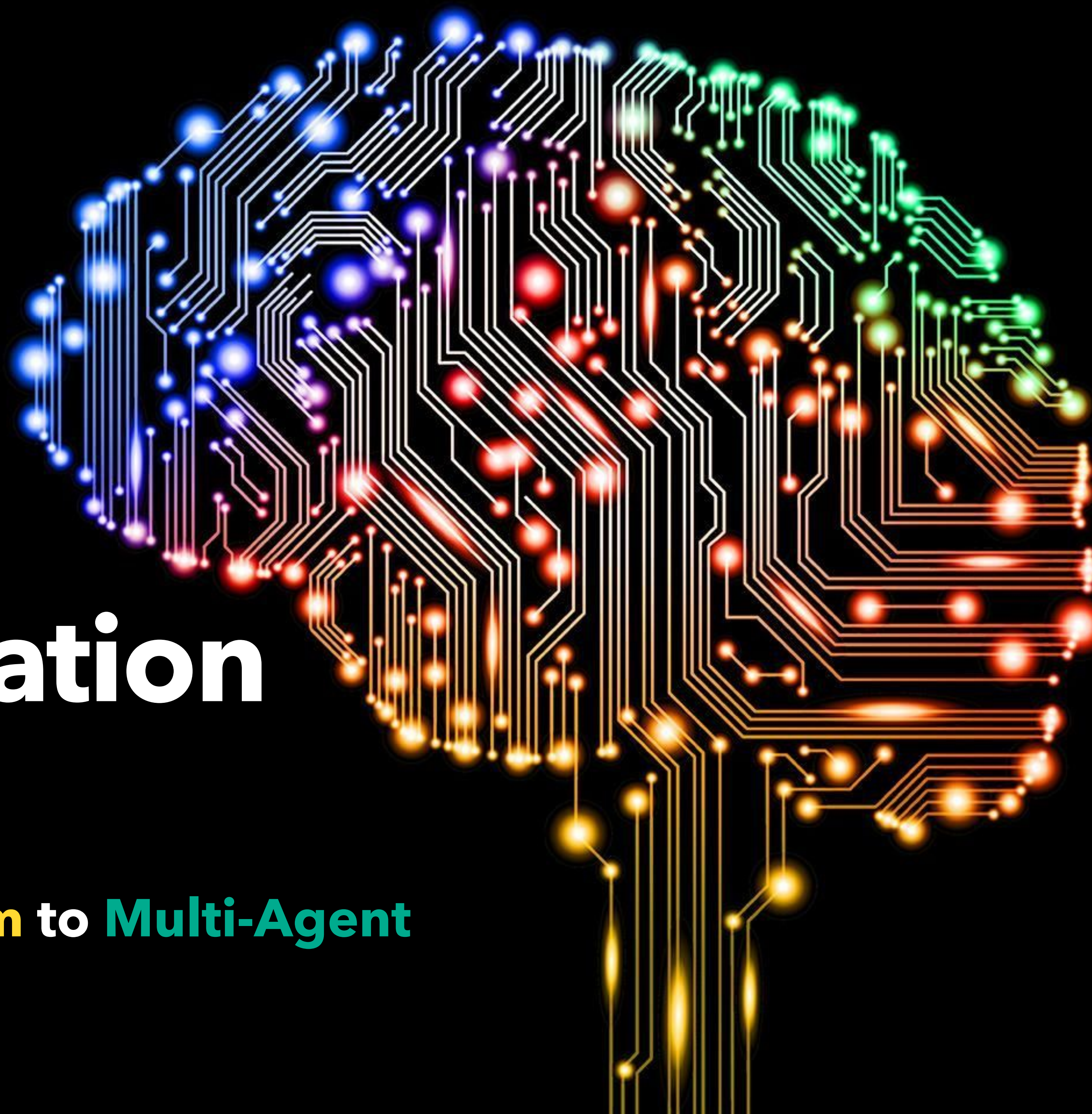


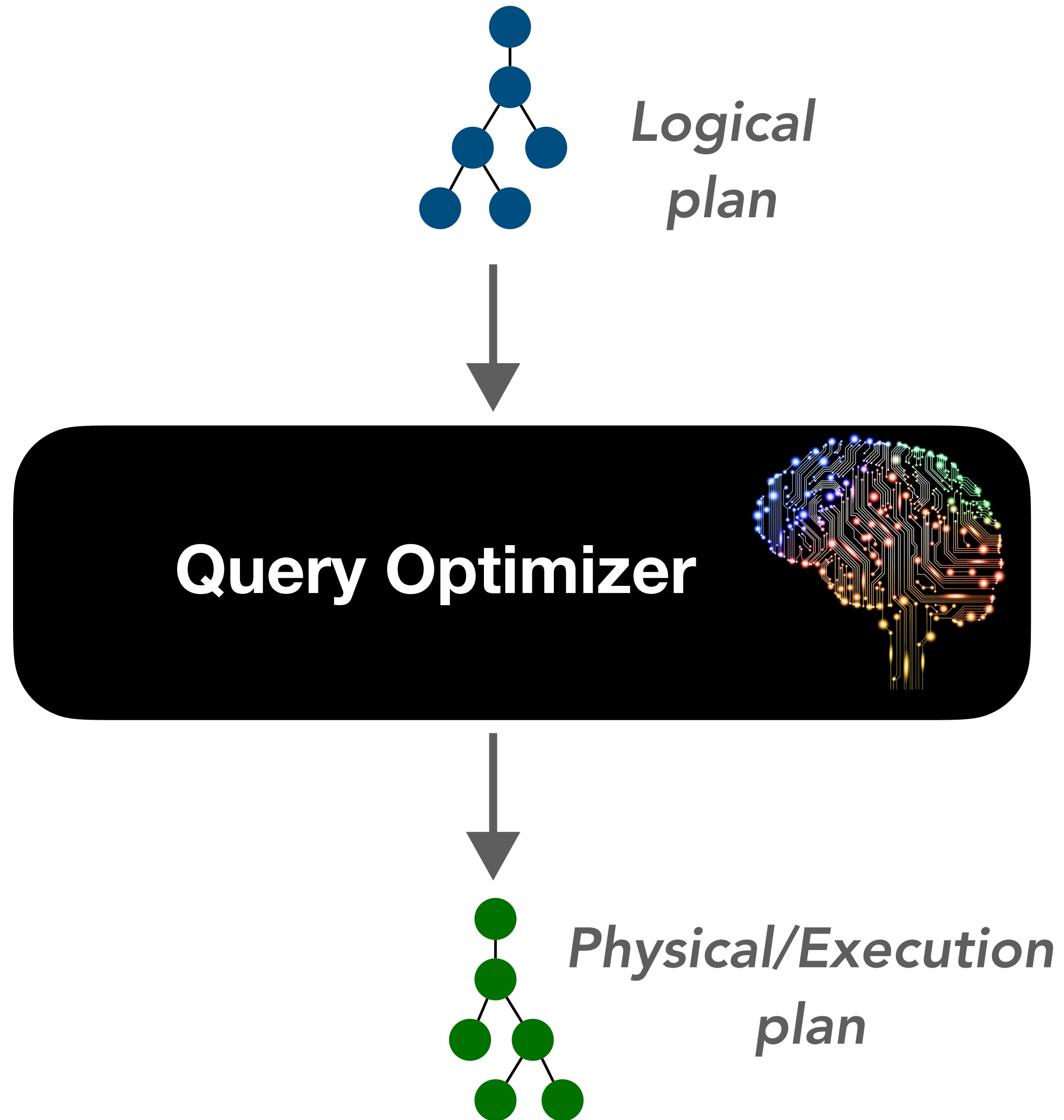
Zoi Kaoudi - ADBIS 2026

Query Optimization In The Wild

From **Relational** to **Cross-platform** to **Multi-Agent**
Data Systems



Query Optimizer: the brain that makes choices



From rule-based to cost-based query optimizers

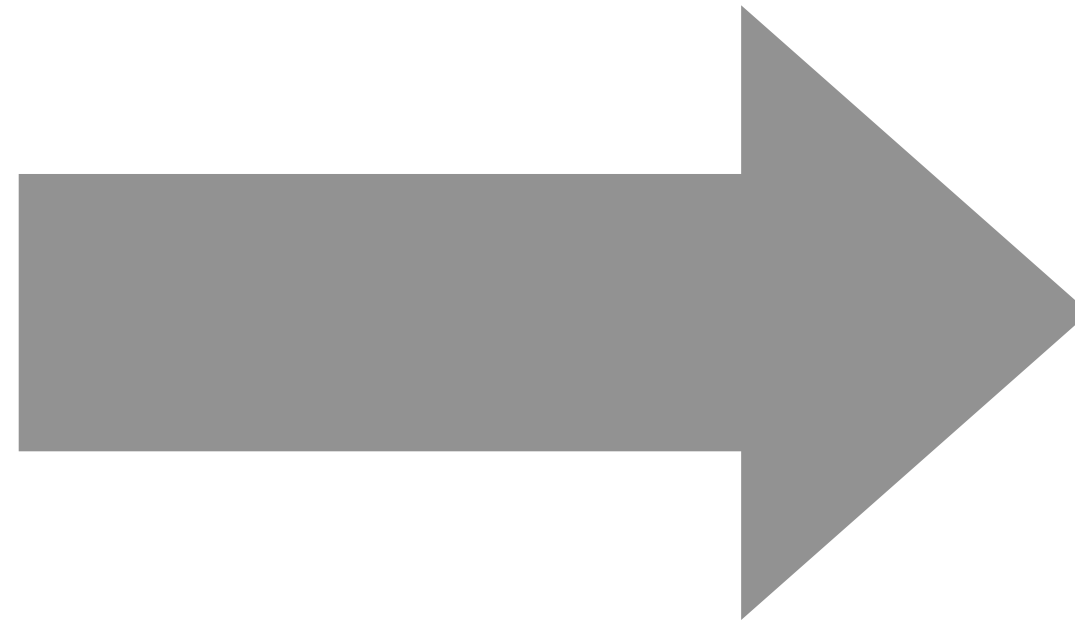
Relational



Rules

IF filter is on R,
push filter before join

IF R is smaller than S,
join R first



Cost

$$Cost_{nested_loop_join} = Cost_{outer} + N \times Cost_{inner}$$

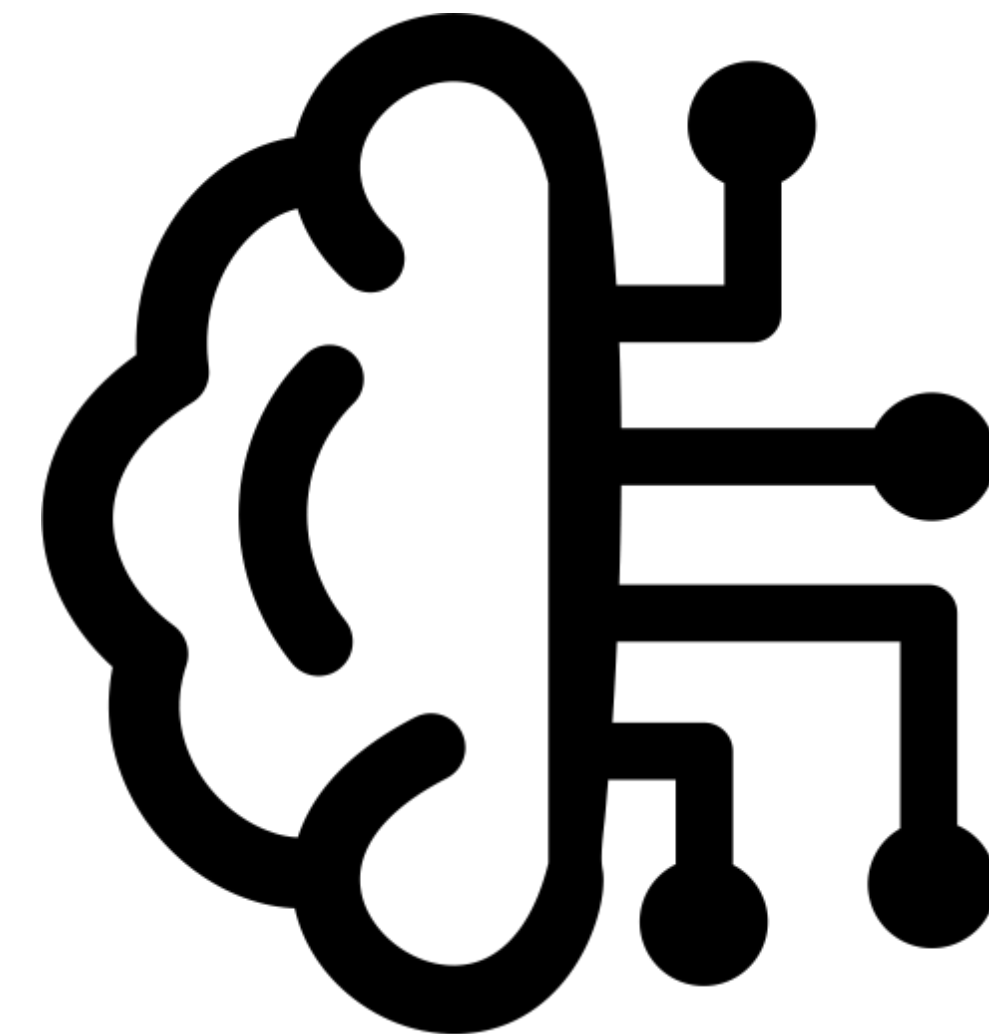
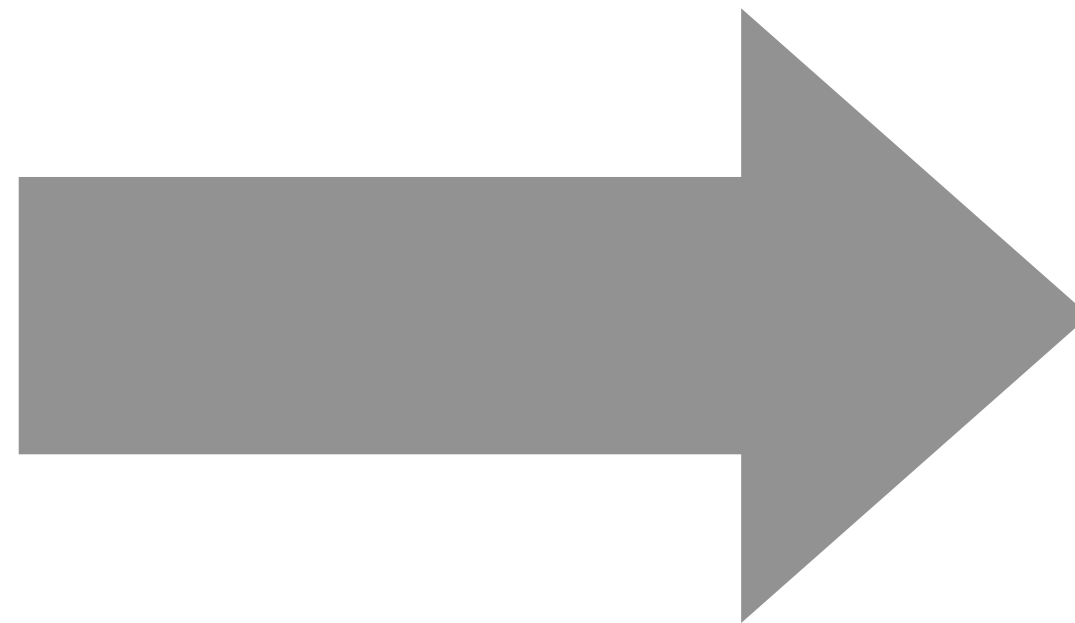
$$Cost_{outer} = Cost_{IO} + Cost_{cpu}$$

