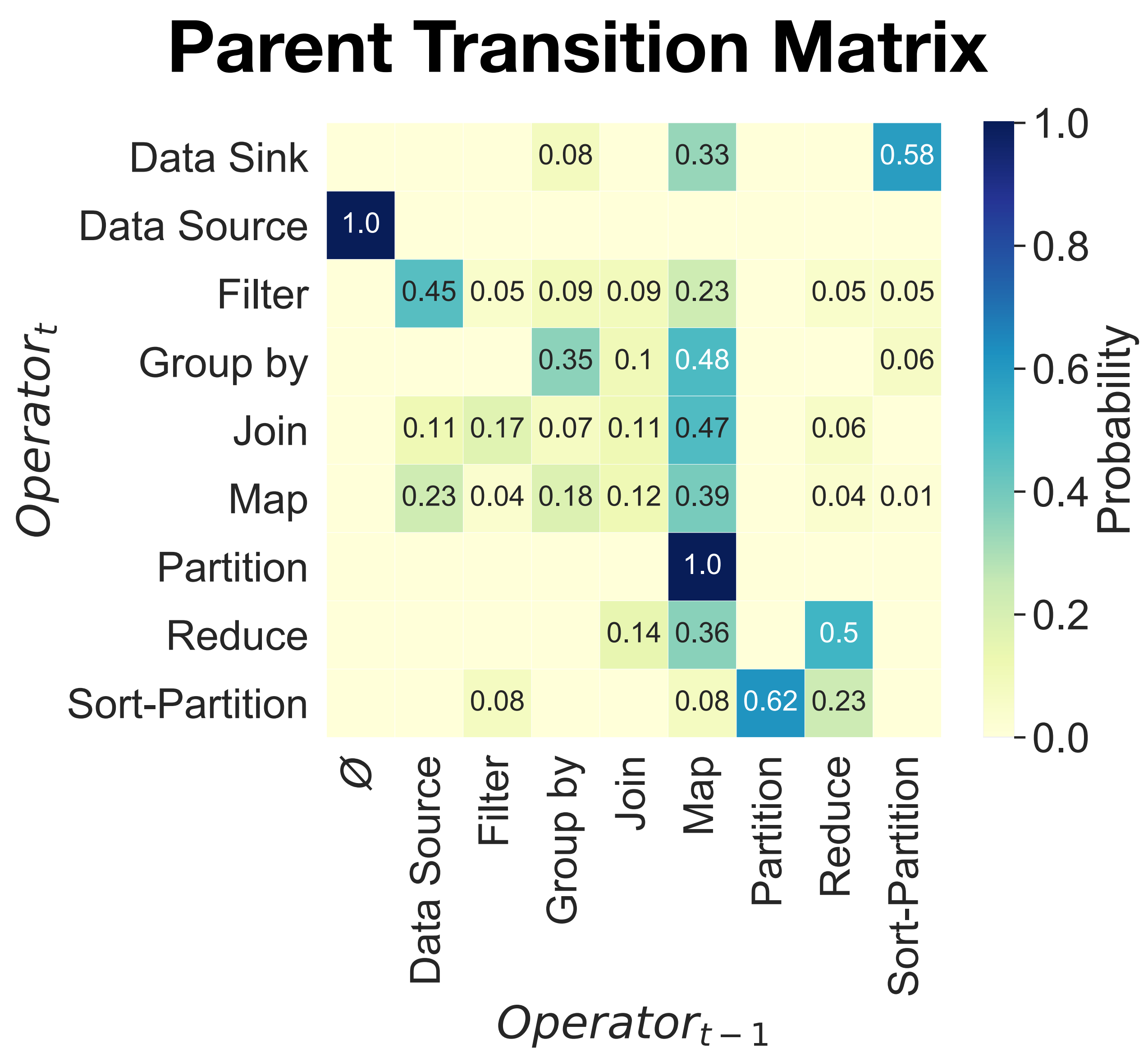
DataFarm: Training data generator for learning-based QO