$$Cost_{IO} = \#pages \times Cost_{page_fetch}$$

ML opportunity



Cost



ML

Relational

$$Cost_{nested_loop_join} = Cost_{outer} + N \times Cost_{inner}$$

$$Cost_{outer} = Cost_{IO} + Cost_{cpu}$$

$$Cost_{IO} = \#pages \times Cost_{page_fetch}$$

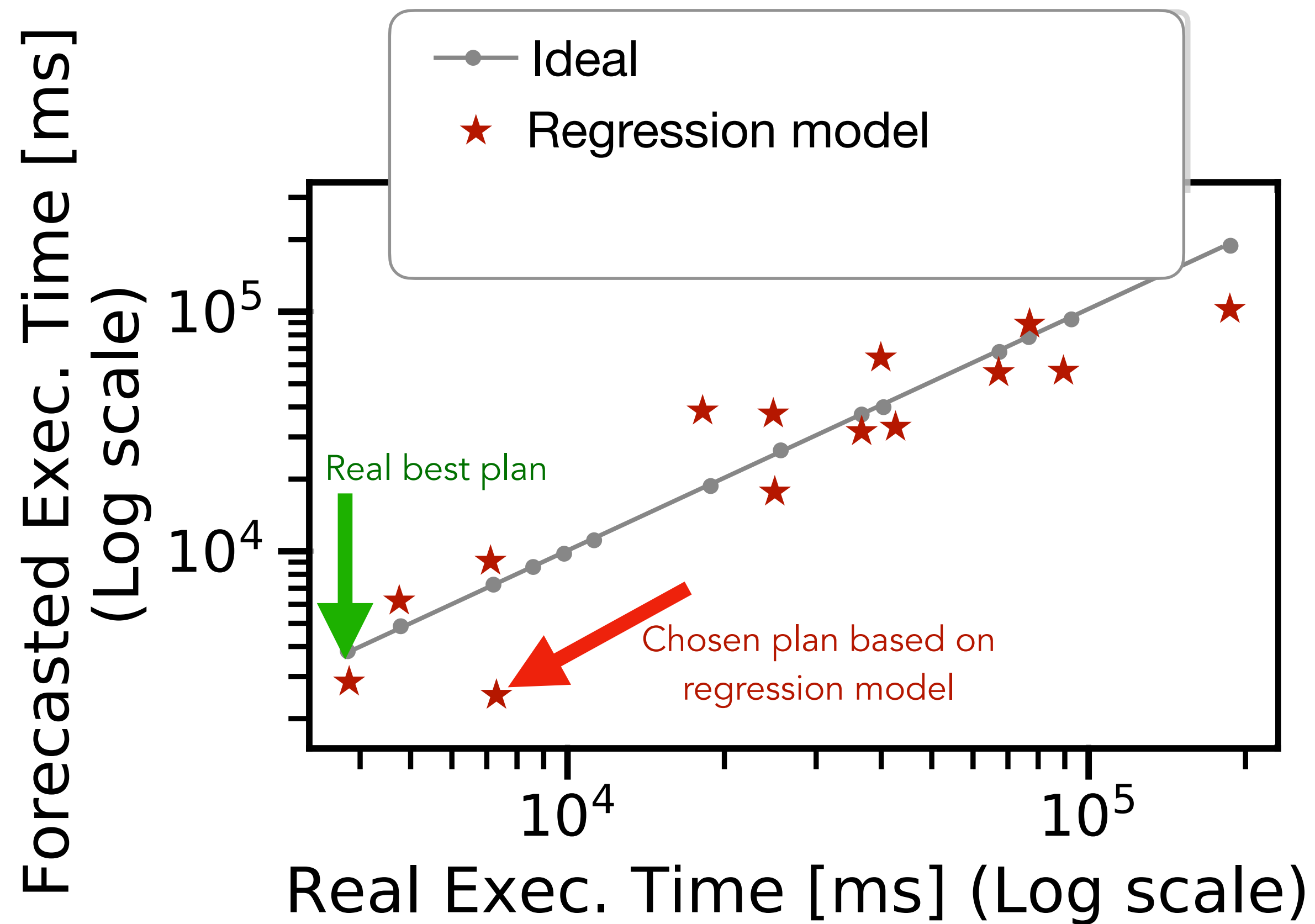
`Predicted_runtime(plan1) = 3sec`

**Latency regression:
hard to get
completely right**



Effect of regression models errors

Relational

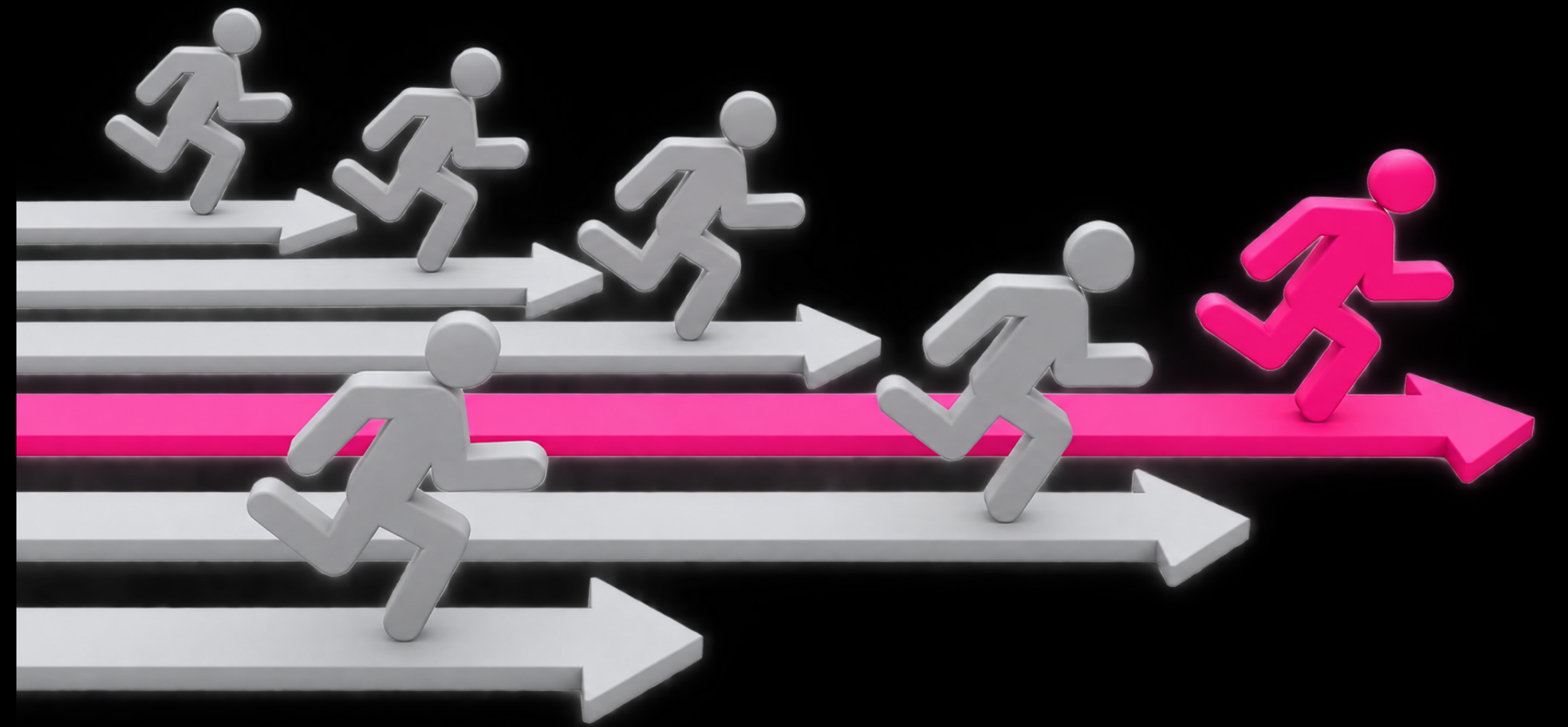


~ 2x worse execution time

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

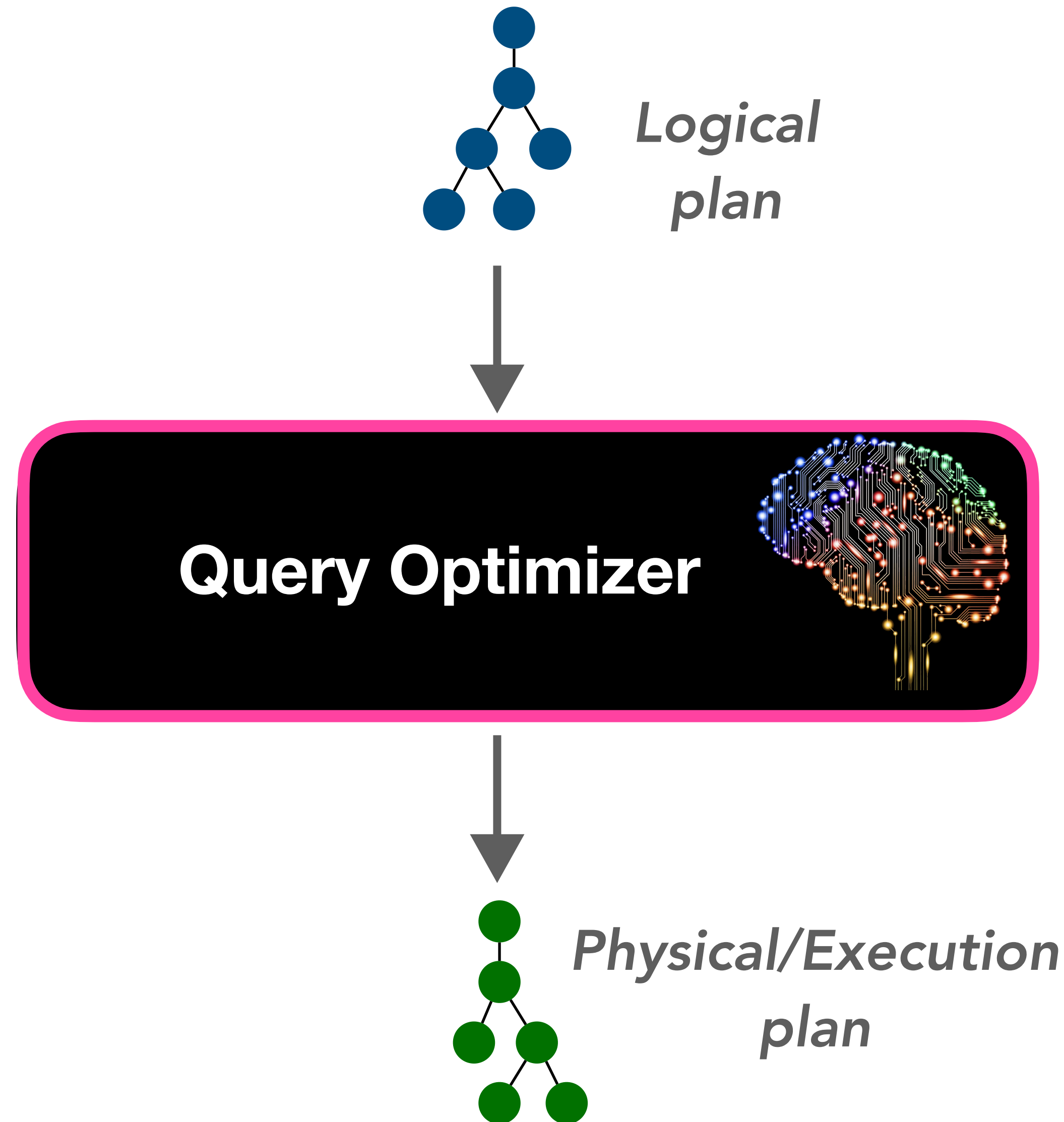


How can we
leverage
learning-to-rank
models in QO?



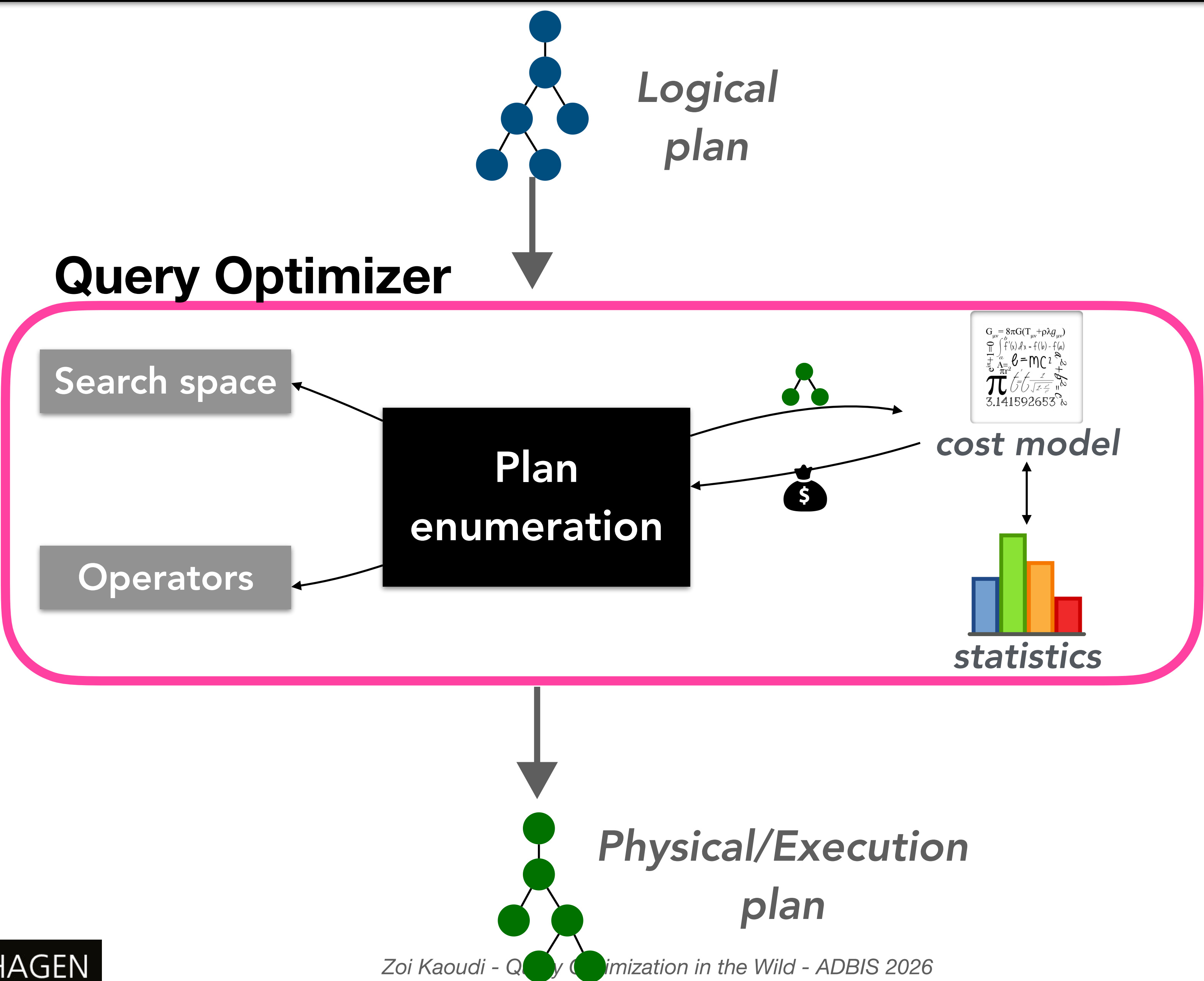
From traditional QO to learning-to-rank (LTR) QO

Relational



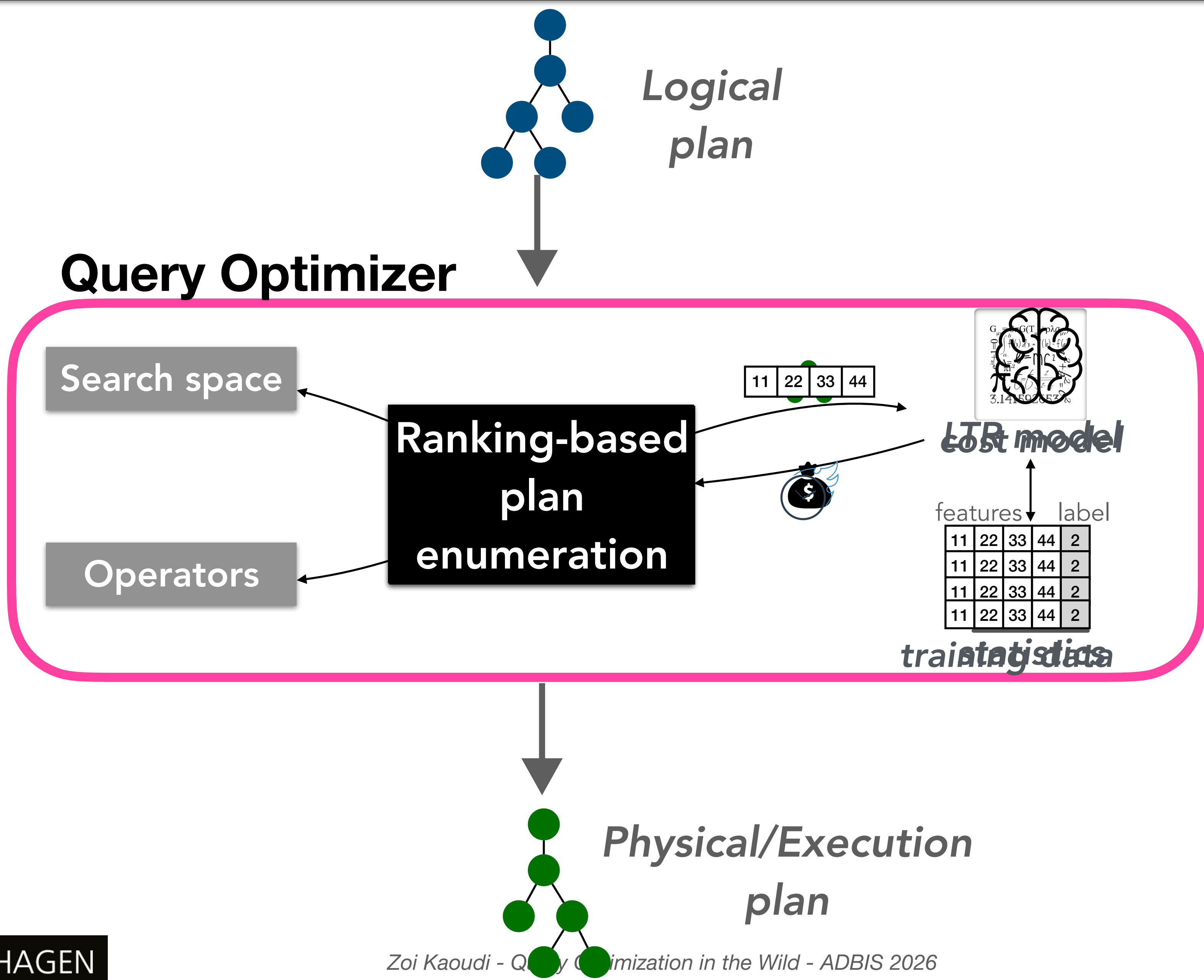
From traditional QO to learning-to-rank (LTR) QO