✦ **Learns** real execution **patterns**



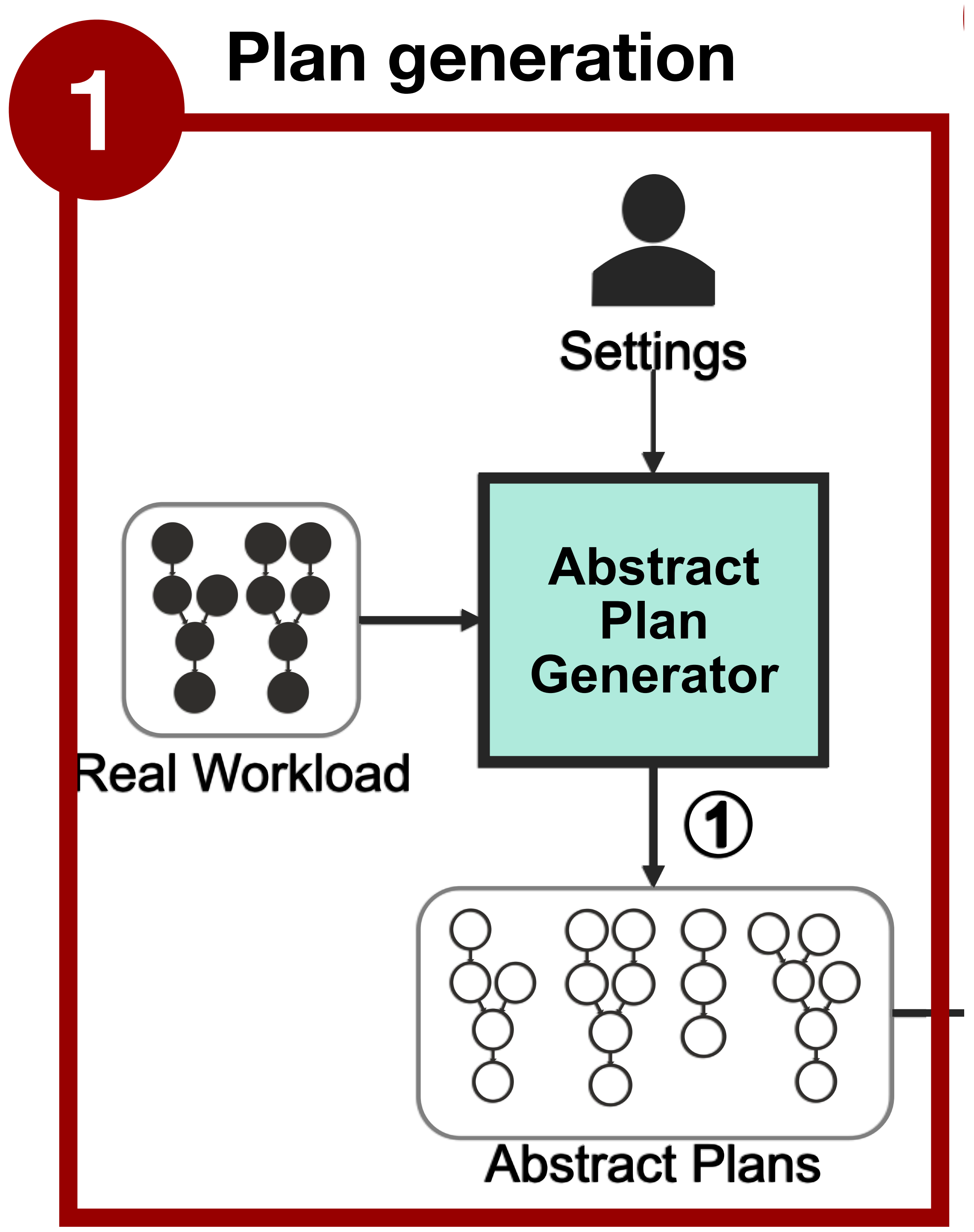


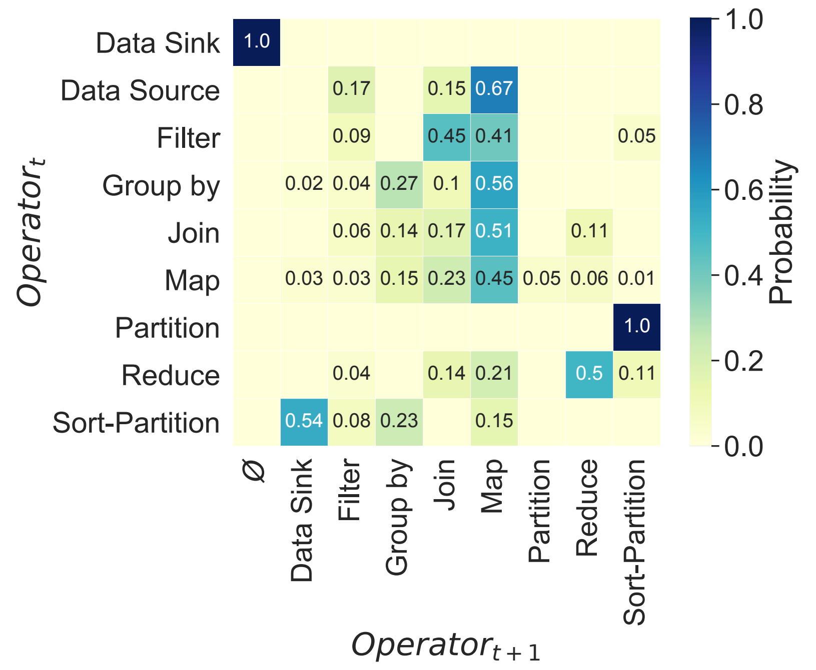


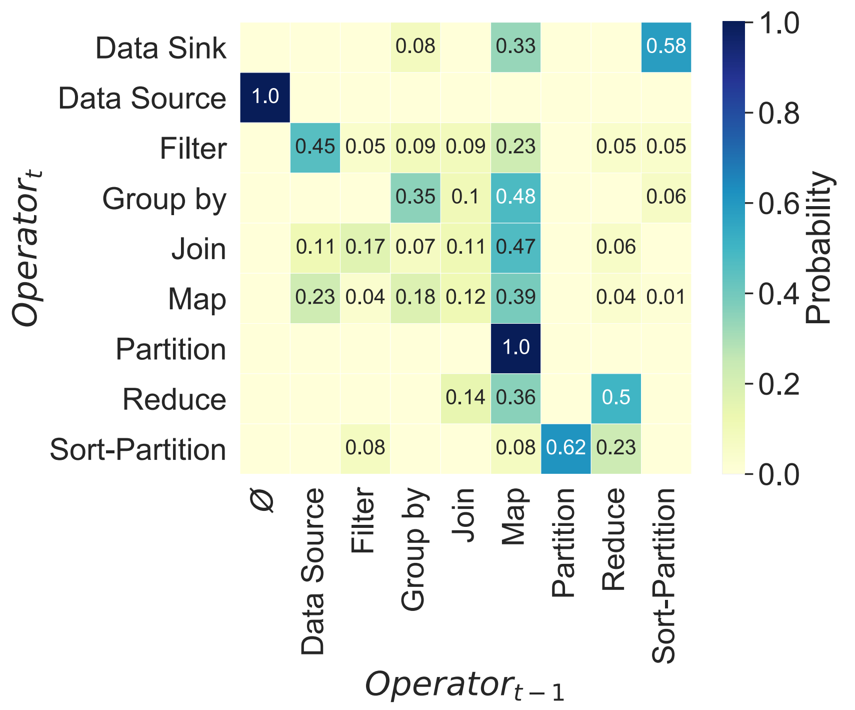


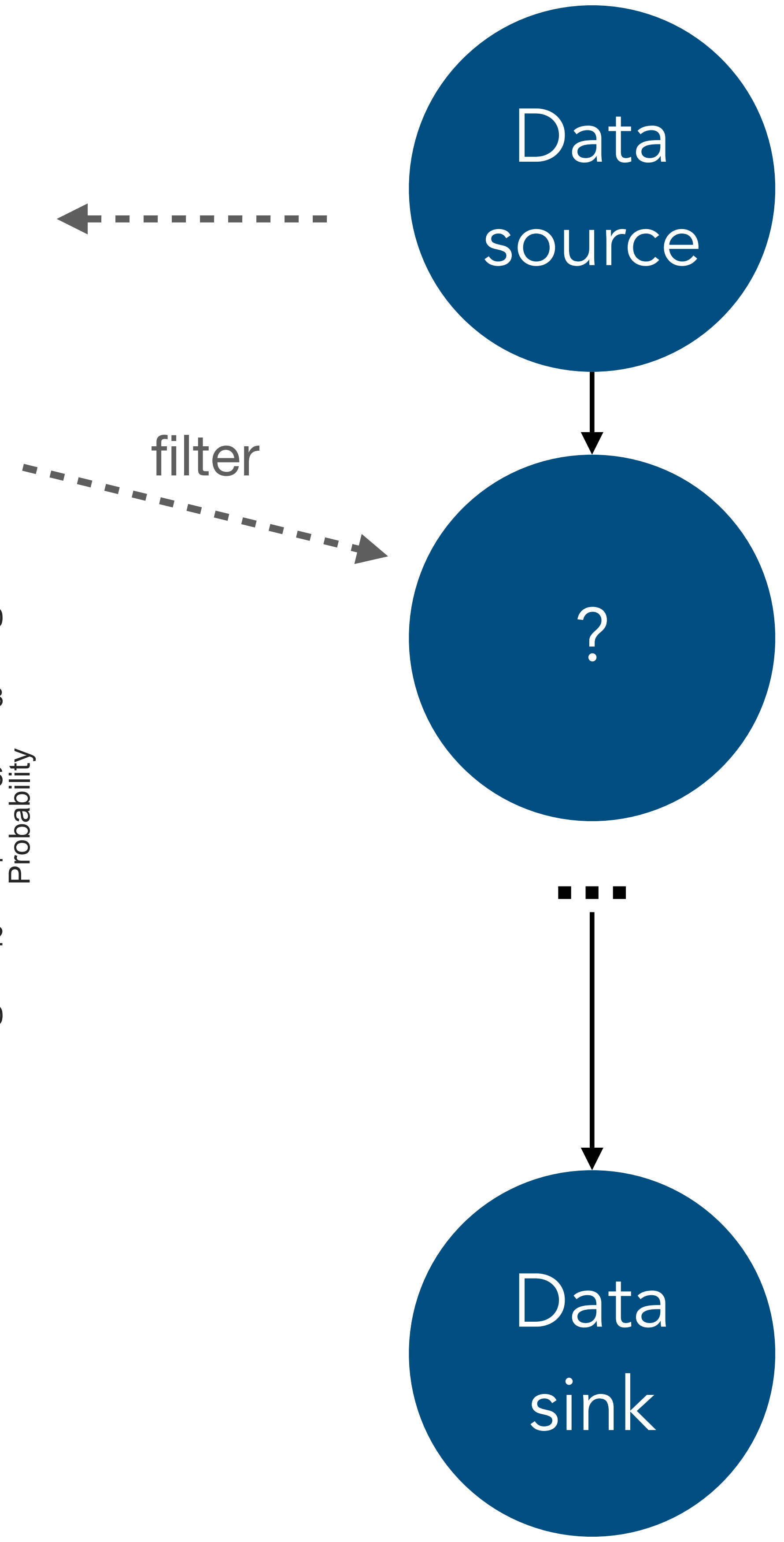
DataFarm: Training data generator for learning-based QO

- ✦ **Learns** real execution **patterns**
- ✦ **Generates** new **representative** plans

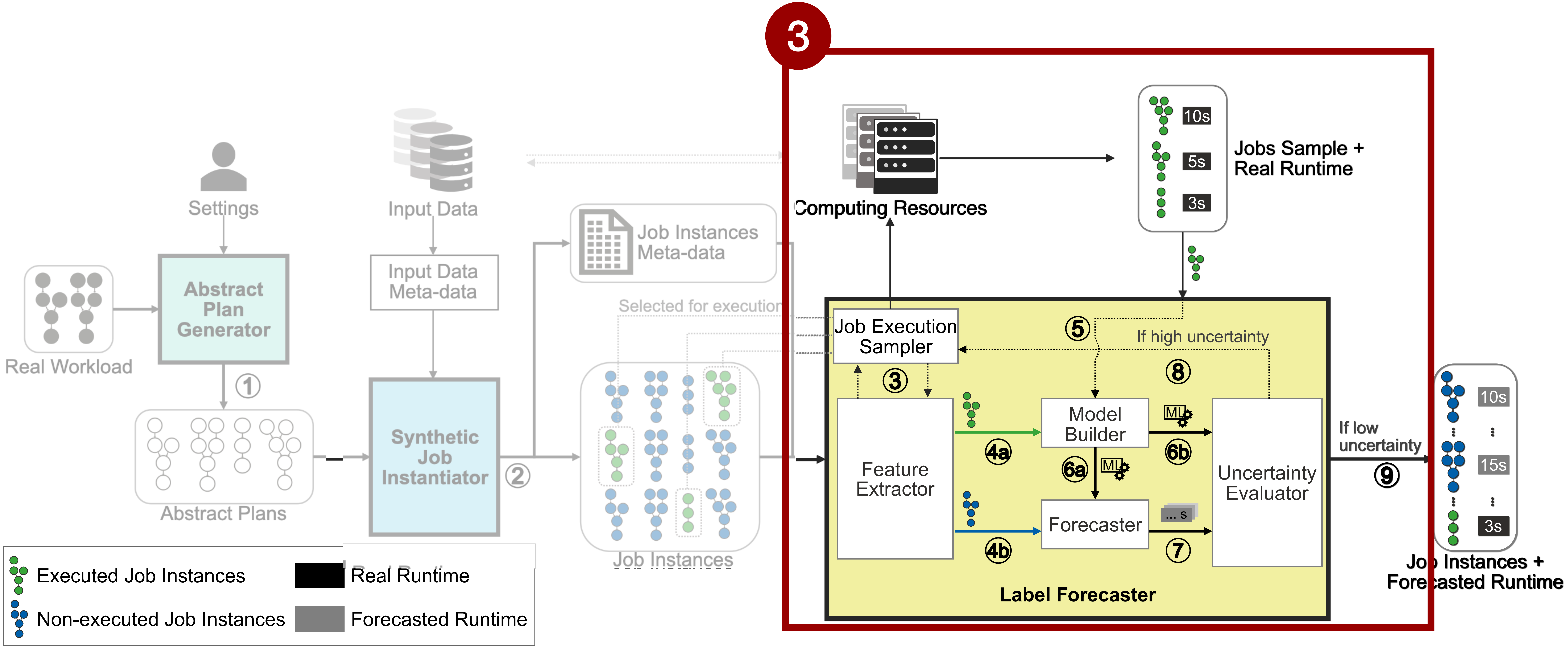






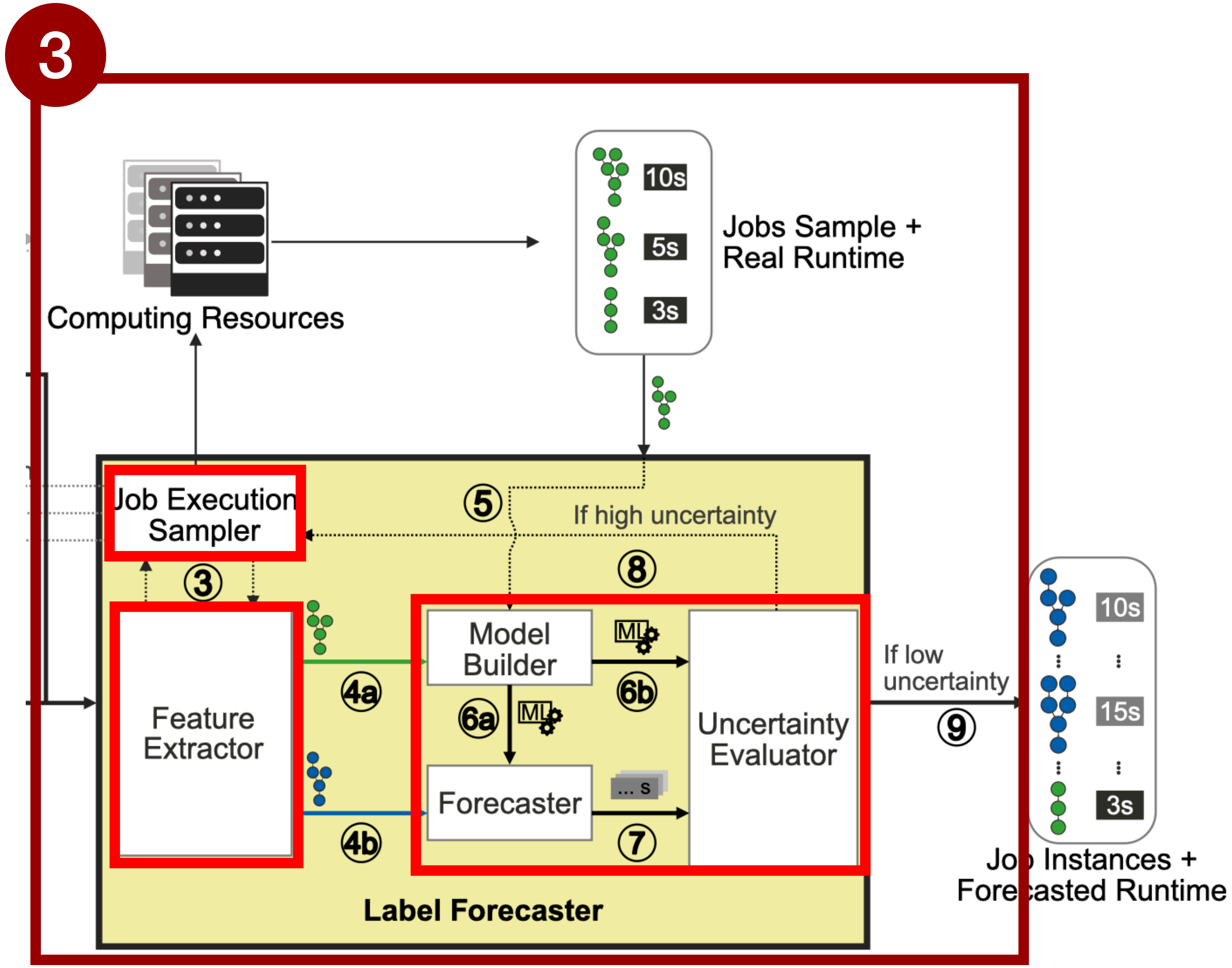


Focus on Label Forecaster

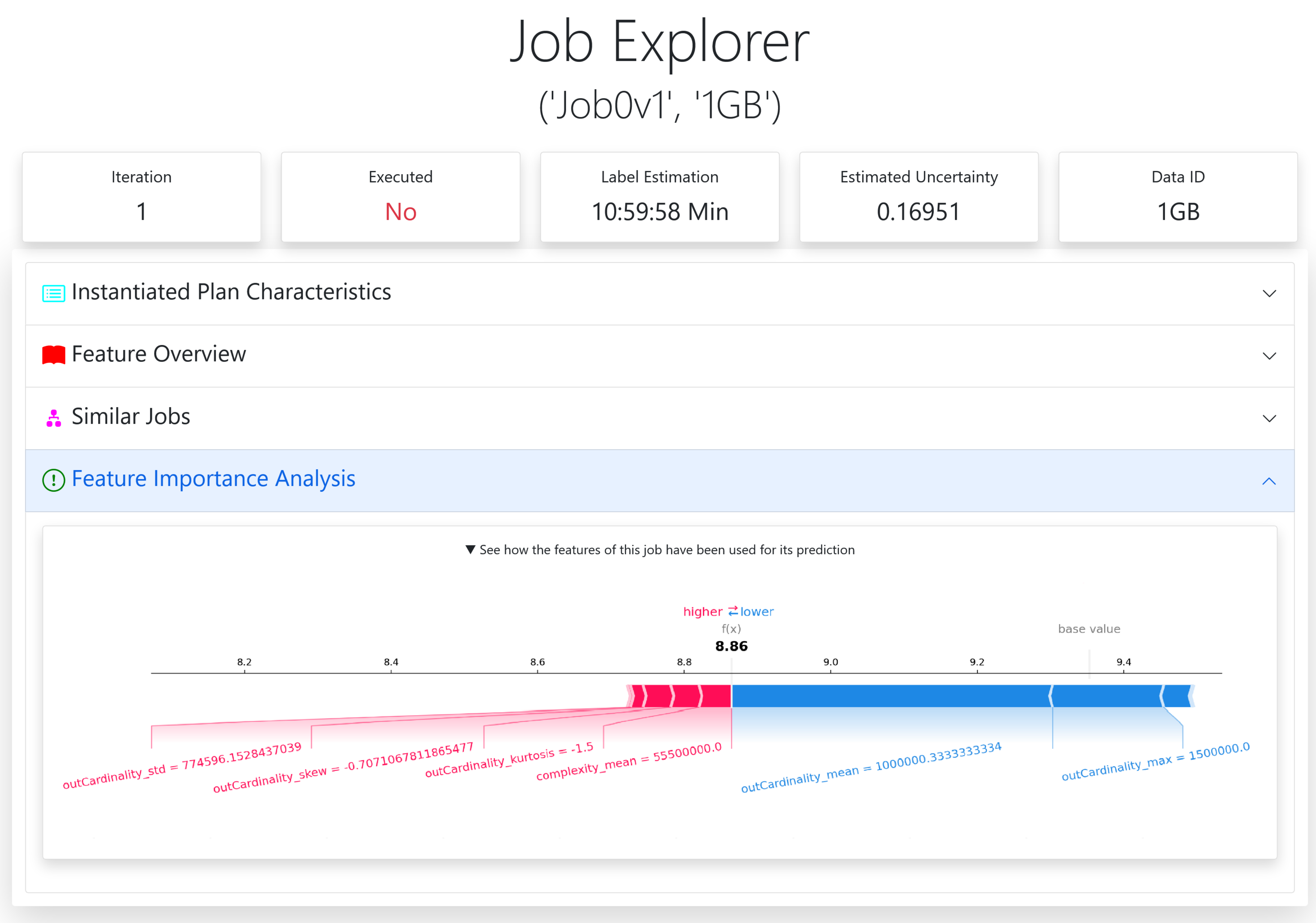


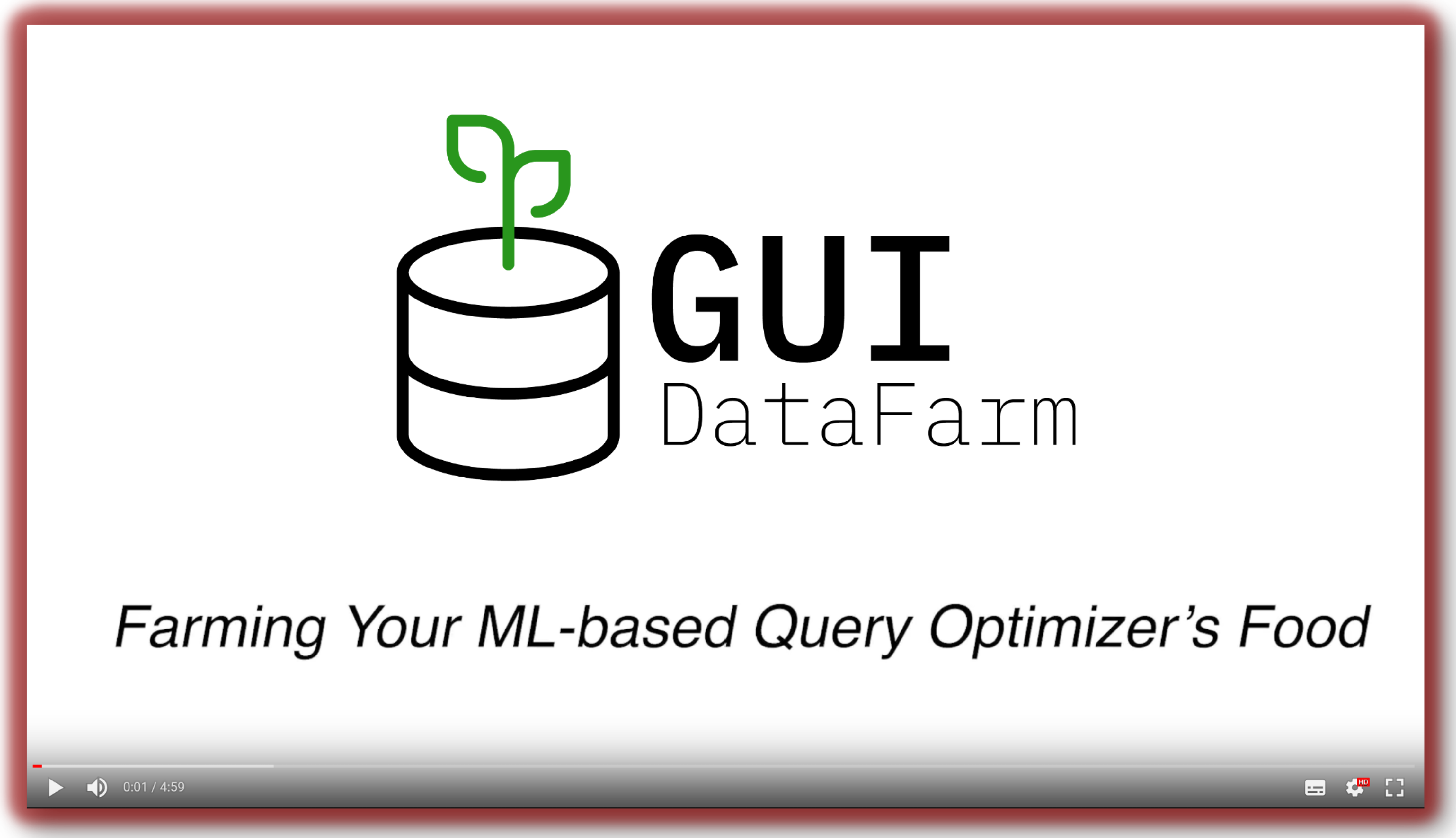
DataFarm: Label Forecaster

- Characterize jobs with **interpretable** features
- Find the smallest set of **representative** jobs to execute
- Predicts the **labels** and **uncertainty** for the non-executed jobs
- **Incrementally** execute more jobs and improve the model