Relational



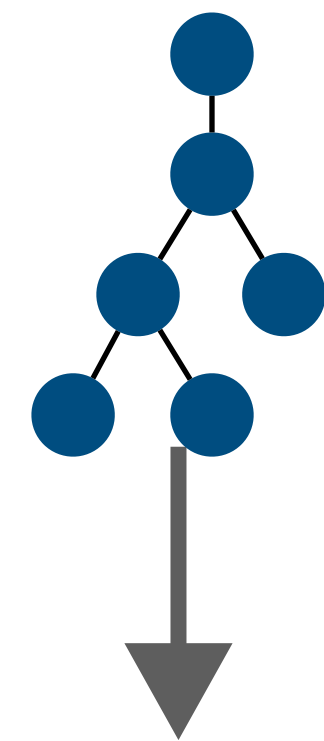
From traditional QO to learning-to-rank (LTR) QO

Relational



From traditional QO to learning-to-rank (LTR) QO

Relational



Logical
plan

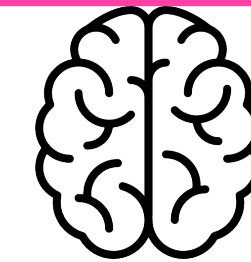
Query Optimizer

Search space

Operators

Ranking-based
plan
enumeration

11 22 33 44



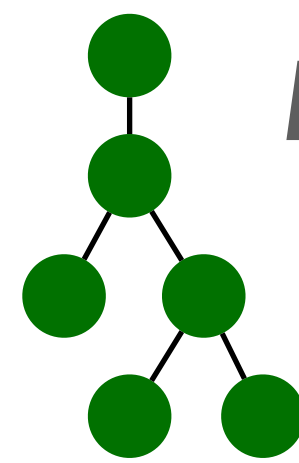
LTR model



features label

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

training data



Physical/Execution
plan

Questions

Type of LTR

Plan enumeration

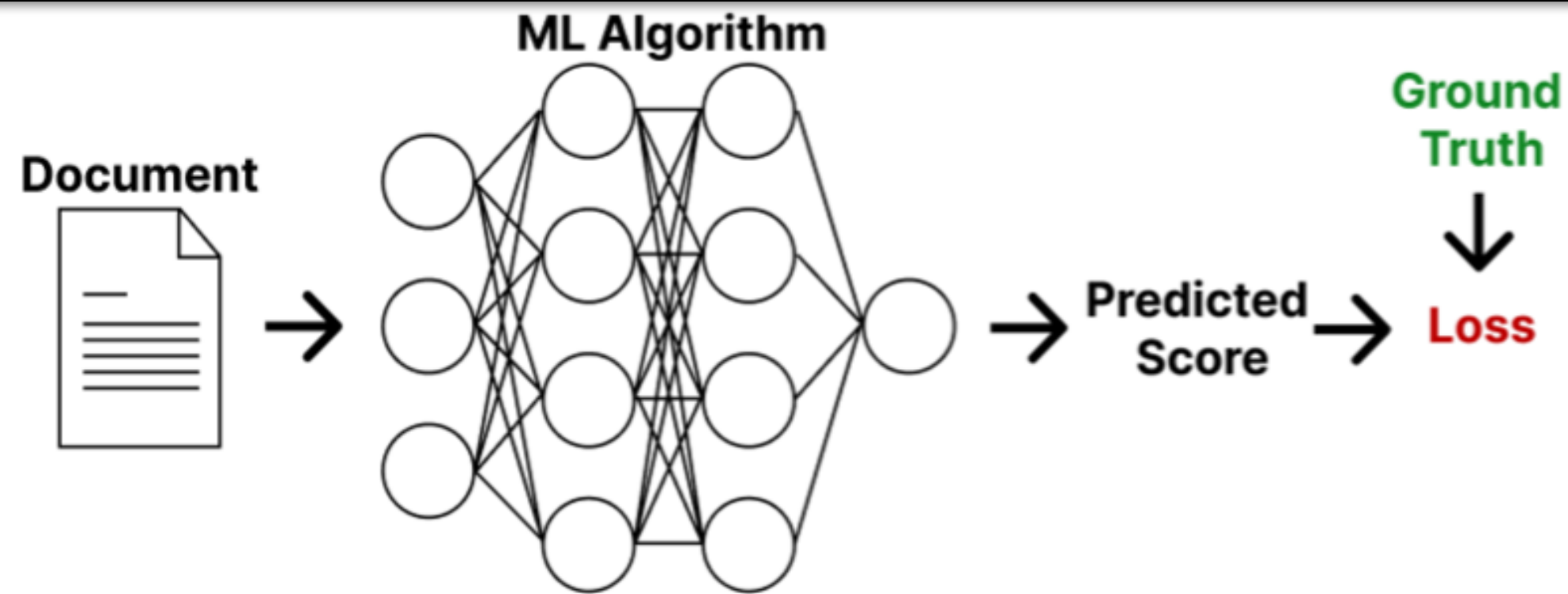
Model architecture

Training data

Learning-to-rank approach

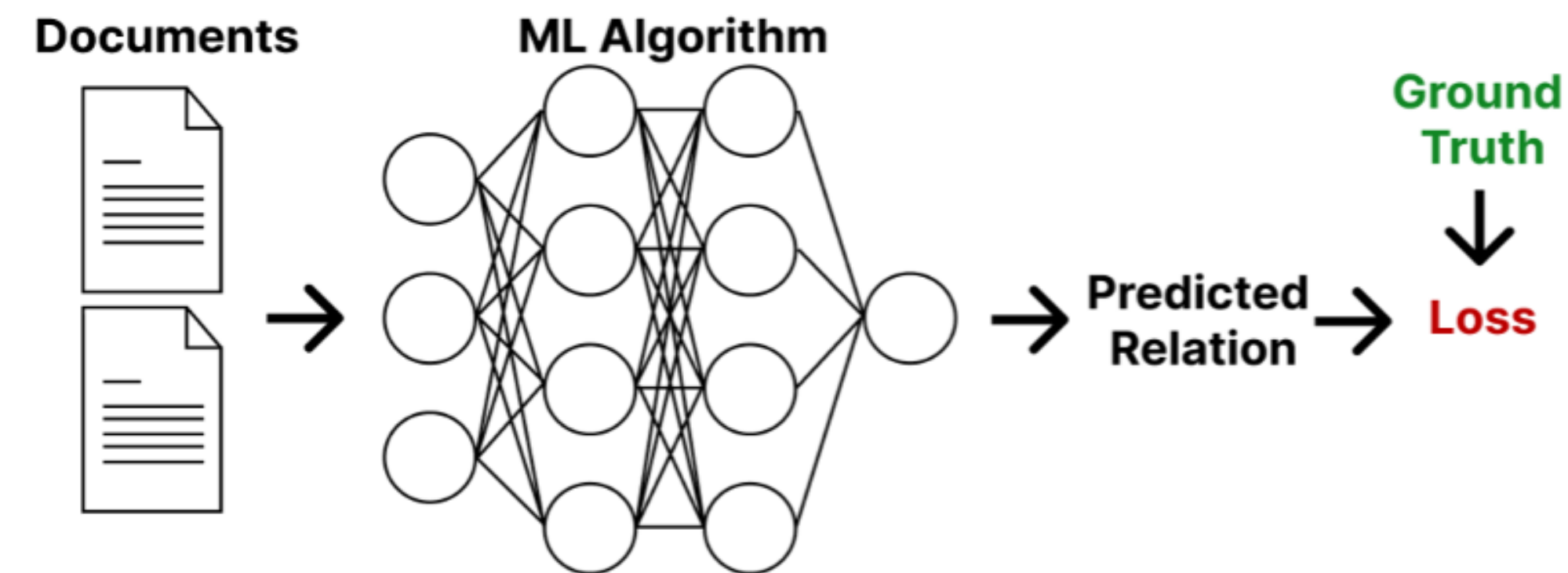
Relational

Pointwise



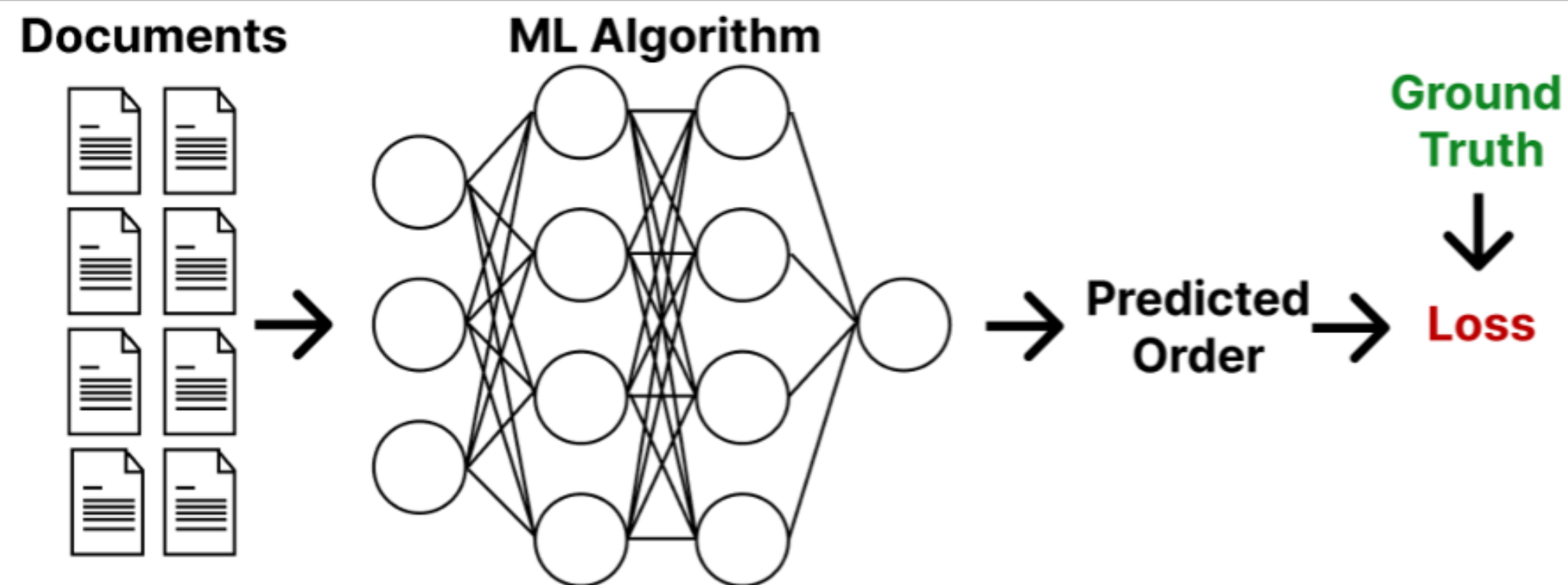
No comparison among items

Pairwise



Expensive training
Assumes pairs are i.i.d

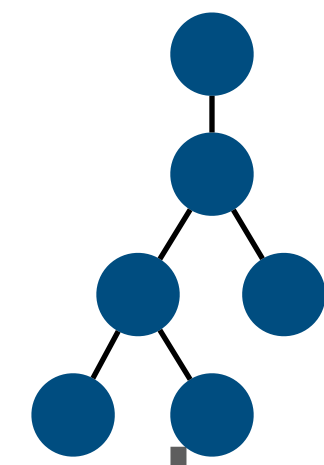
Listwise



Emphasises the ranking objective

From traditional QO to learning-to-rank QO

Relational



Logical
plan

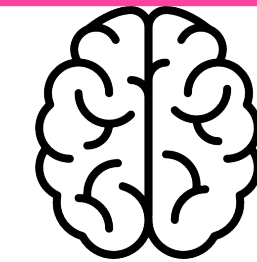
Query Optimizer

Search space

Operators

Ranking-based
plan
enumeration

11 22 33 44

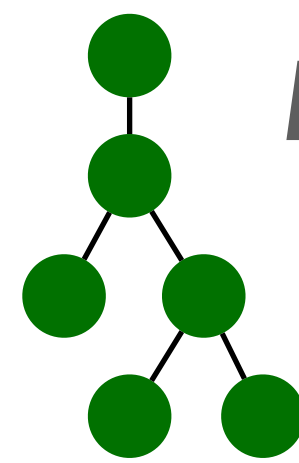


LTR model

features label

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

training data



Physical/Execution
plan

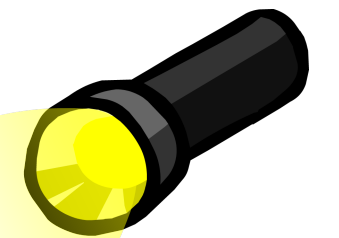
◆ Questions

◆ Type of LTR

◆ Plan enumeration

◆ Model architecture

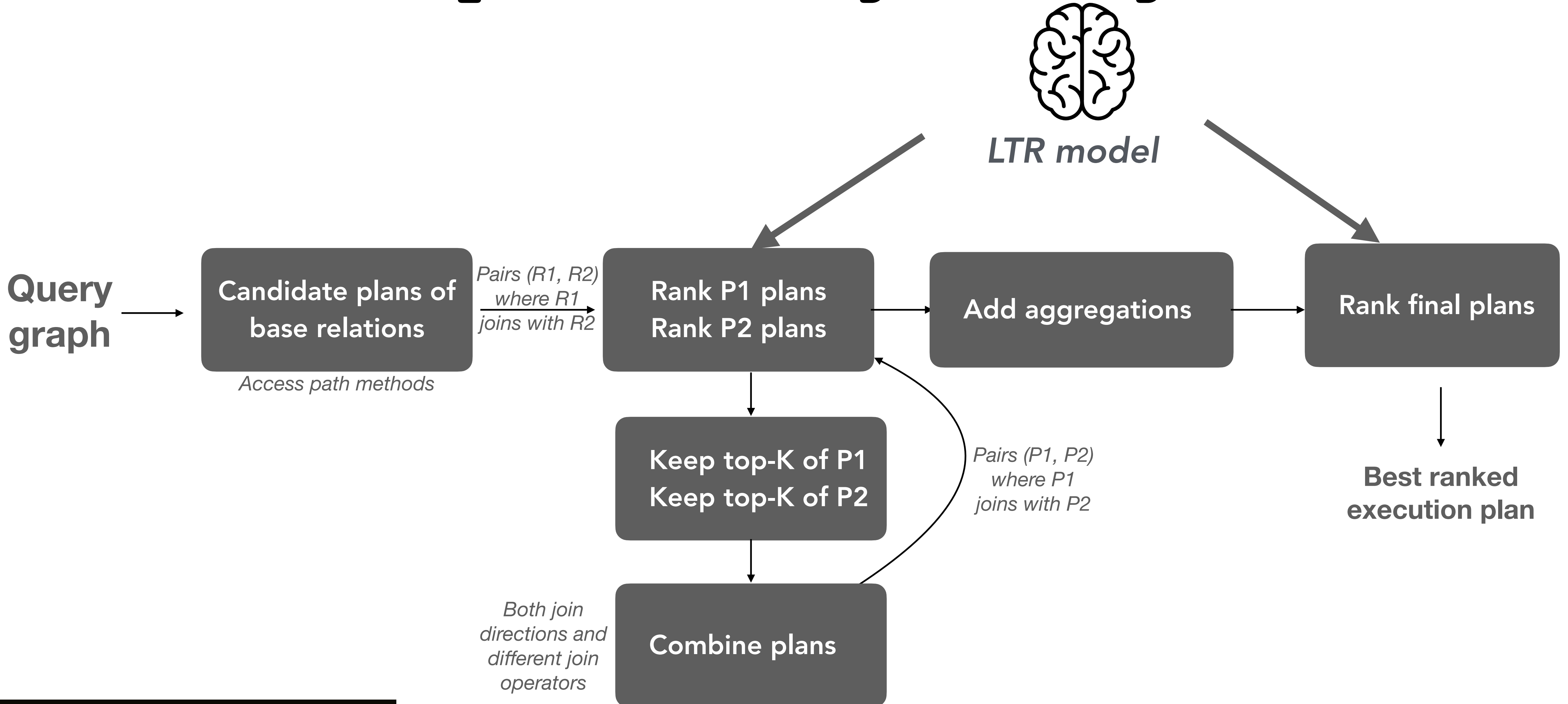
◆ Training data



Ranking-based plan enumerator

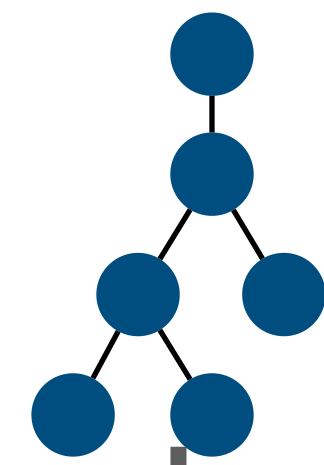
LTR-based Dynamic Programming

Relational



From traditional QO to learning-to-rank QO

Relational



Logical
plan

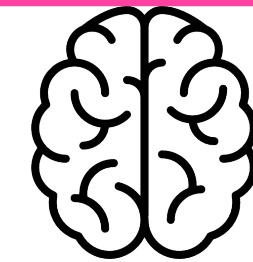
Query Optimizer

Search space

Operators

Ranking-based
plan
enumeration

11 22 33 44

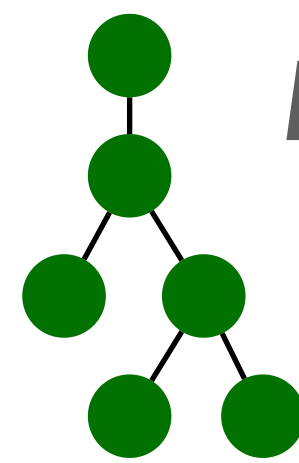


LTR model

features label

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

training data



Physical/Execution
plan

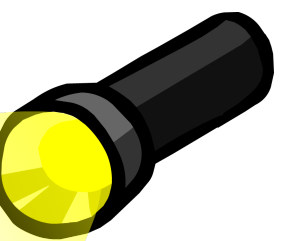
Questions

◆ Type of LTR

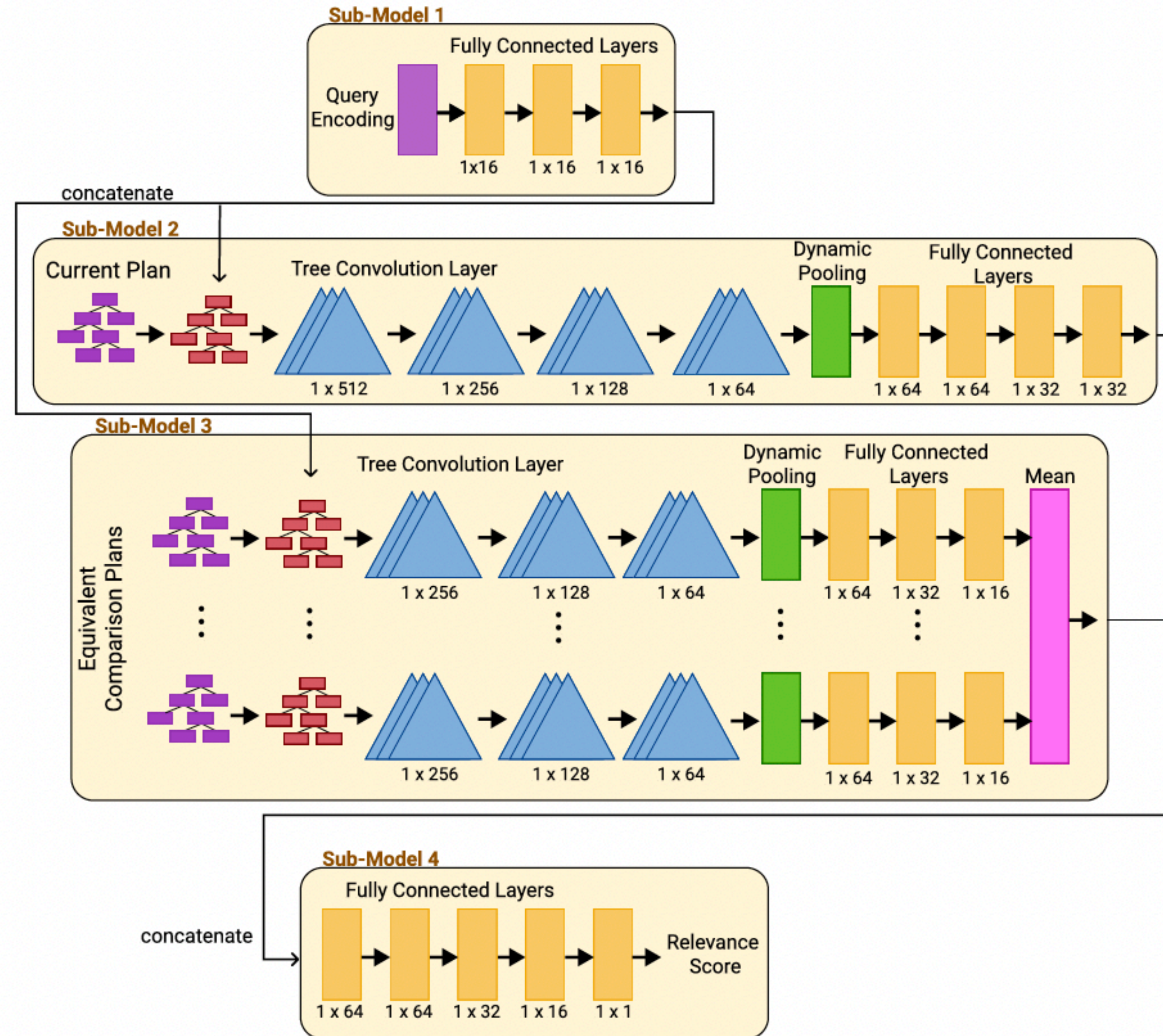
◆ Plan enumeration

◆ Model architecture

◆ Training data



LTR model architecture



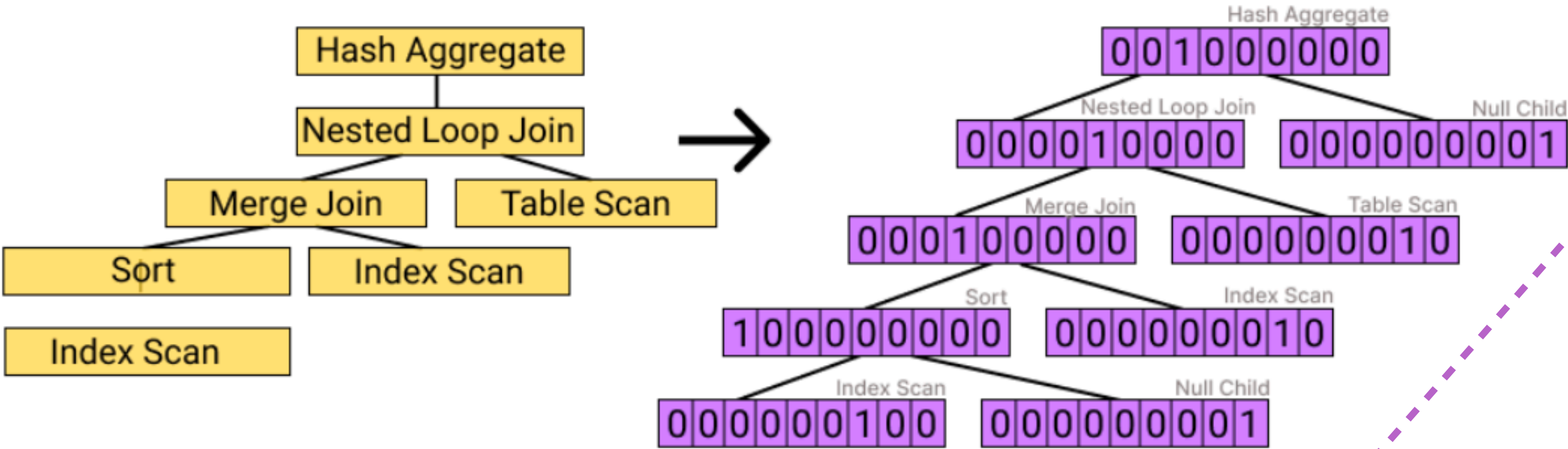
Featurization

Query encoding (logical)

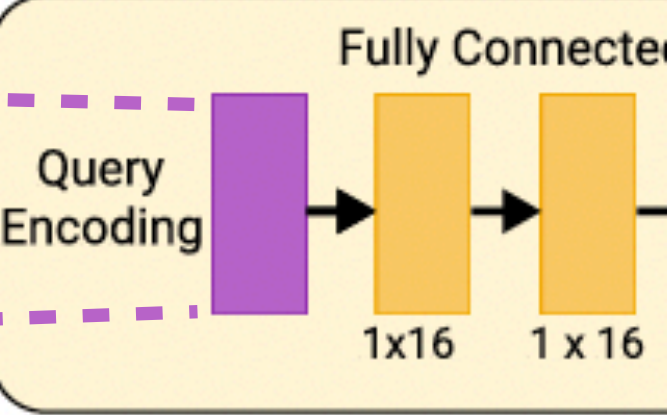
Has Order By	Has Group By	Number of Joins	Estimated Number of Rows	Max. Relation	Min. Relation
--------------	--------------	-----------------	--------------------------	---------------	---------------

Plan encoding (physical)