GUIDataFarm: Human-in-the-loop data generation

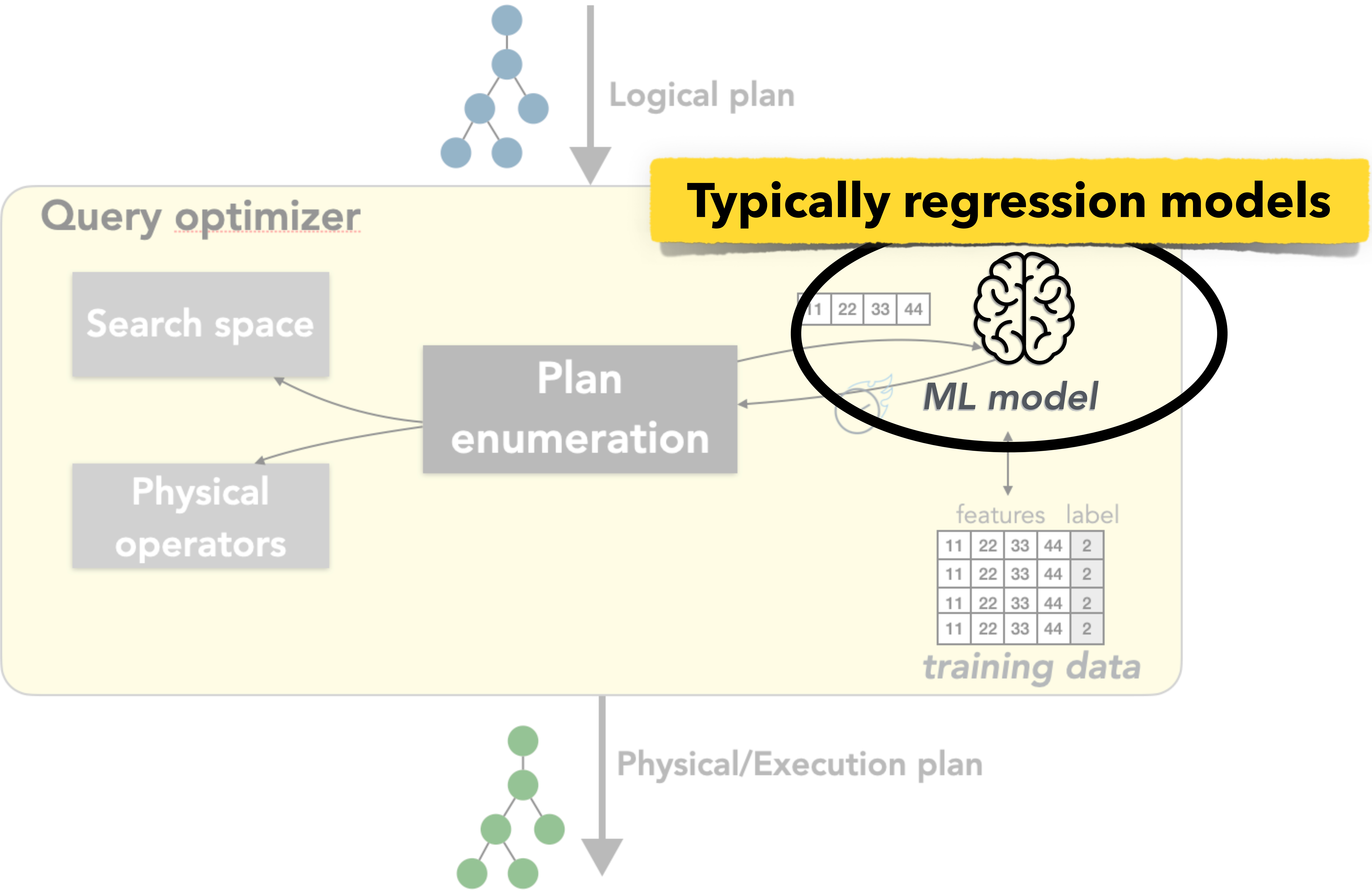




Github

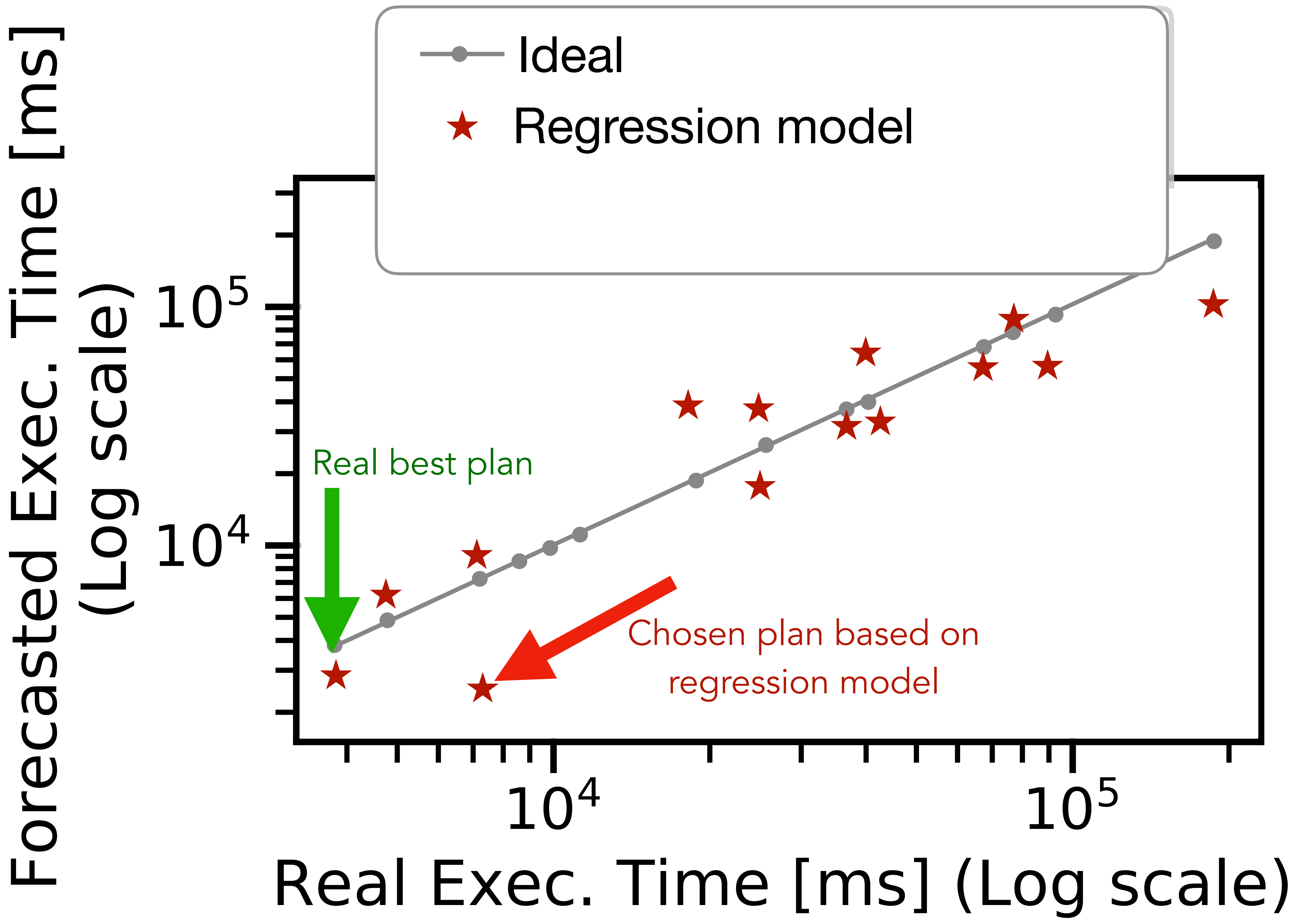
Video demonstration

Learning-based query optimization



Regression: hard to get completely right

Effect of regression models errors

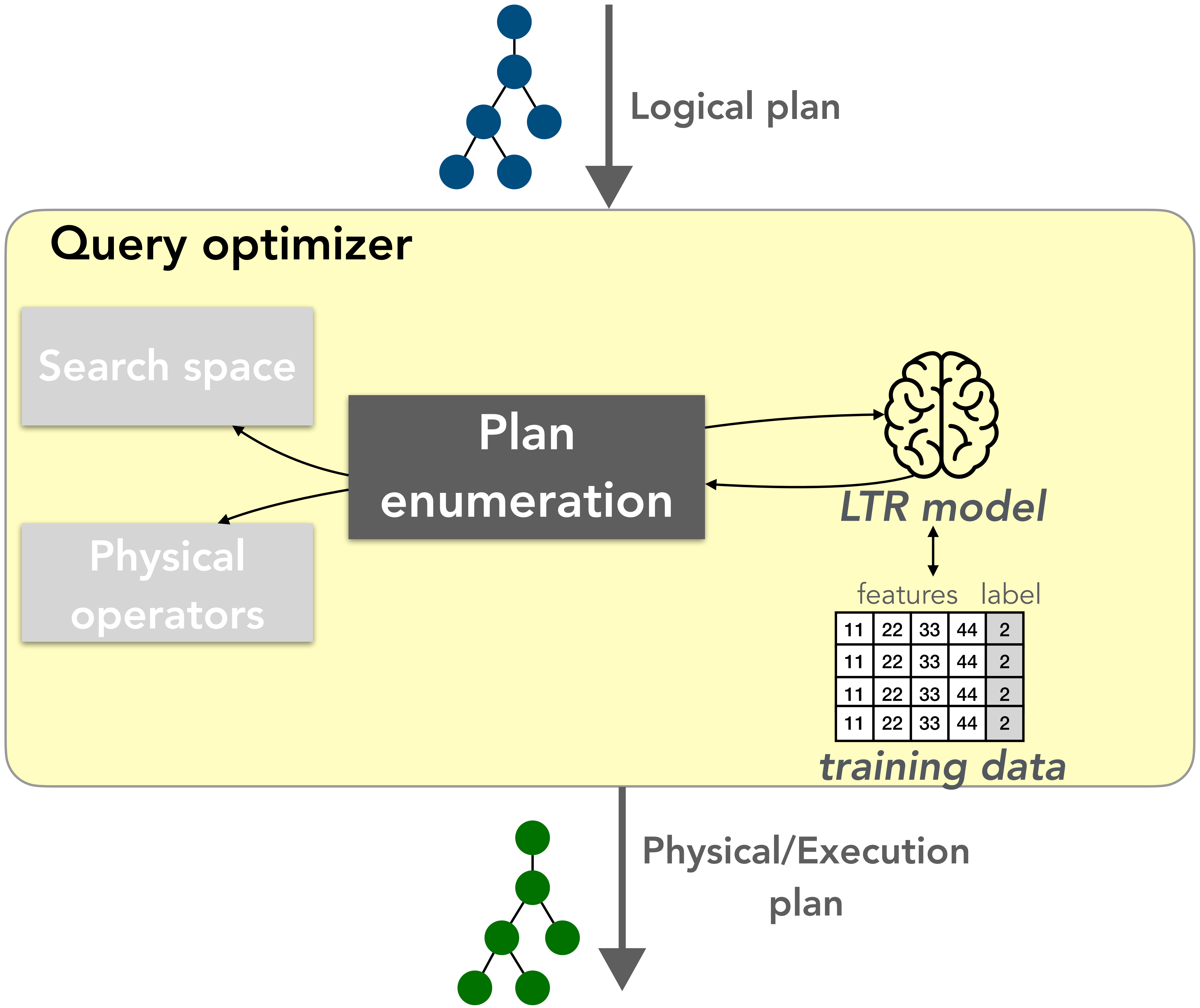


~ 2x worse execution time

What really matters in QO is the **relative order** of the plans

Can we leverage learning-to-rank models?