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

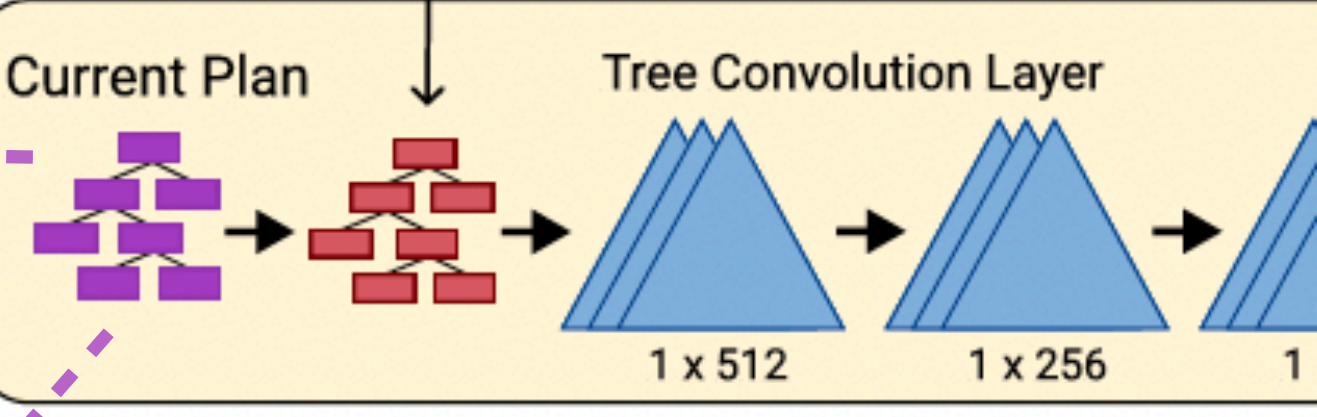


Sub-Model 1

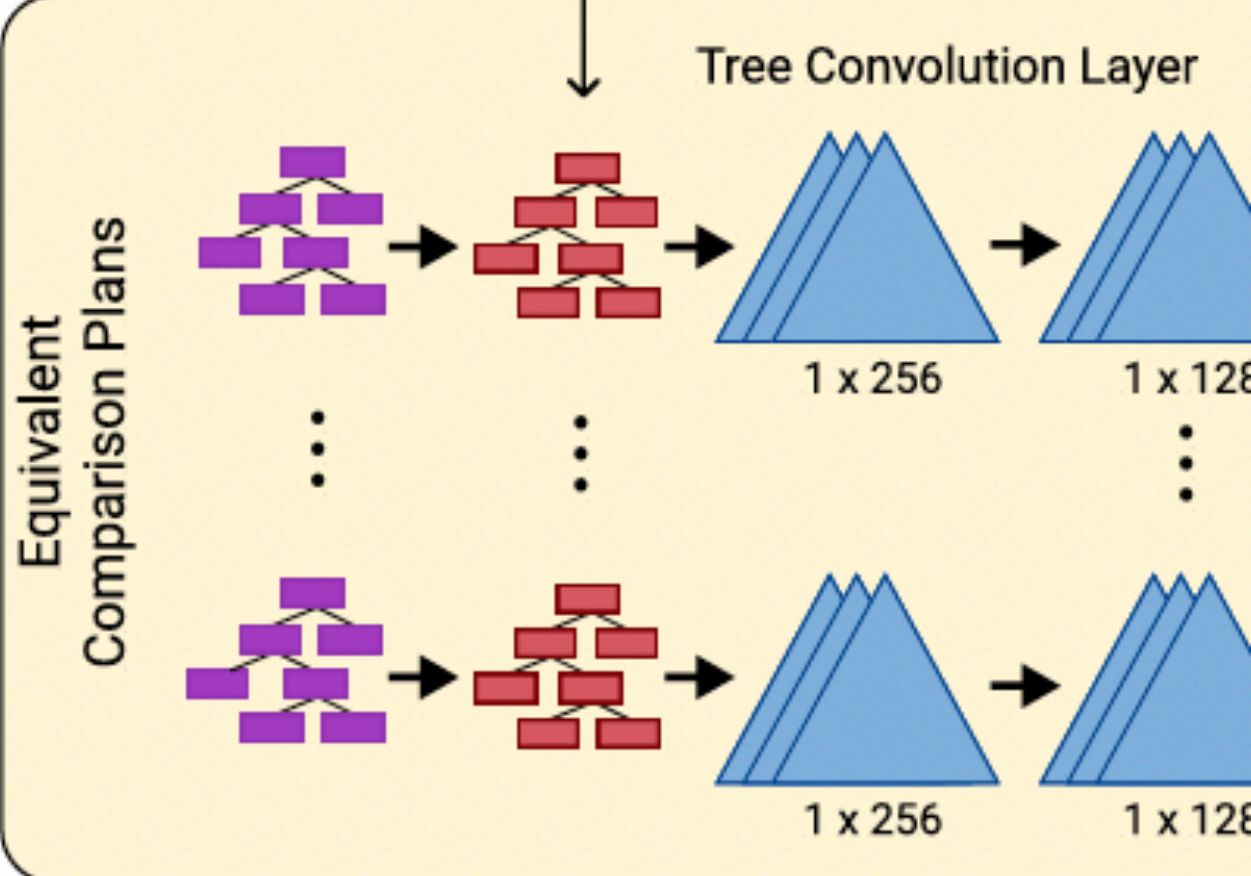


concatenate

Sub-Model 2

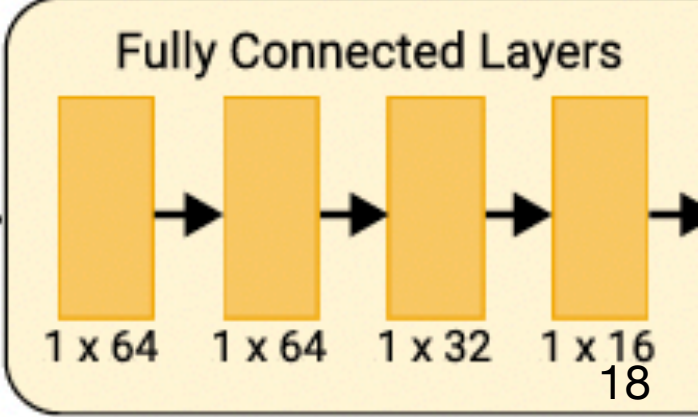


Sub-Model 3



concatenate

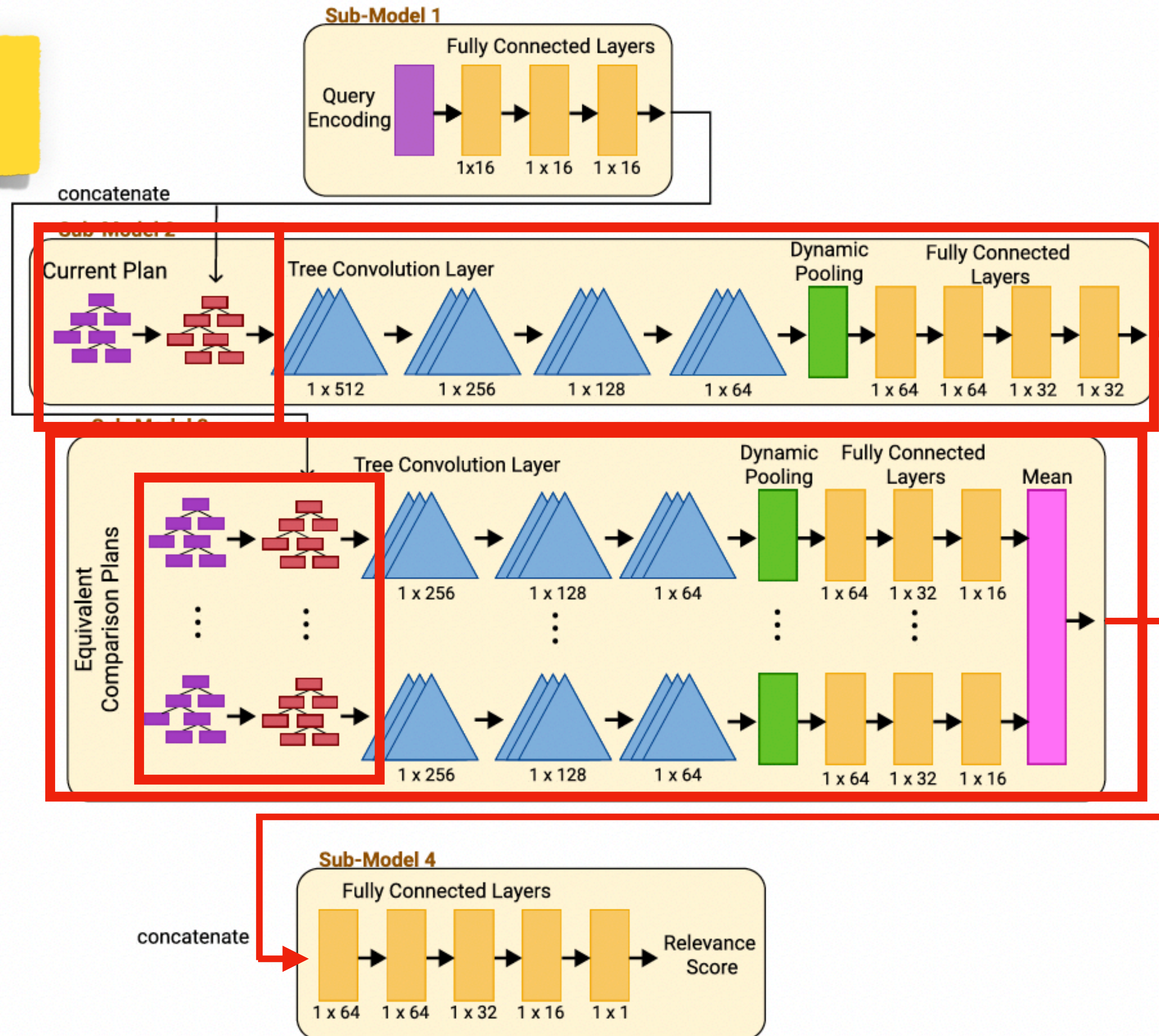
Sub-Model 4



LTR model architecture

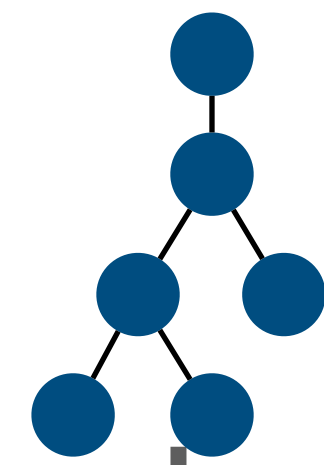
Relational

Each plan against the rest equivalent plans



From traditional QO to learning-to-rank QO

Relational



Logical
plan

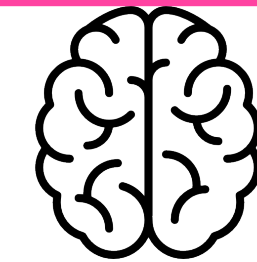
Query Optimizer

Search space

Operators

Ranking-based
plan
enumeration

11 22 33 44

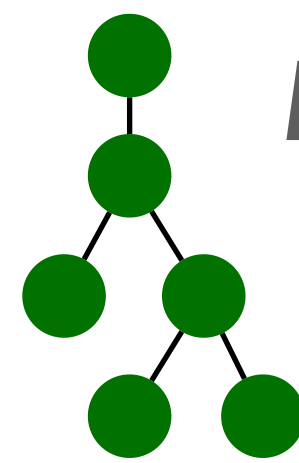


LTR model

features label

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

training data



Physical/Execution
plan

Questions

◆ Type of LTR

◆ Plan enumeration

◆ Model architecture

◆ Training data

Training data collection challenges

Relational

Pre-optimized plans lead to
biased training samples

Prohibitive labeling cost:
executing (bad) plans may take months

features				label
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2

training data

Diverse subplan generation

Relational

features label

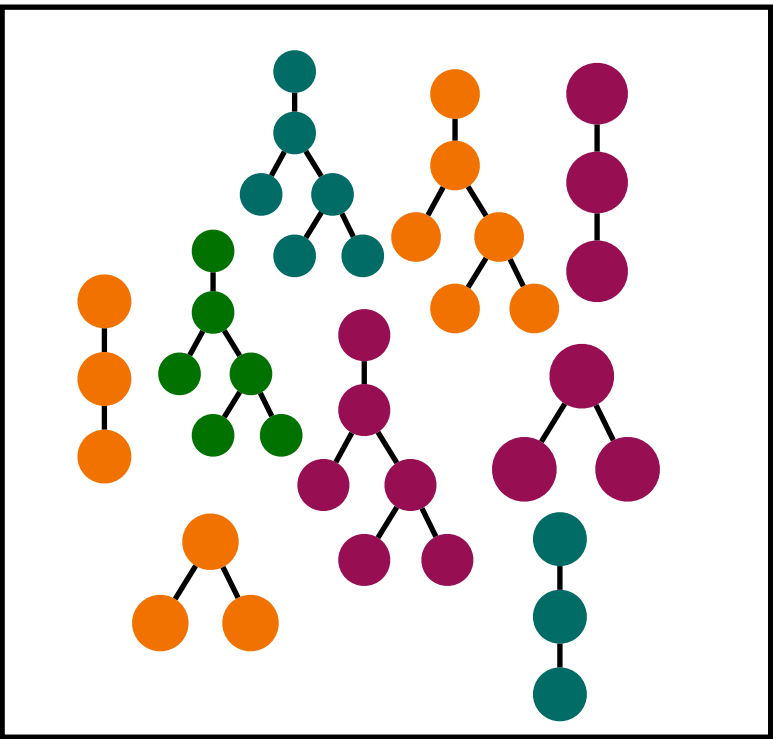
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2
11	22	33	44	2

training data

```
SELECT FROM ..  
      SELECT FROM ..  
SELECT FROM ..  
      SELECT FROM ..  
SELECT FROM ..  
      SELECT FROM ..  
SELECT FROM ..  
      SELECT FROM ..
```

Training queries

Random plans
→
using DP



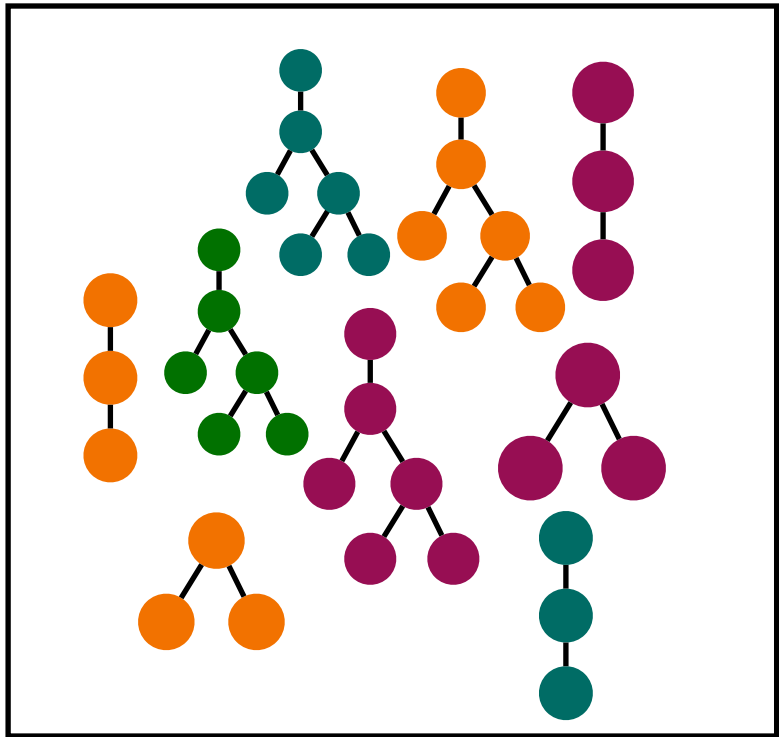
Diverse execution (sub)plans

Broad coverage of subplan sizes, join orders, operator choices

Rank-based labels based on cost estimation

features				label
11	22	33	44	2
11	22	33	44	3
11	22	33	44	5
11	22	33	44	1

training data



Diverse execution (sub)plans

$$\begin{aligned} G_{\mu\nu} &= 8\pi G(T_{\mu\nu} + \rho\lambda g_{\mu\nu}) \\ \int_a^b f'(x) dx &= f(b) - f(a) \\ \Lambda_{-2} &= \frac{1}{\pi^{1/2}} \\ \pi &= 3.141592653\dots \\ \ell &= mc^2 \\ \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} &= \gamma \end{aligned}$$

cost model

	Cost	Label	Relational
Subquery 1	100	1	
	530	2	
	1146	3	
	...	...	
Subquery 2	1100	1	
	2900	2	
	9000	3	
	...	...	
Subquery 3	300	1	
	107	2	
	867	3	
	1798	4	