Learning-to-rank (LTR) in query optimization



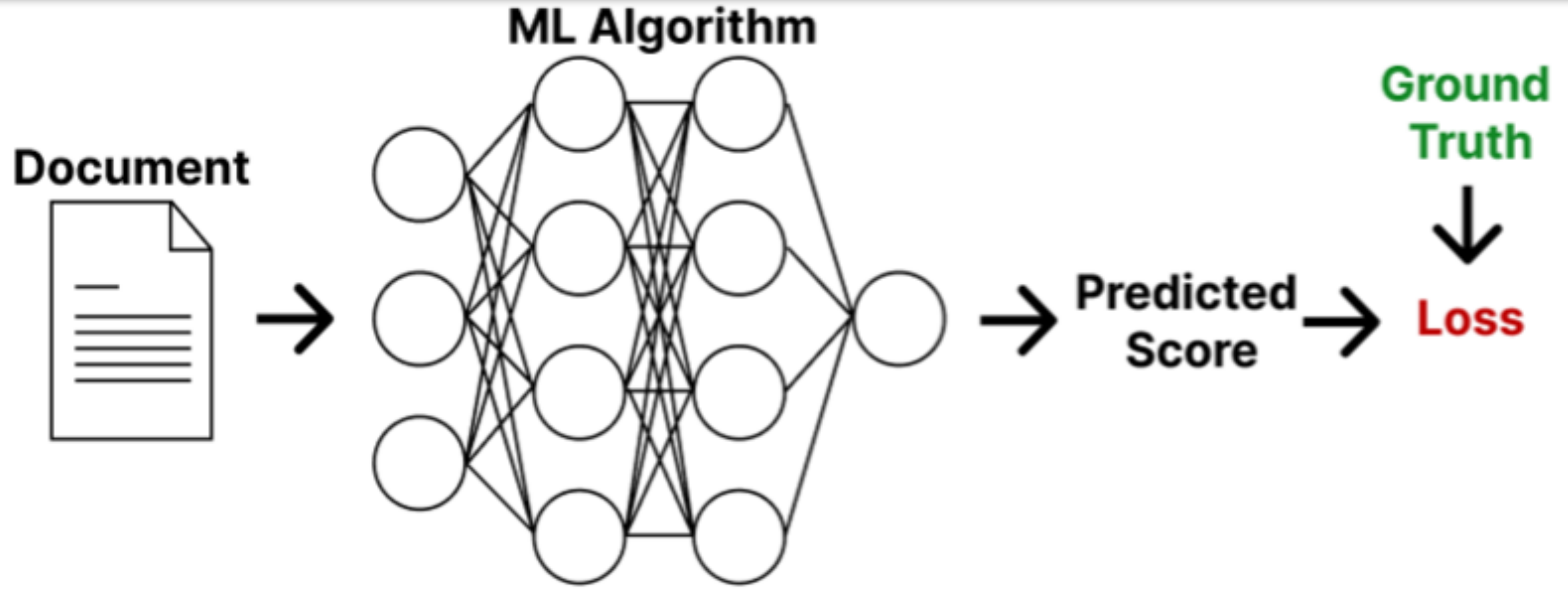
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2

- **Questions**
  - Type of LTR
  - Model architecture and features
  - Ranking scores from training data

Learn What Really Matters: A Learning-to-Rank Approach for ML-based Query Optimization. ***BTW 2023***