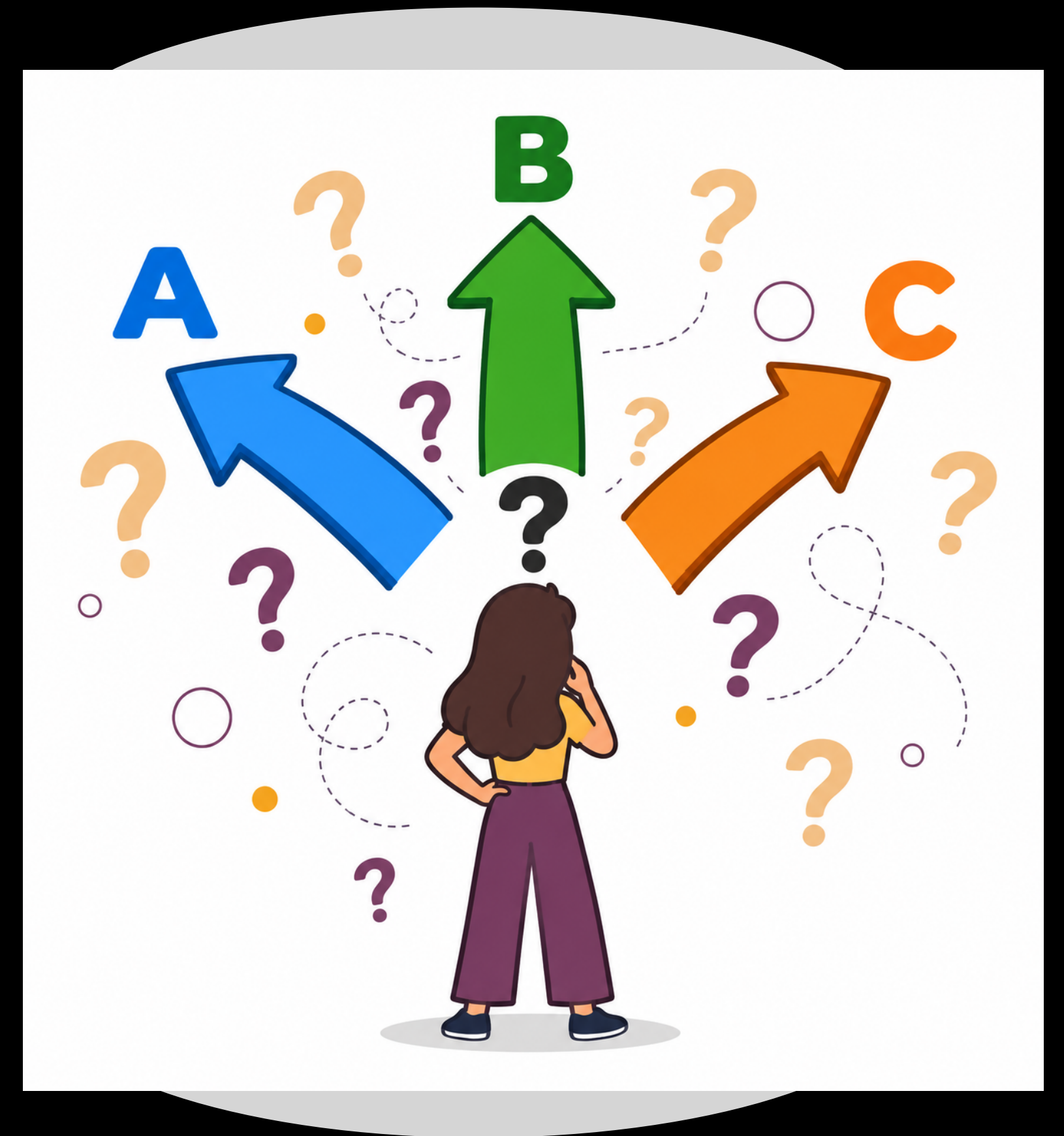
Fast training data

Estimation errors are mitigated due to randomness

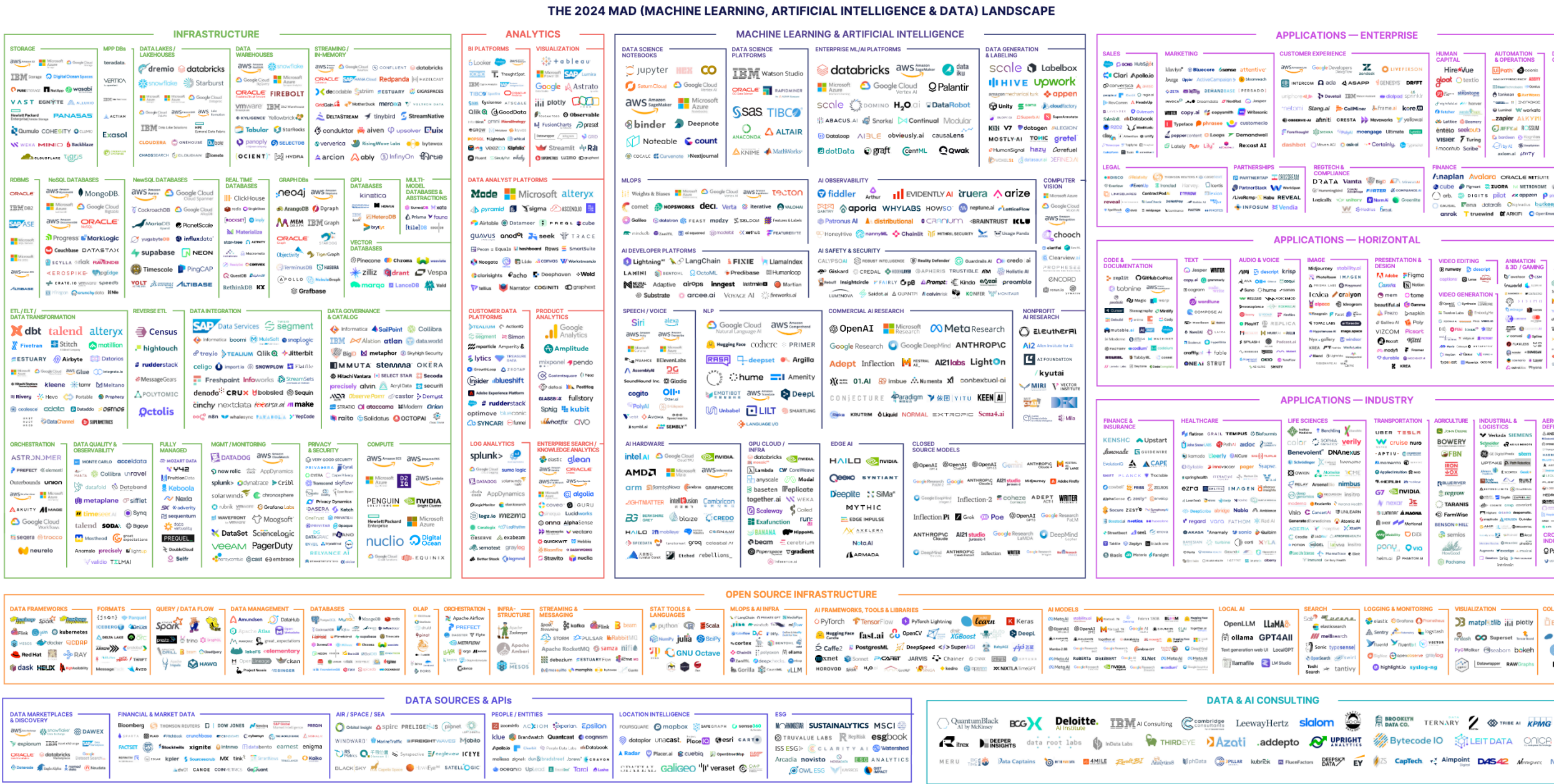
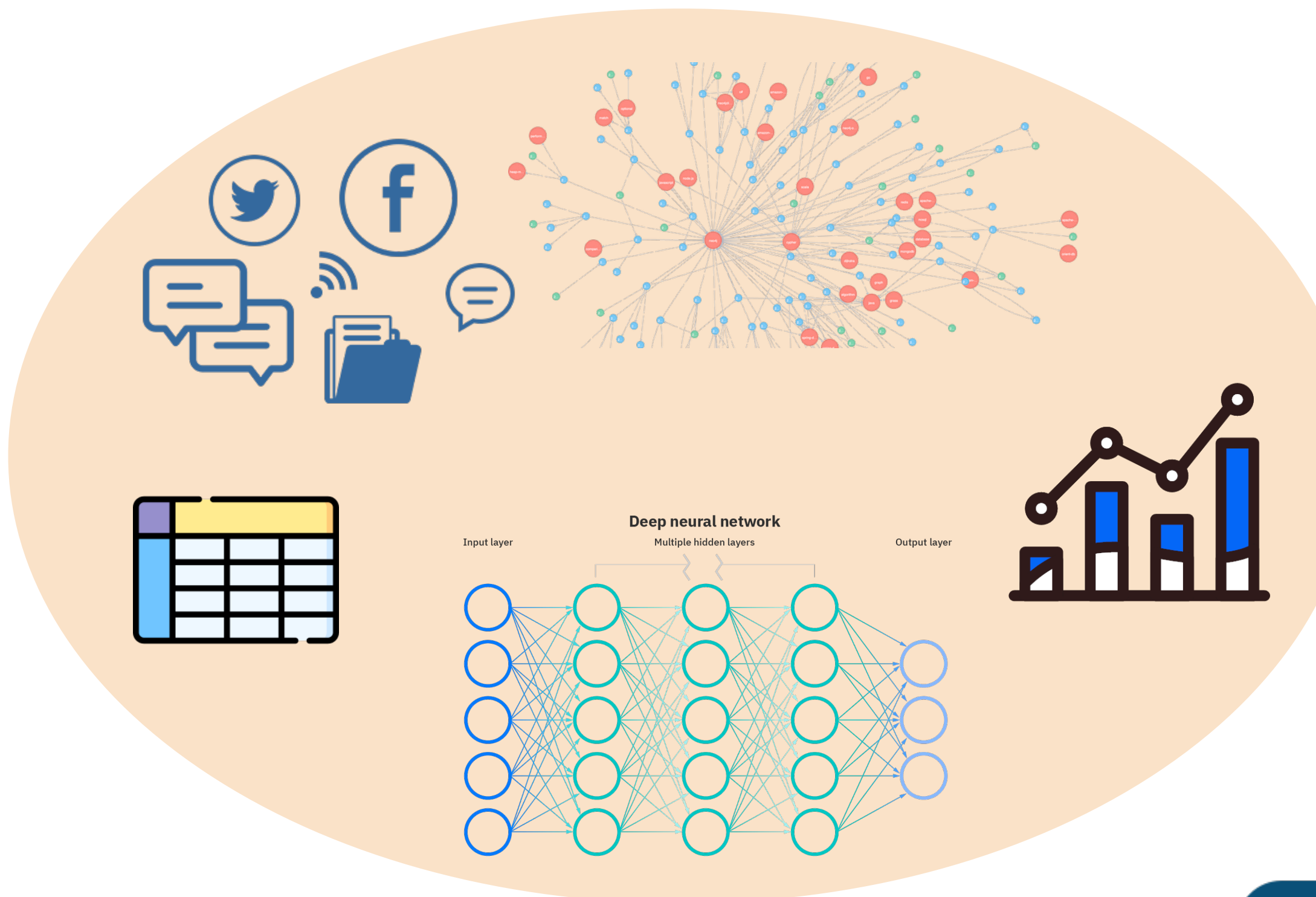
Query Optimization In The Wild

Relational

QO goes far beyond relational model



Data analytics landscape

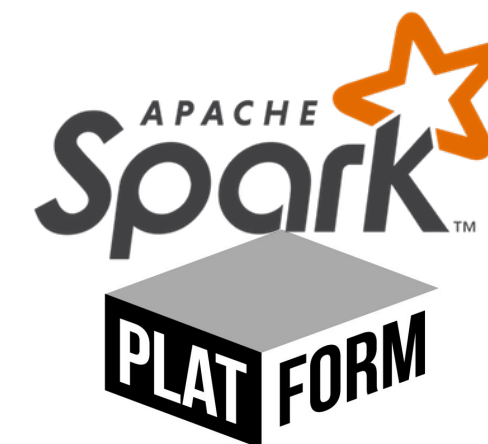
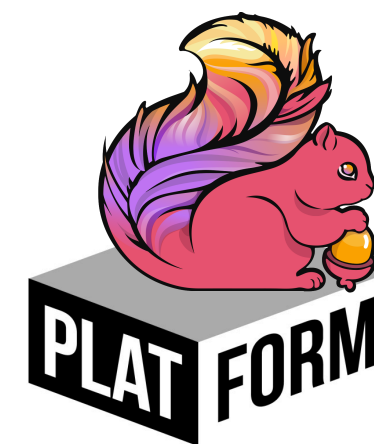
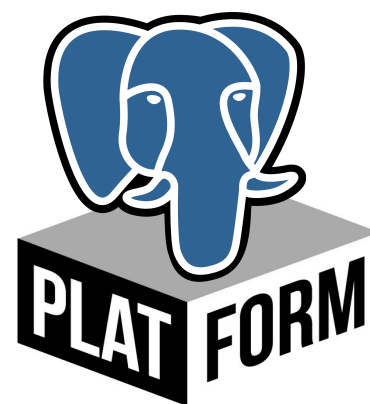
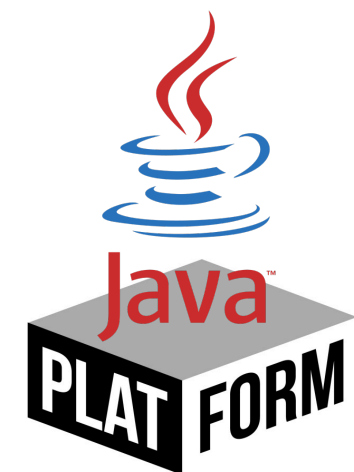
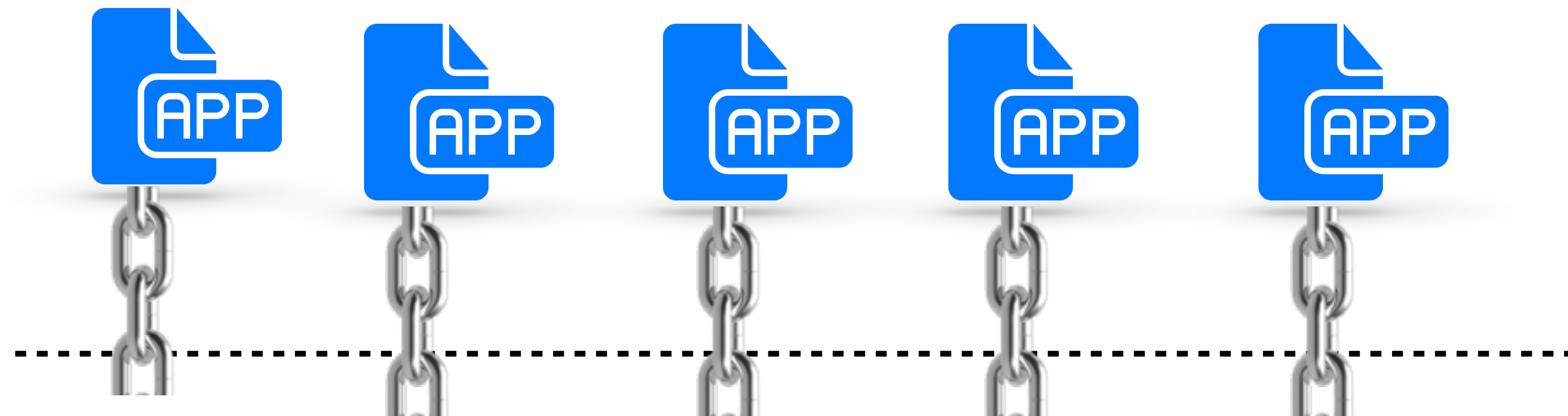


Complex analytics

Zoo of systems



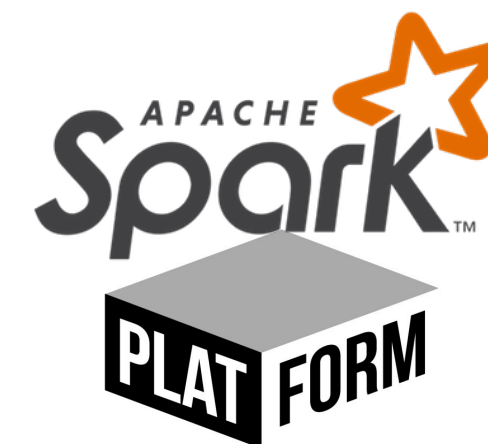
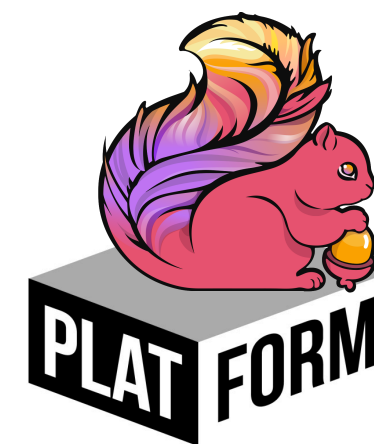
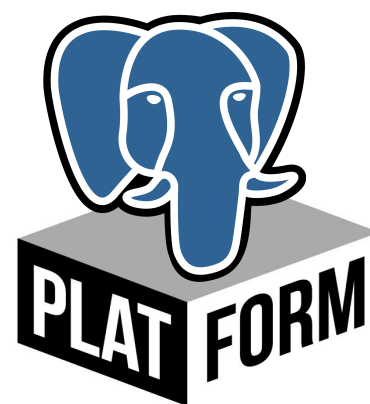
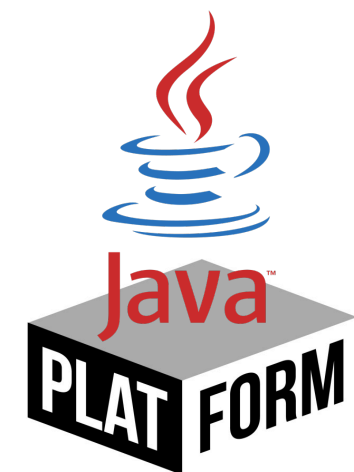
How can we help developers?



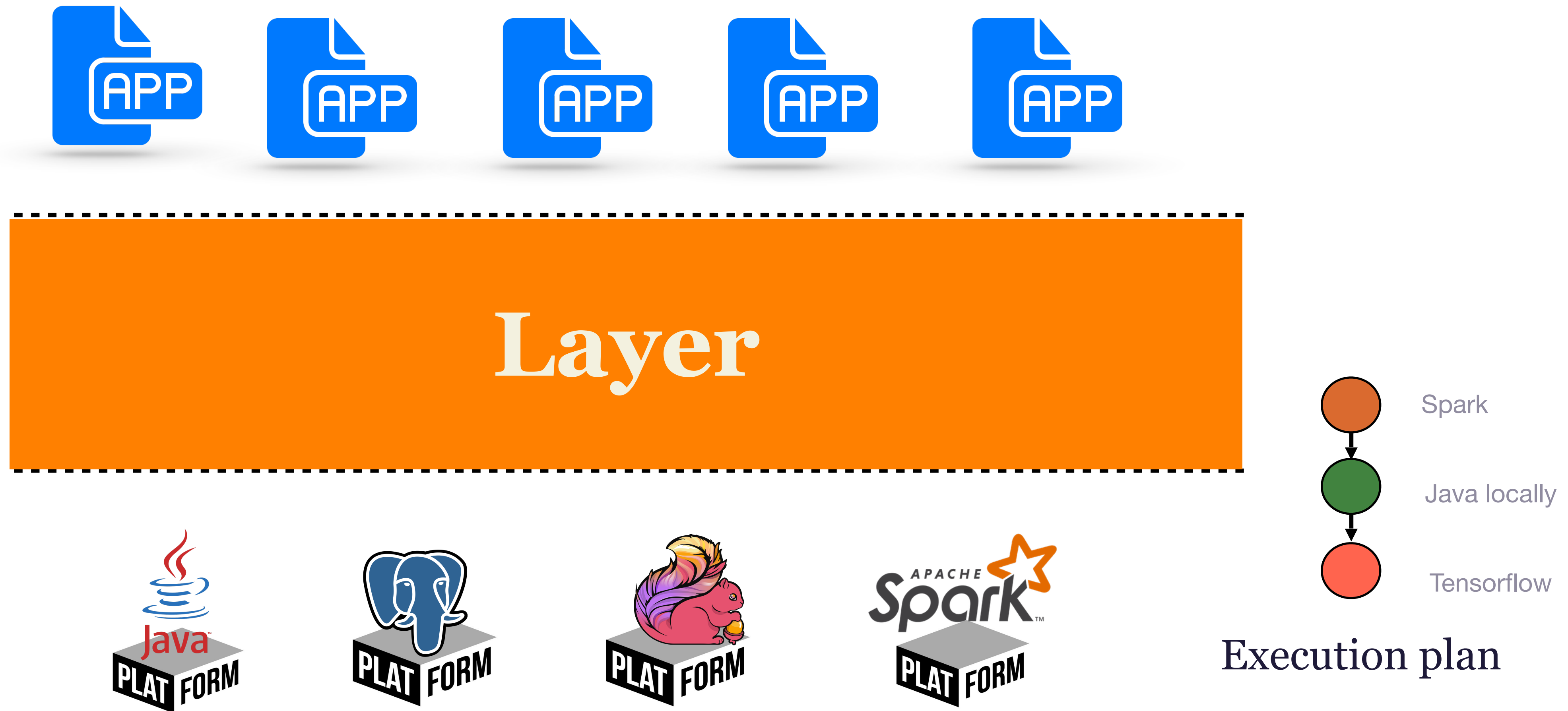
I. Decouple applications from underlying platforms



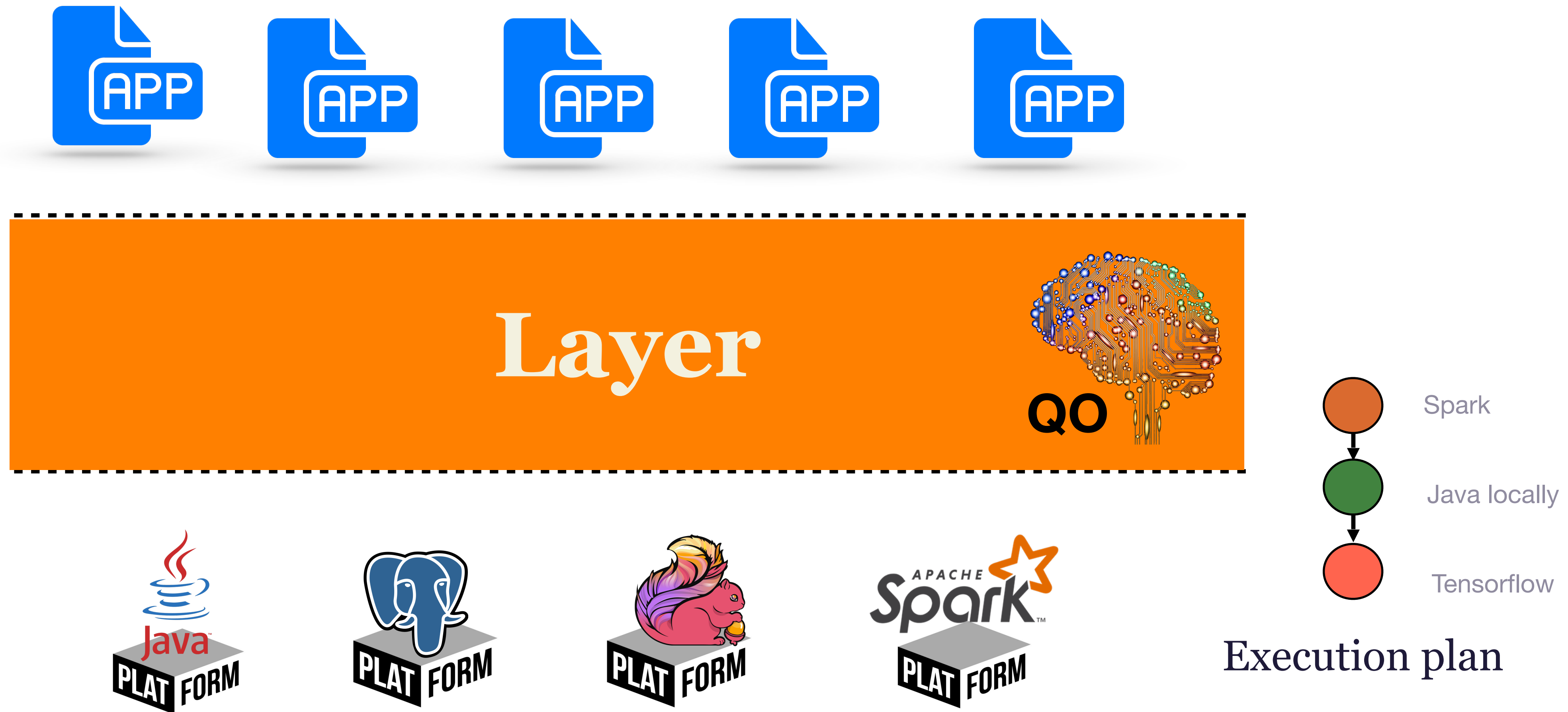
Layer



II. Allow for complex execution plans



III. Provide a query optimizer

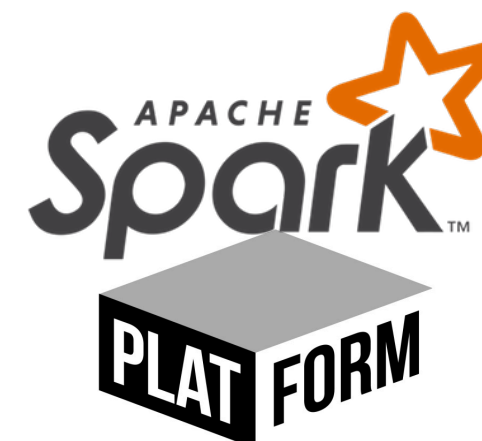
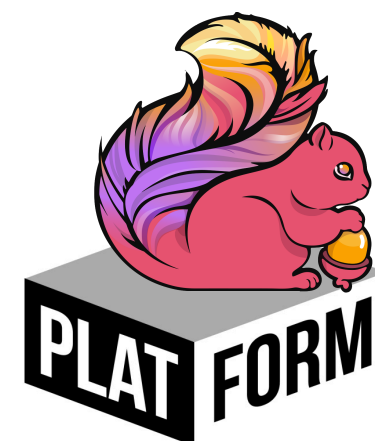
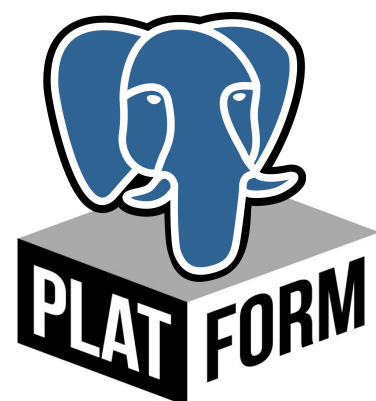
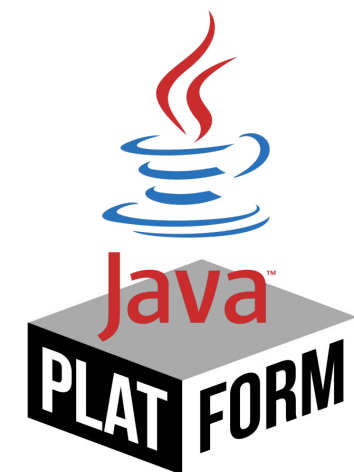