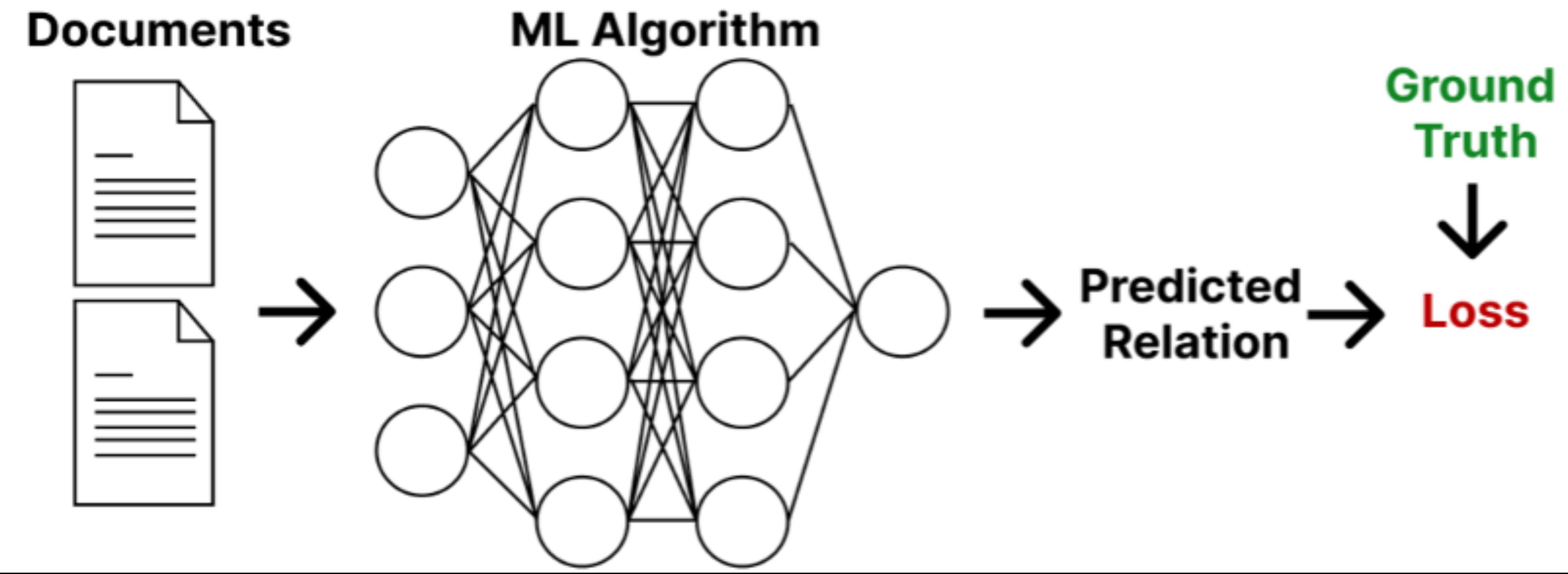
Learning-to-rank (LTR) approach

Pointwise



No comparison among items

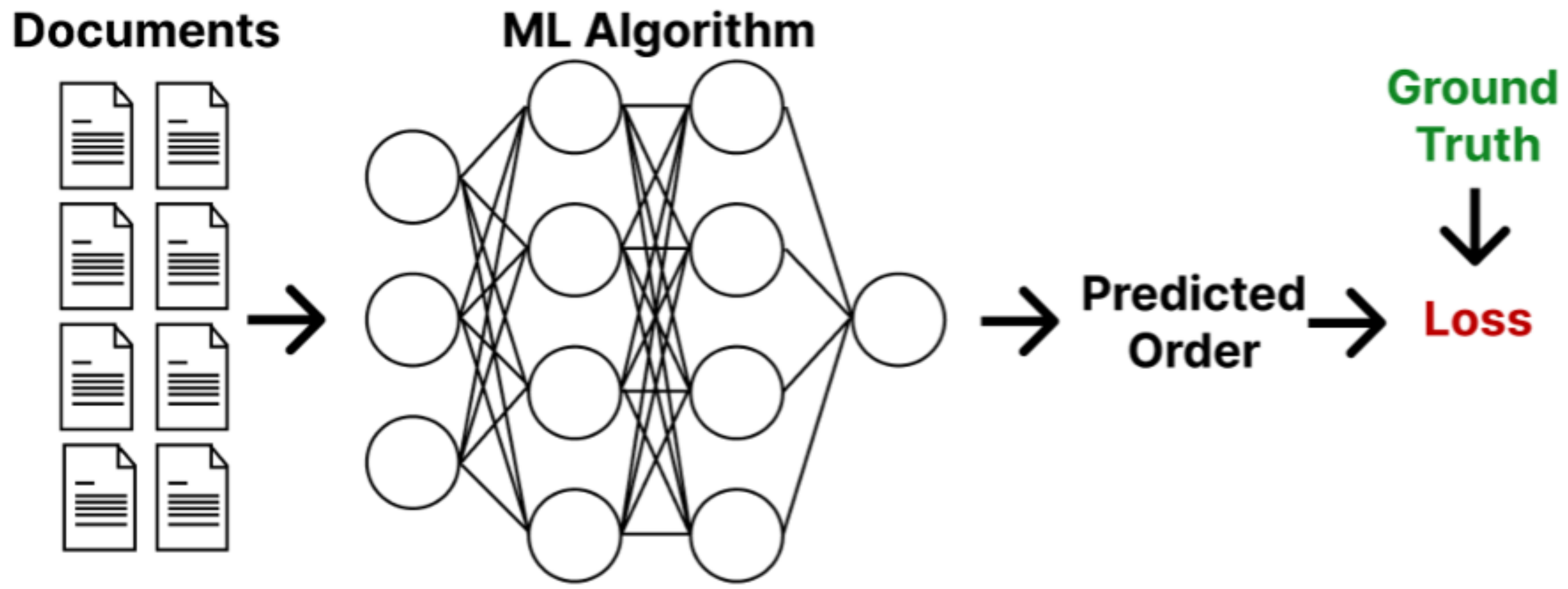
Pairwise



Too expensive

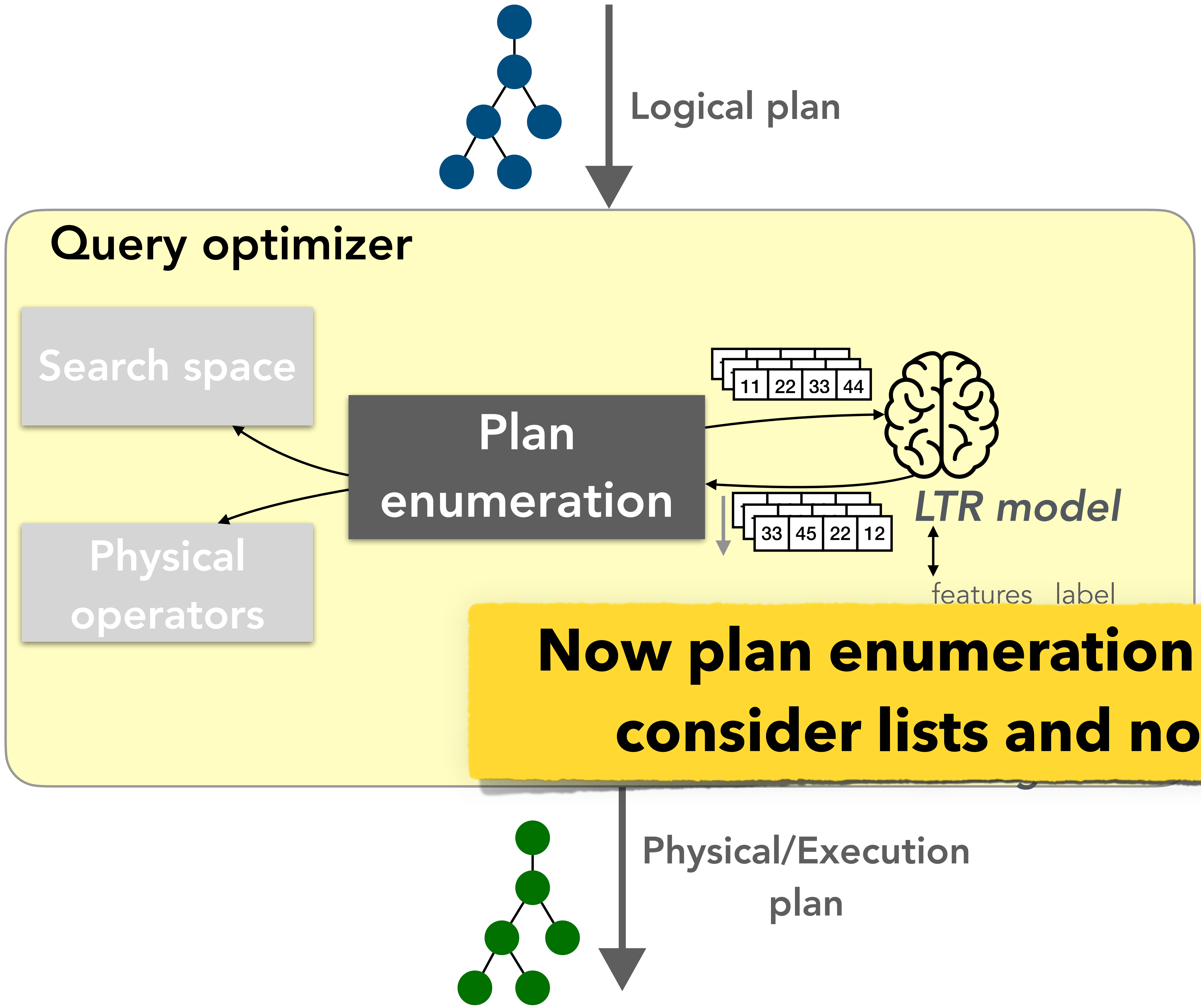
Assumes pairs are i.i.d

Listwise



Emphasises the ranking objective

Learning-to-rank (LTR) in query optimization



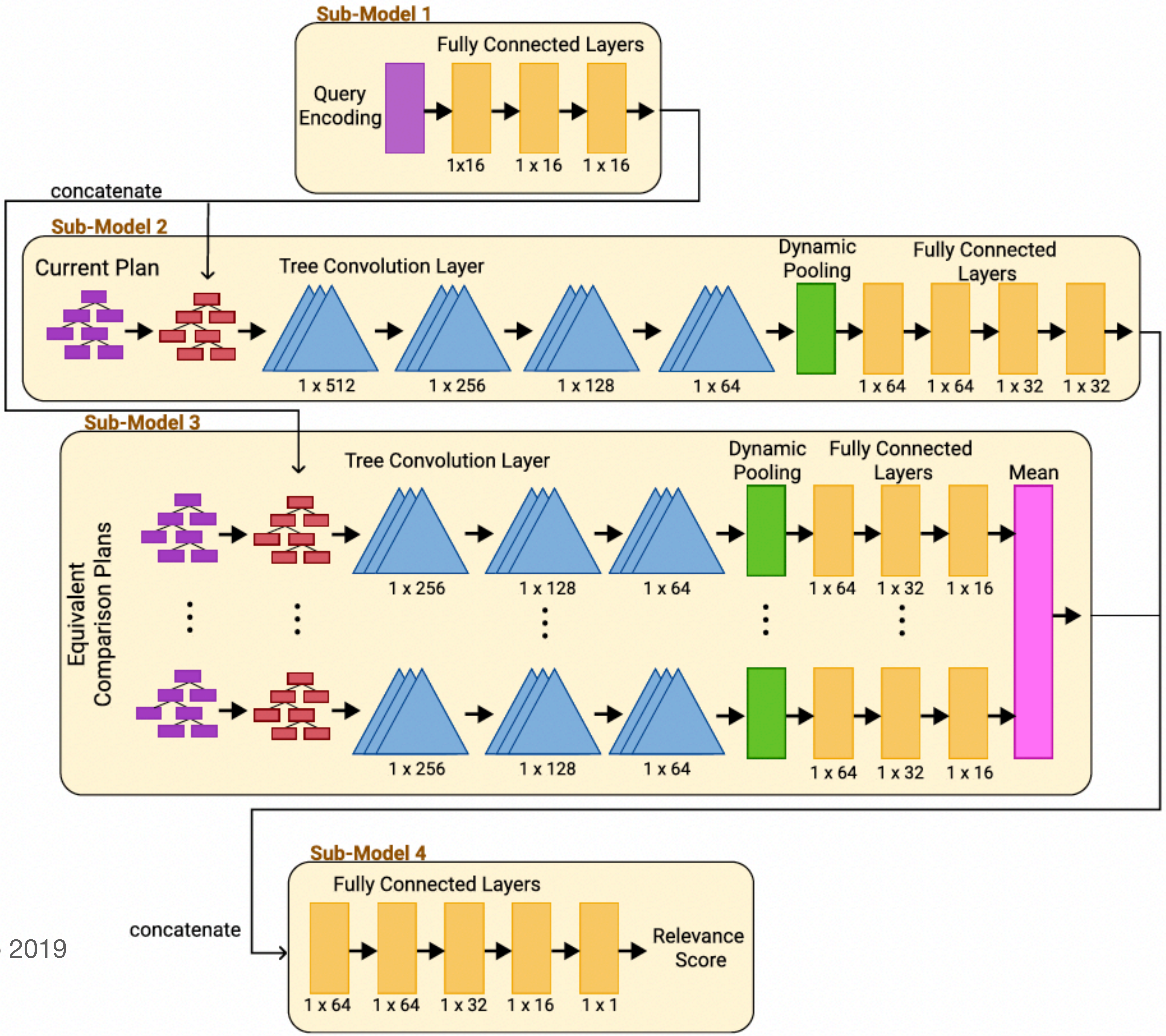
Now plan enumeration needs to consider lists and not pairs

- **Questions**
  - Type of LTR
  - Model architecture and features
  - Ranking scores from training data

LTR model architecture

Inspired by FATE [1] and Neo [2]

Each plan against the rest equivalent plans



[1] K. Pfannschmidt et al.: Deep architectures for learning context-dependent ranking functions. CoRR abs/1803.05796 (2018)]