An open-source cross-platform system

Cross-platform



Apache Wayang



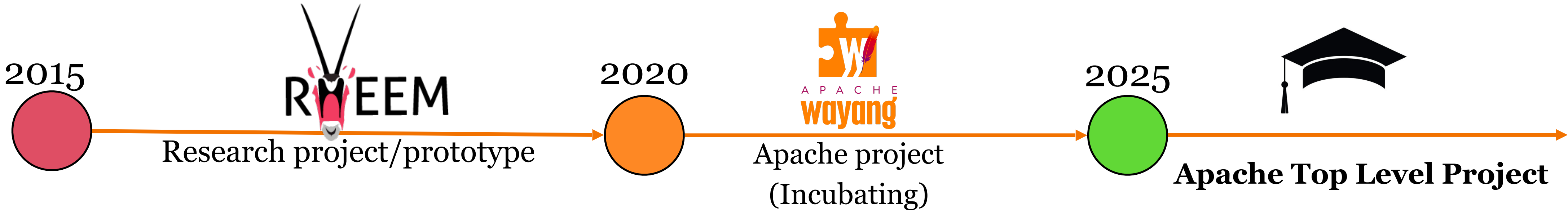
Write code once
Run anywhere

Cross-platform
execution

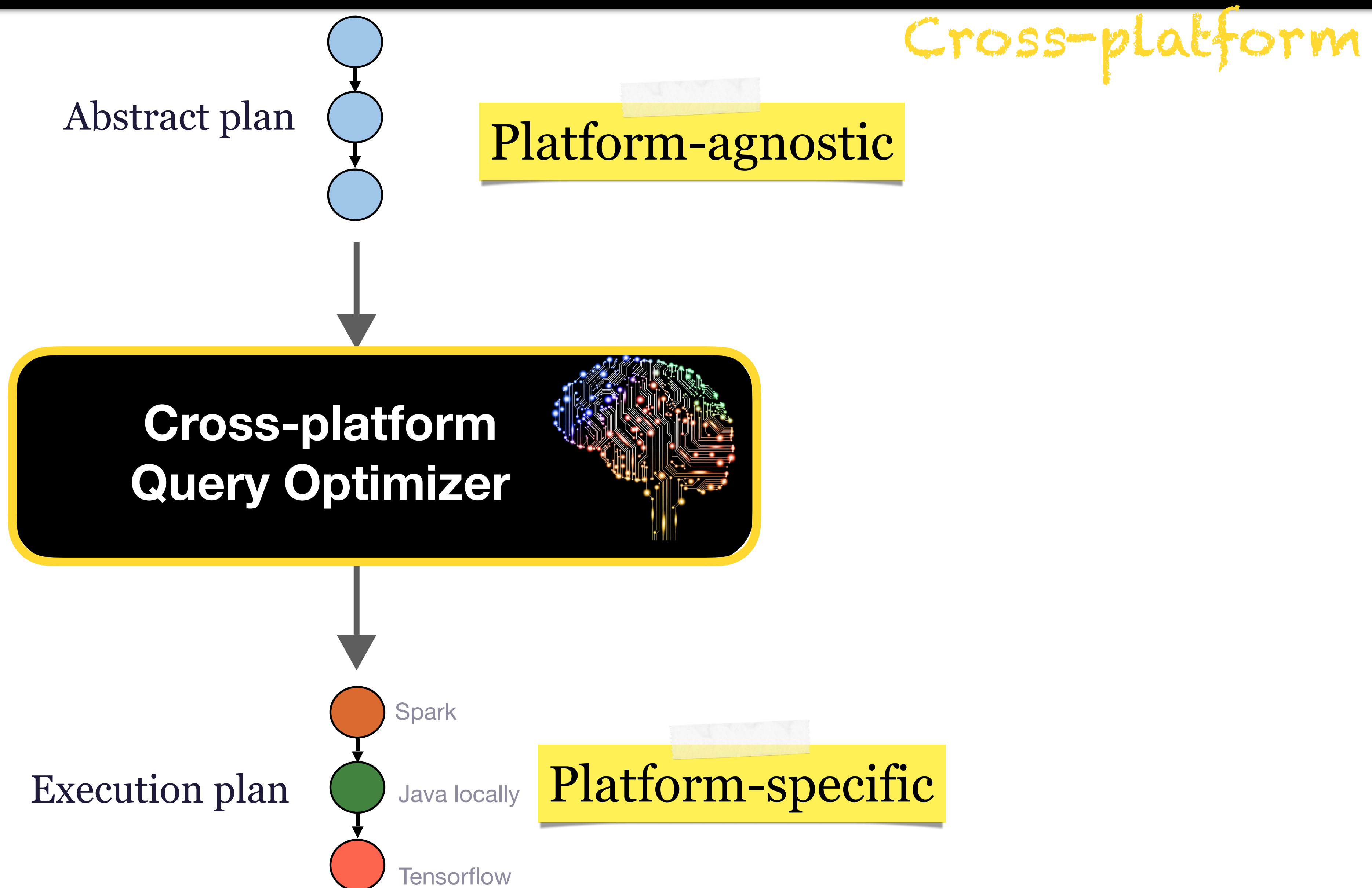
Automated
platform selection

Apache Wayang: From research prototype to Apache TLP

Cross-platform



Goal of QO in cross-platform settings



Cost-based QO in Apache Wayang

Cross-platform

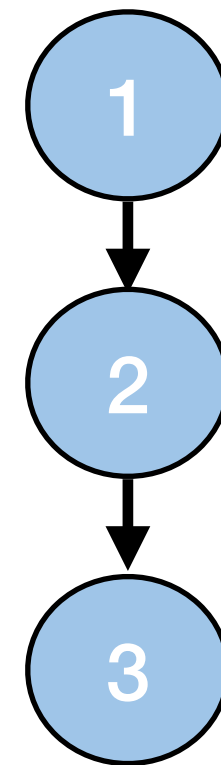


Cost-based QO in Apache Wayang

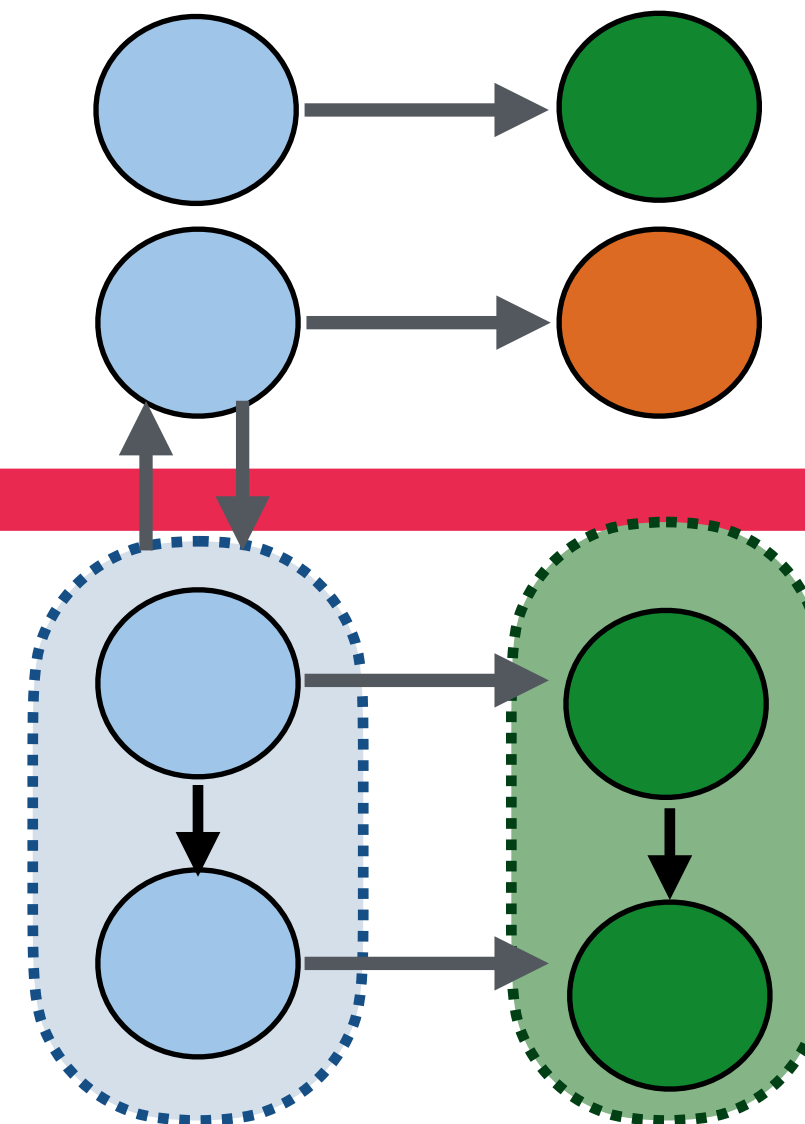
Cross-platform

Plan Inflation

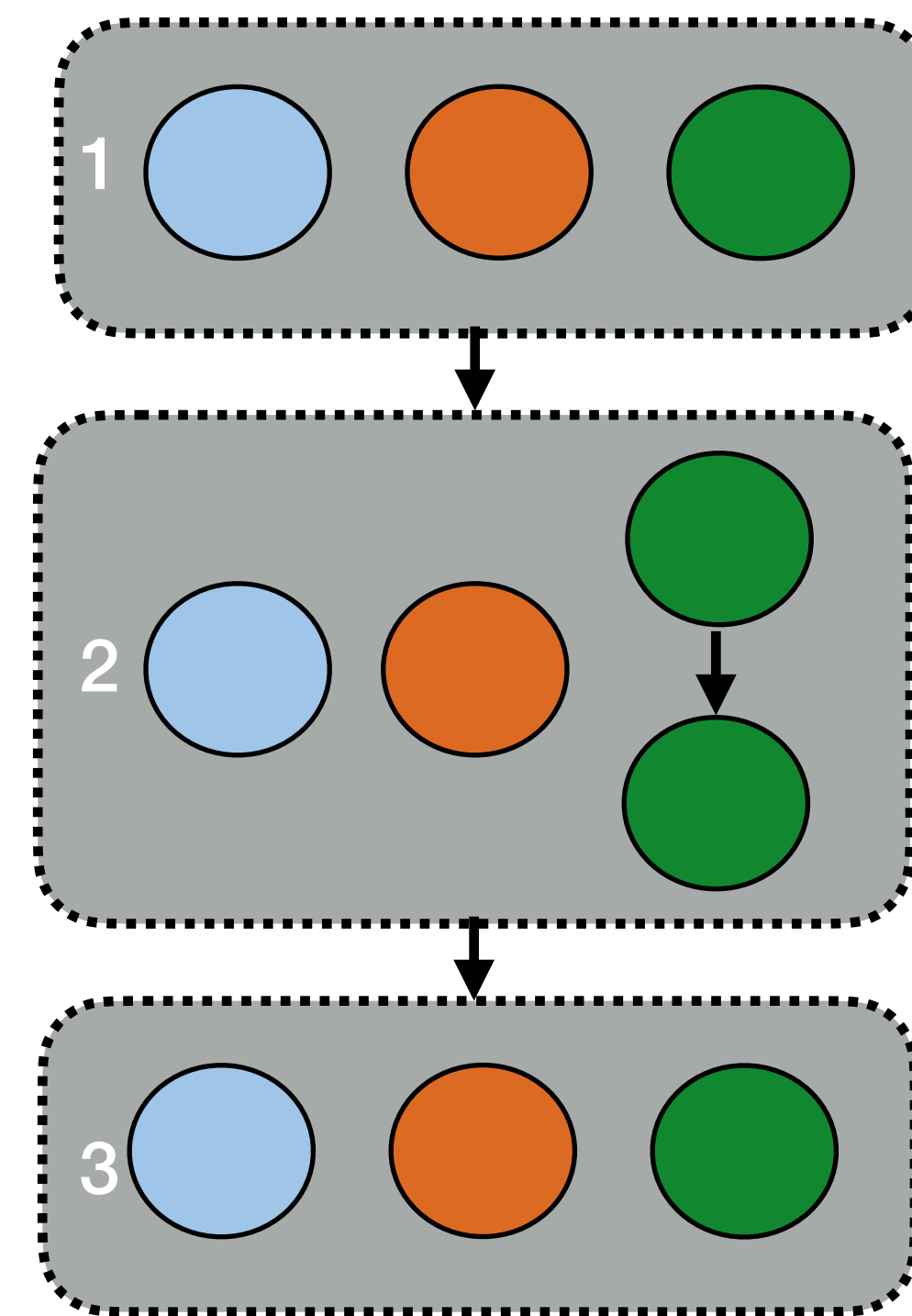
Wayang plan



Mappings

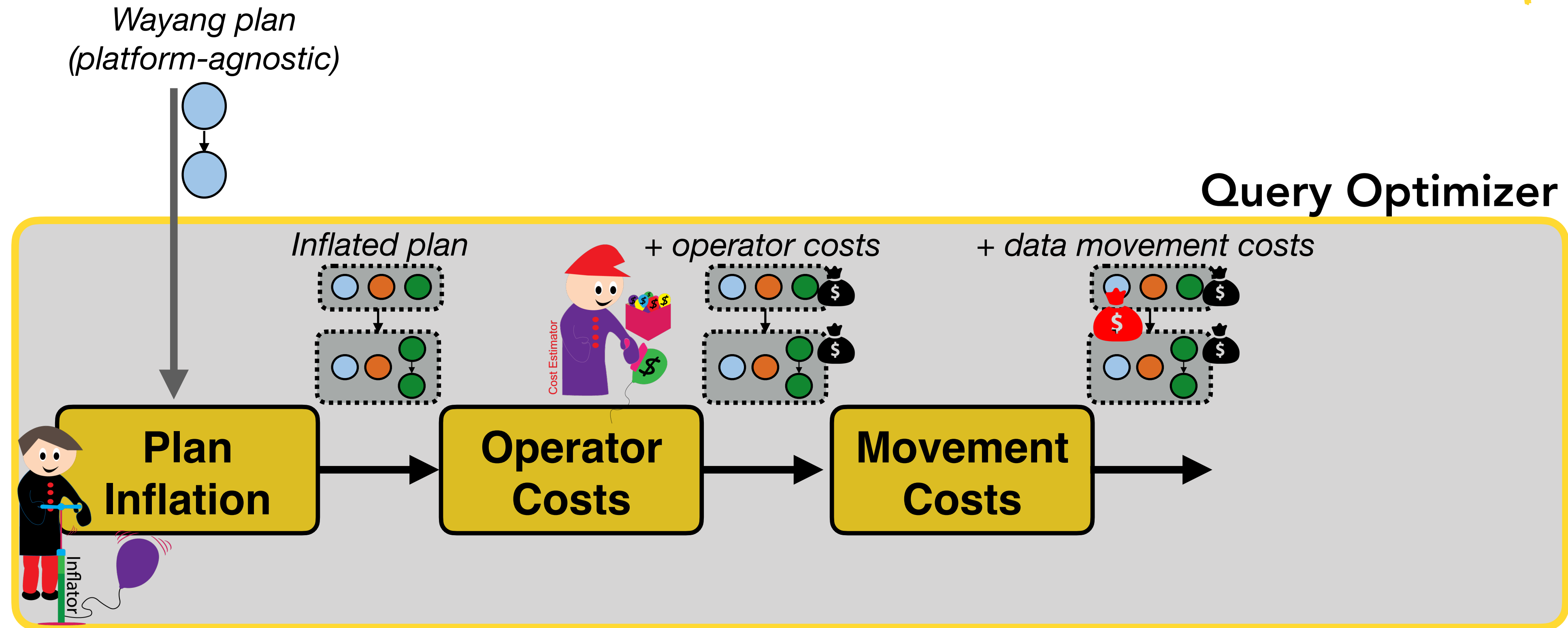


Inflated plan



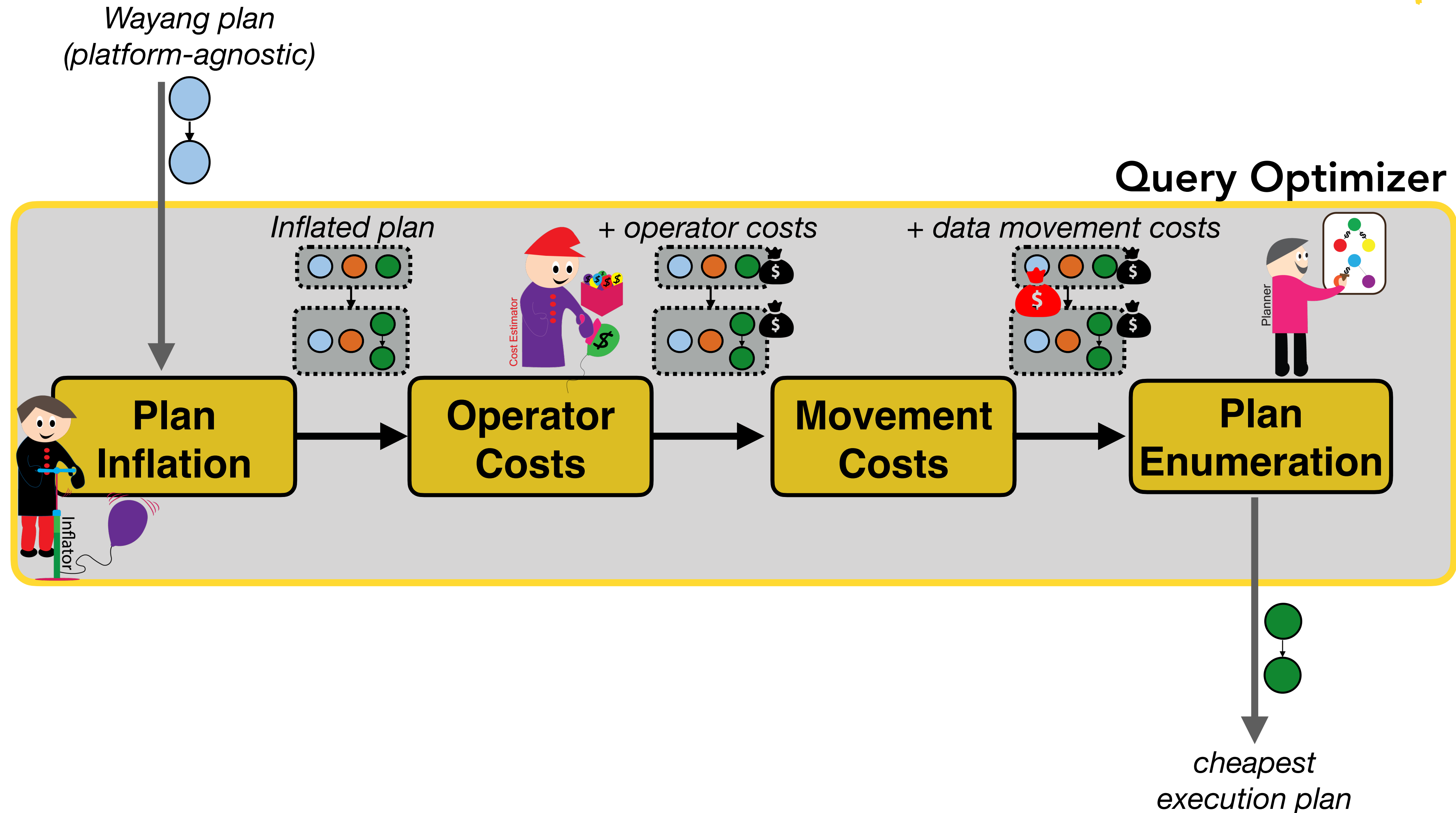
Cost-based QO in Apache Wayang

Cross-platform



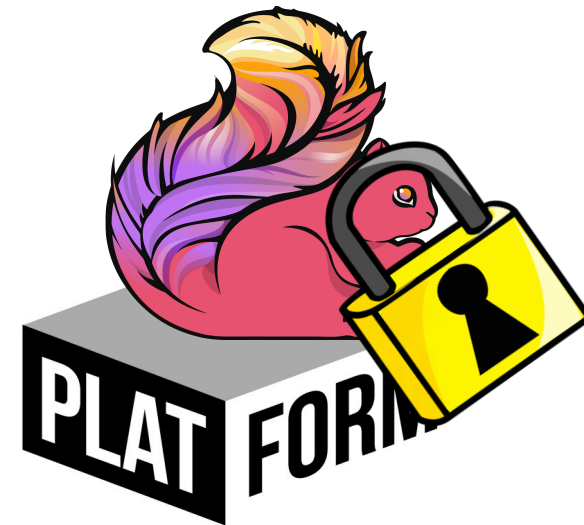
Cost-based QO in Apache Wayang