[2] R. Markus et al.: Neo: A Learned Query Optimizer. In: PVLDB 12(11) 2019

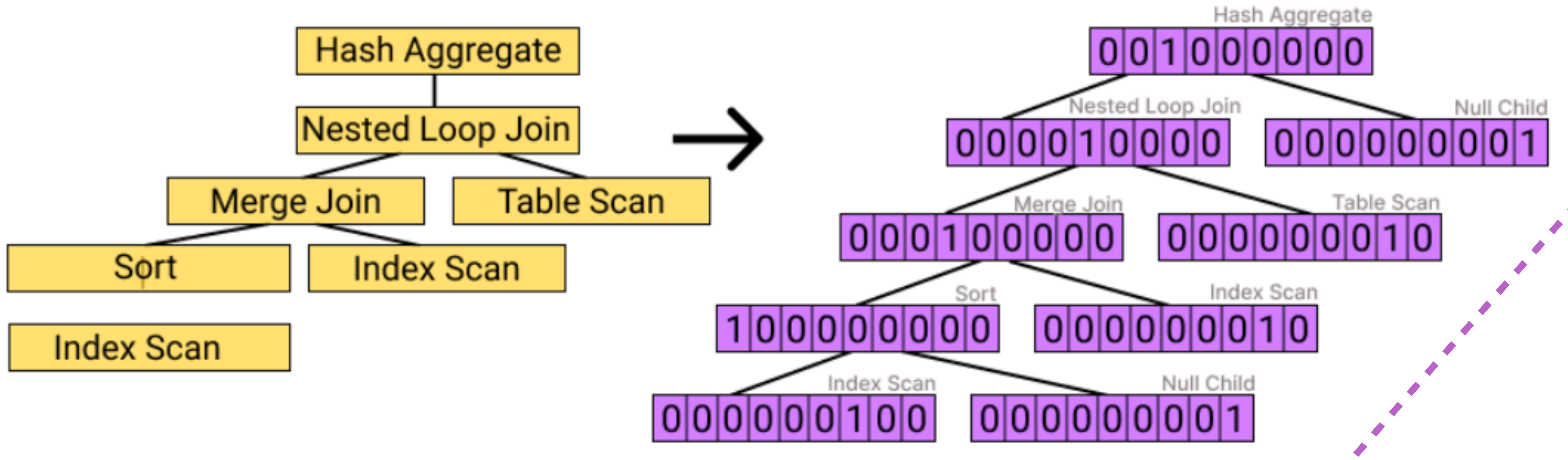
Featurization

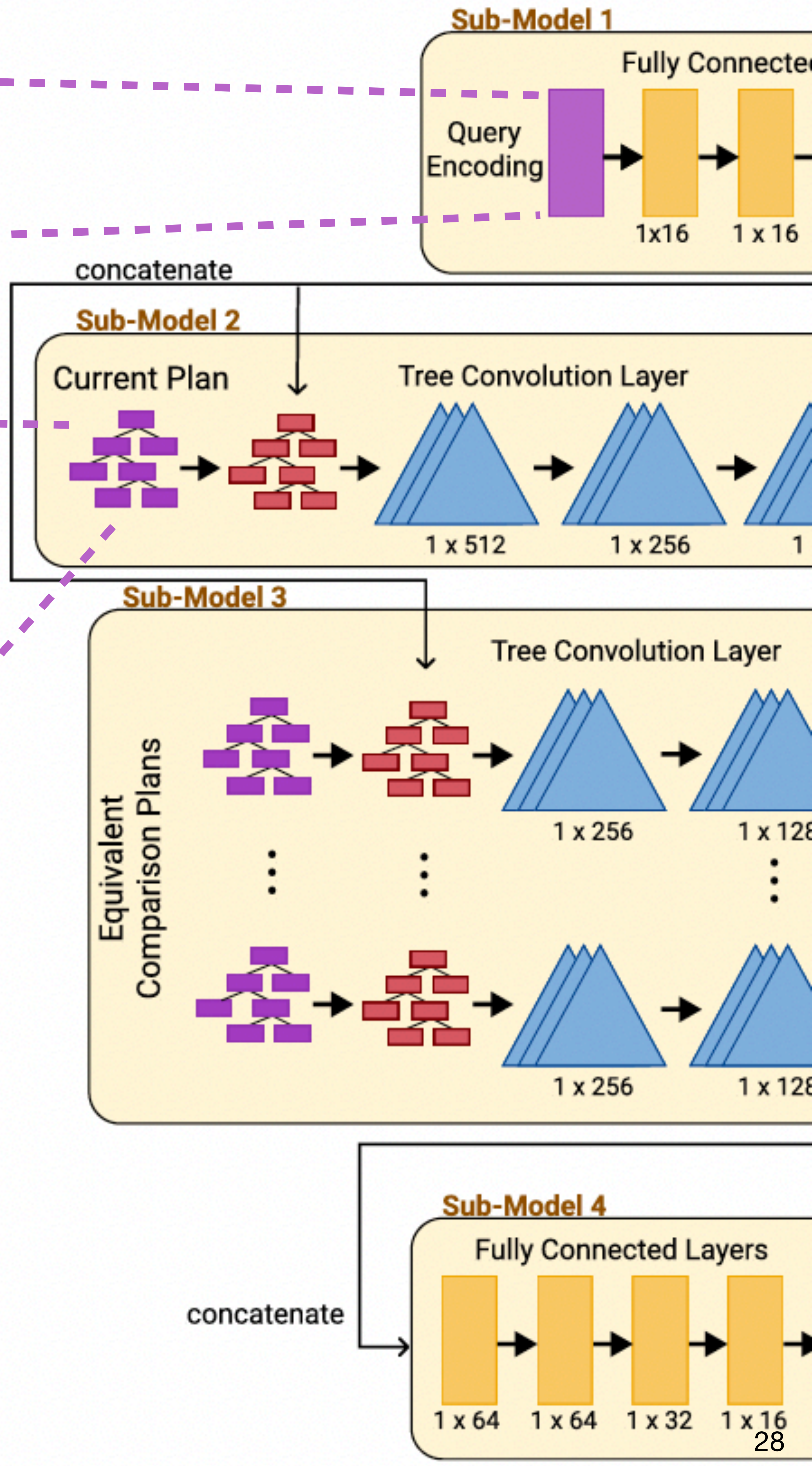
Query encoding (logical)

Has Order By	Has Group By	Number of Joins	Estimated Number of Rows	Max. Relation	Min. Relation

Plan encoding (physical)

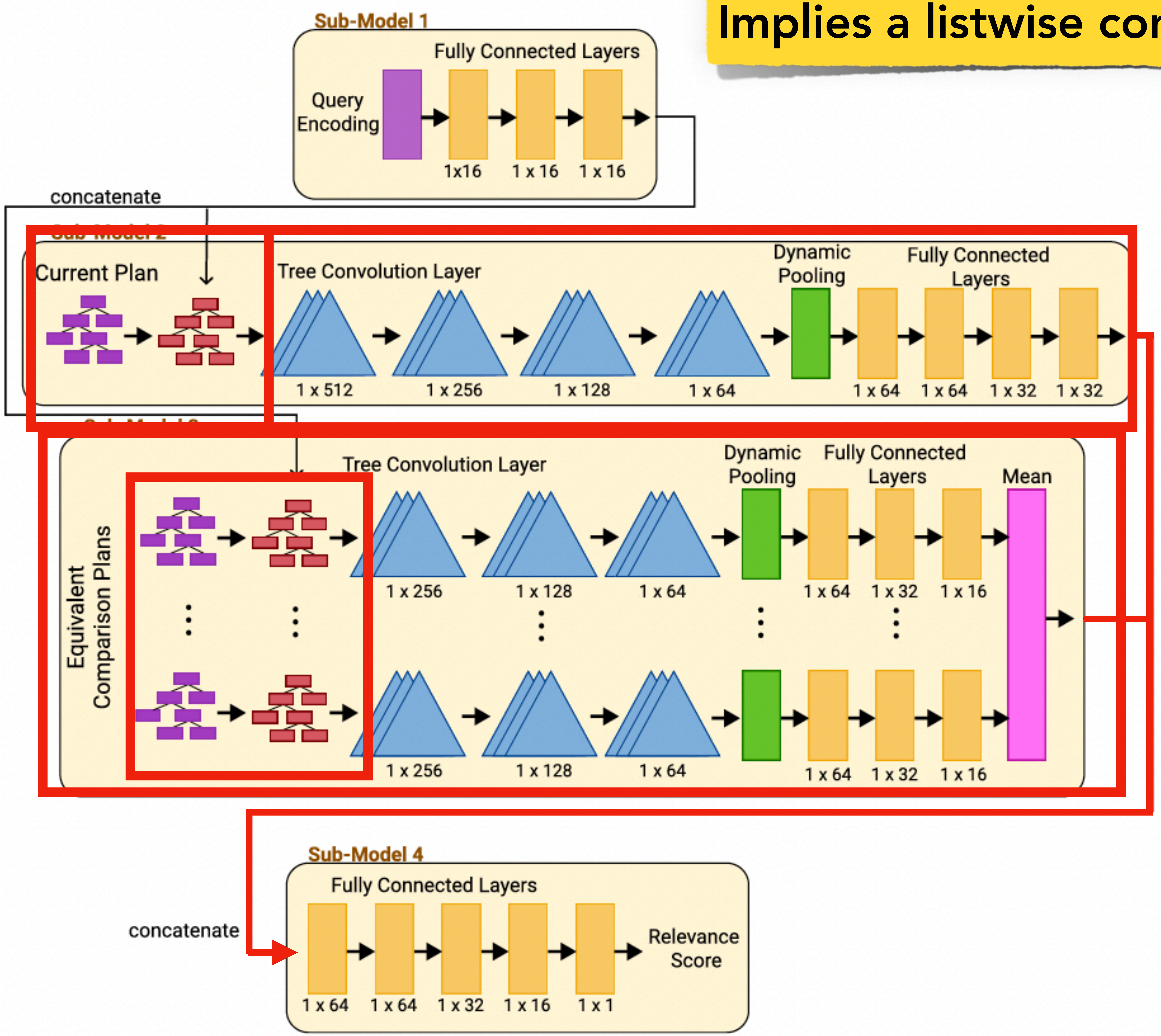
Sort	Stream Aggregate	Hash Aggregate	Merge Join	Nested Loop Join	Hash Join	Index Scan	Table Scan	Null	Estimated Rows



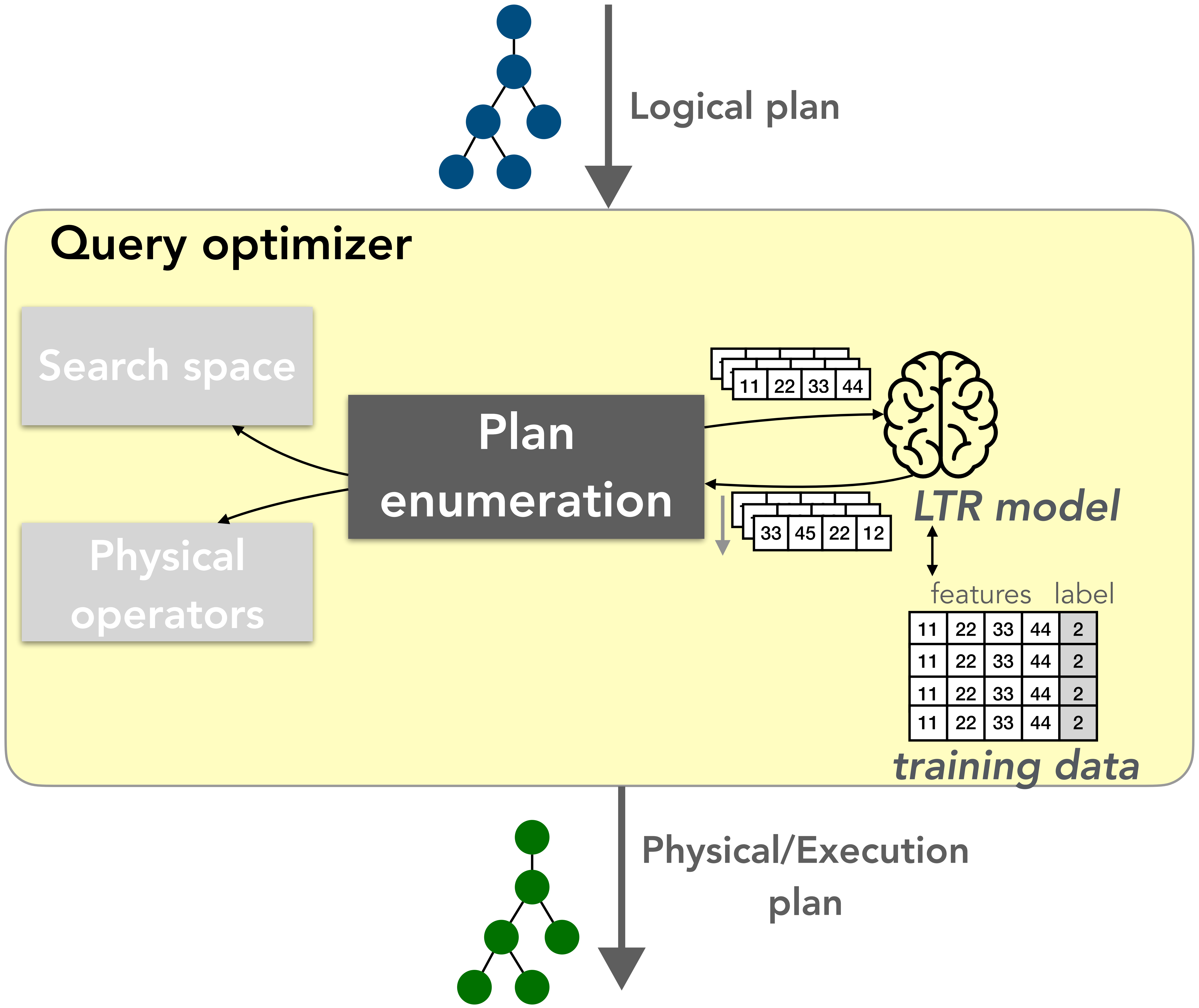


LTR model architecture

Implies a listwise comparison of plans



Learning-to-rank (LTR) in query optimization



- ✦ **Questions**
  - ✦ Type of LTR
  - ✦ Model architecture and features
  - ✦ Ranking scores from training data

Scoring training plans

Execution time

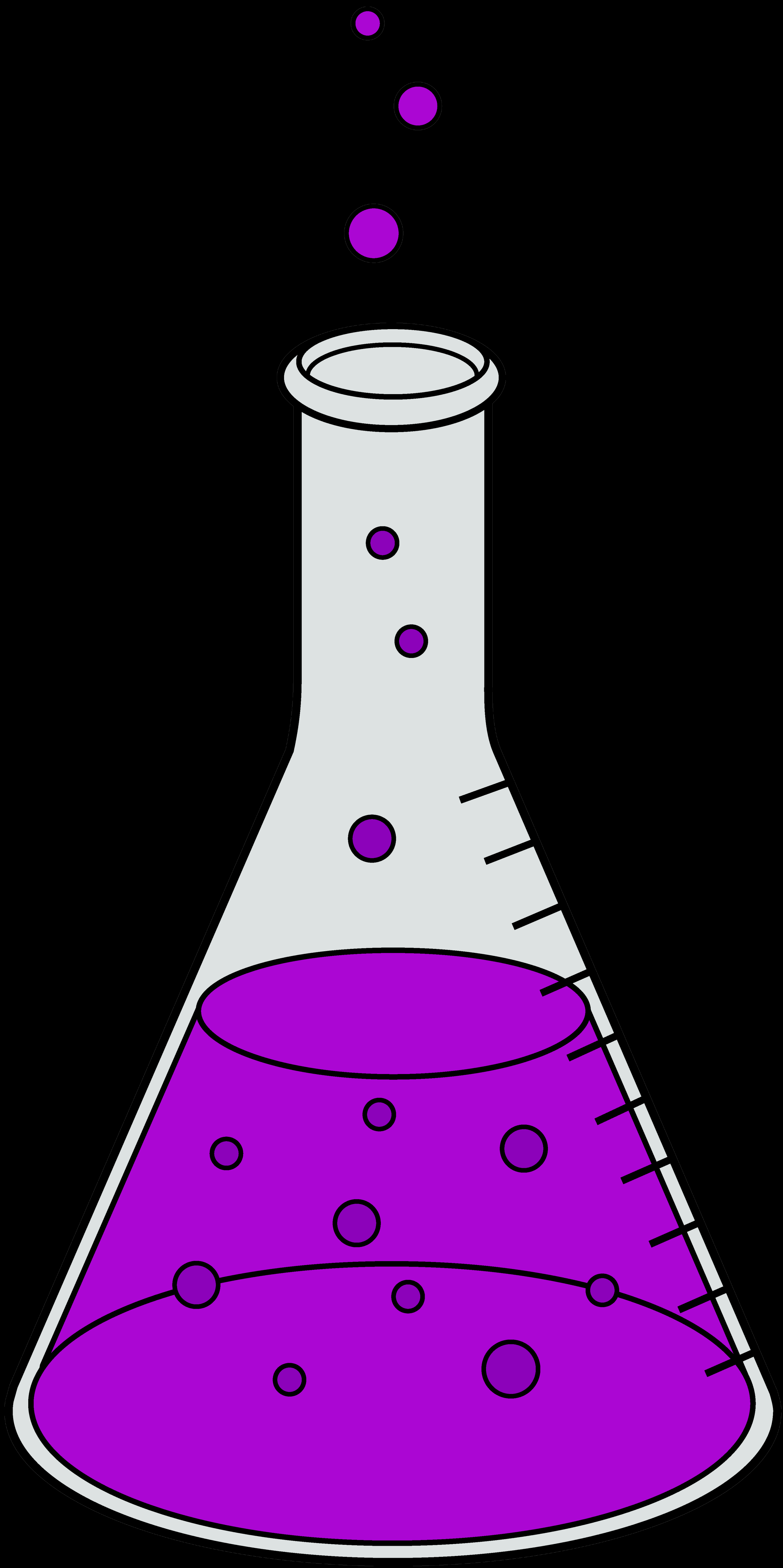
features label

11	22	33	44	78
11	22	33	44	23
11	22	33	44	99
11	22	33	44	7

training data

- Option 1: Use execution time, sort, extract rank
  - Information loss on how two plans differ
- Option 2: Devise a **score function**: given the runtime of a plan and a maximum score value, output its score w.r.t. a list of plans and their runtimes
  - **Local linear** score function
  - **Global** agglomerative function

Preliminary Results



Experimental setup

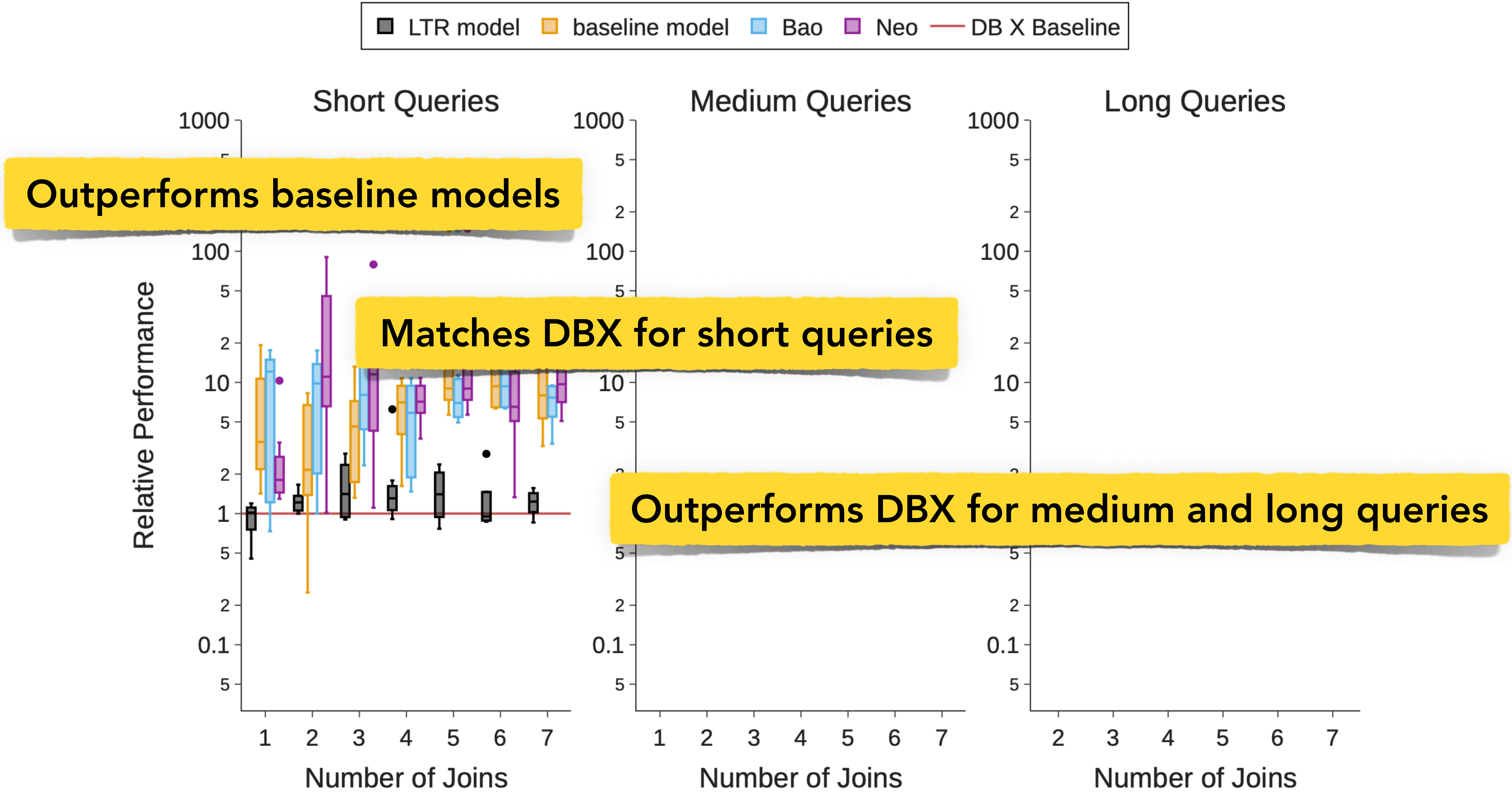
- Datasets: TPC-H and JOB
- Queries:

Query types	1 Join	2 Joins	3 Joins	4 Joins	5 Joins	6 Joins	7 Joins
Short (< 2s)	8	8	8	8	8	8	8
Medium (>=2s & < 30s)	2	8	8	8	8	8	8
Long (>= 30s)	0	2	4	7	8	8	5

- ~25K execution plans produced by DataFarm [1]
- Baselines:
  - DB X (cost-based optimizer)
  - Neo [2]
  - Bao [3]
  - baseline model (pairwise LTR)

[1] F. Ventura et al: Expand your Training Limits! Generating Training Data for ML-based Data Management. SIGMOD 2021
[2] R. Markus et al.: Neo: A Learned Query Optimizer. PVLDB 12(11) 2019
[3] R. Markus et al.: Bao: Making Learned Query Optimization Practical. SIGMOD 2021

Performance on TPC-H (1 GB)



Learning-based Query Optimization

What are we still missing?

Training data generation

GUI DataFarm

Github

Learning-to-rank methods