Cross-platform



Problems with cost-based optimizer

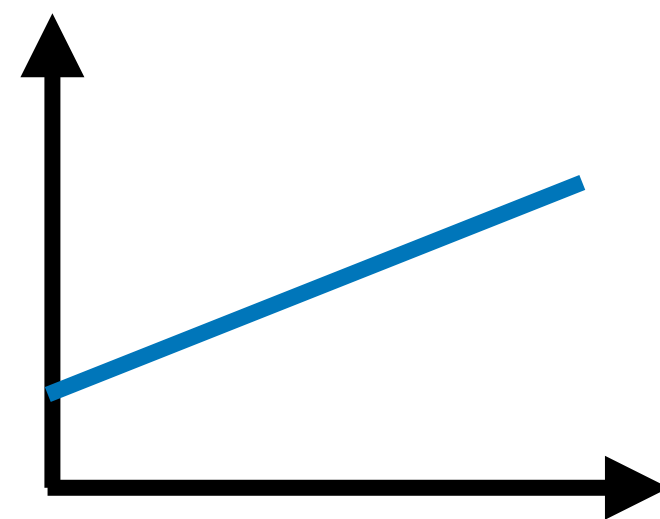
Cross-platform



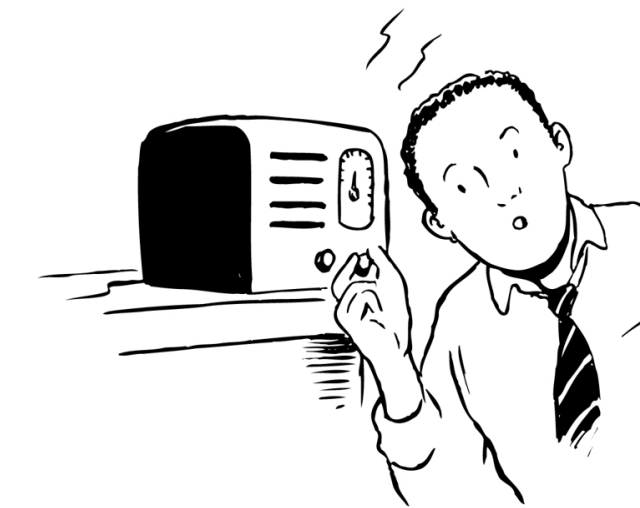
Low control

$$\begin{aligned} G_{\mu\nu} &= 8\pi G(T_{\mu\nu} + \rho\lambda g_{\mu\nu}) \\ \int_a^b f'(x) dx &= f(b) - f(a) \\ \ell &= mc^2 \\ \pi &= 3.141592653 \dots \\ \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} &= \gamma \end{aligned}$$

Tedious definition of cost formulas



Linear function assumption



$$Cost_{sparkjoin} = \textcircled{a_1}x_{cpu} + \textcircled{a_2}x_{memory} + \textcircled{a_n}x_{network} + \dots$$

Hard fine-tuning

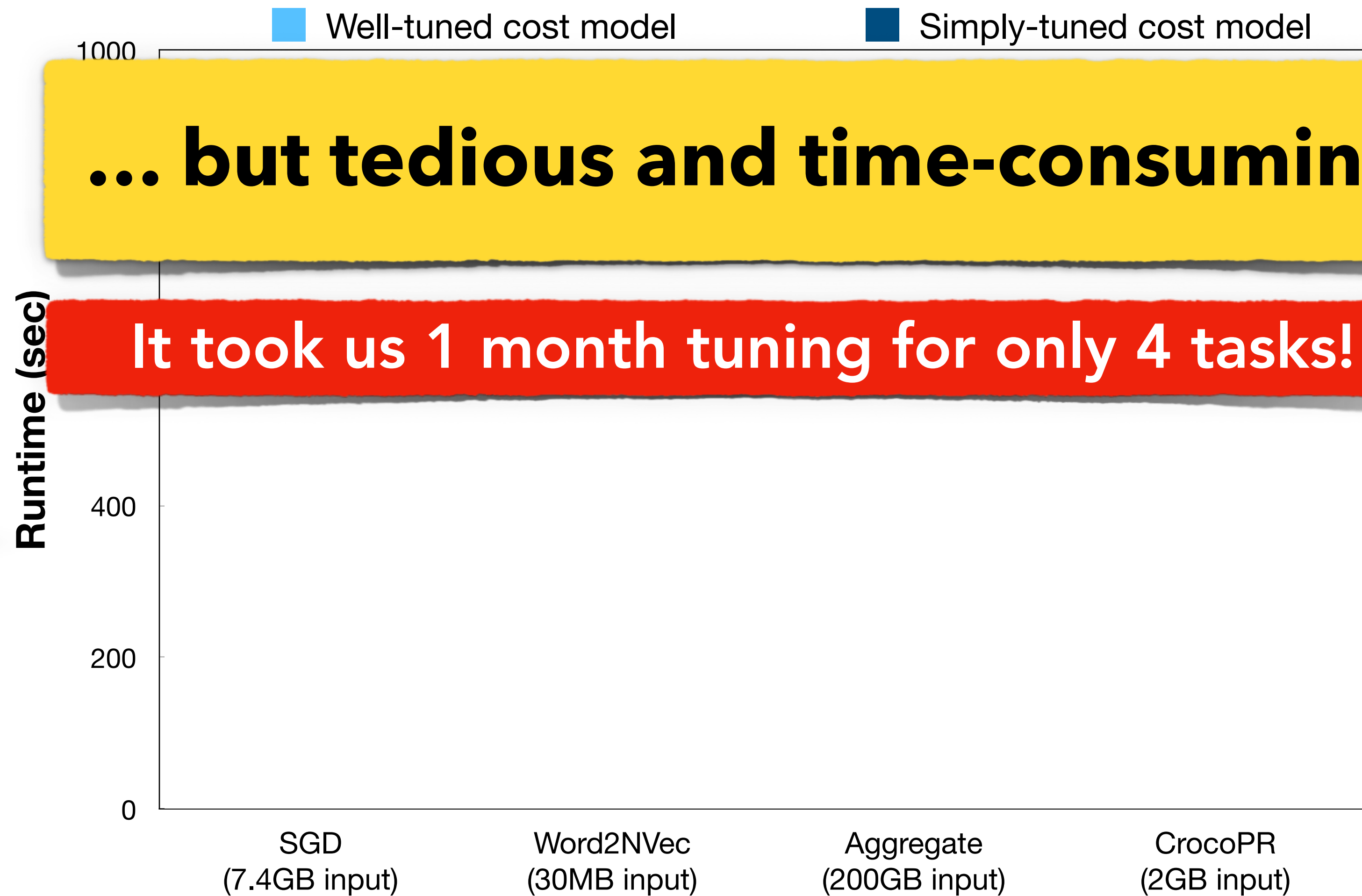
Cost model tuning is important ...

Cross-platform

... but tedious and time-consuming!

It took us 1 month tuning for only 4 tasks!

Using real
cardinalities

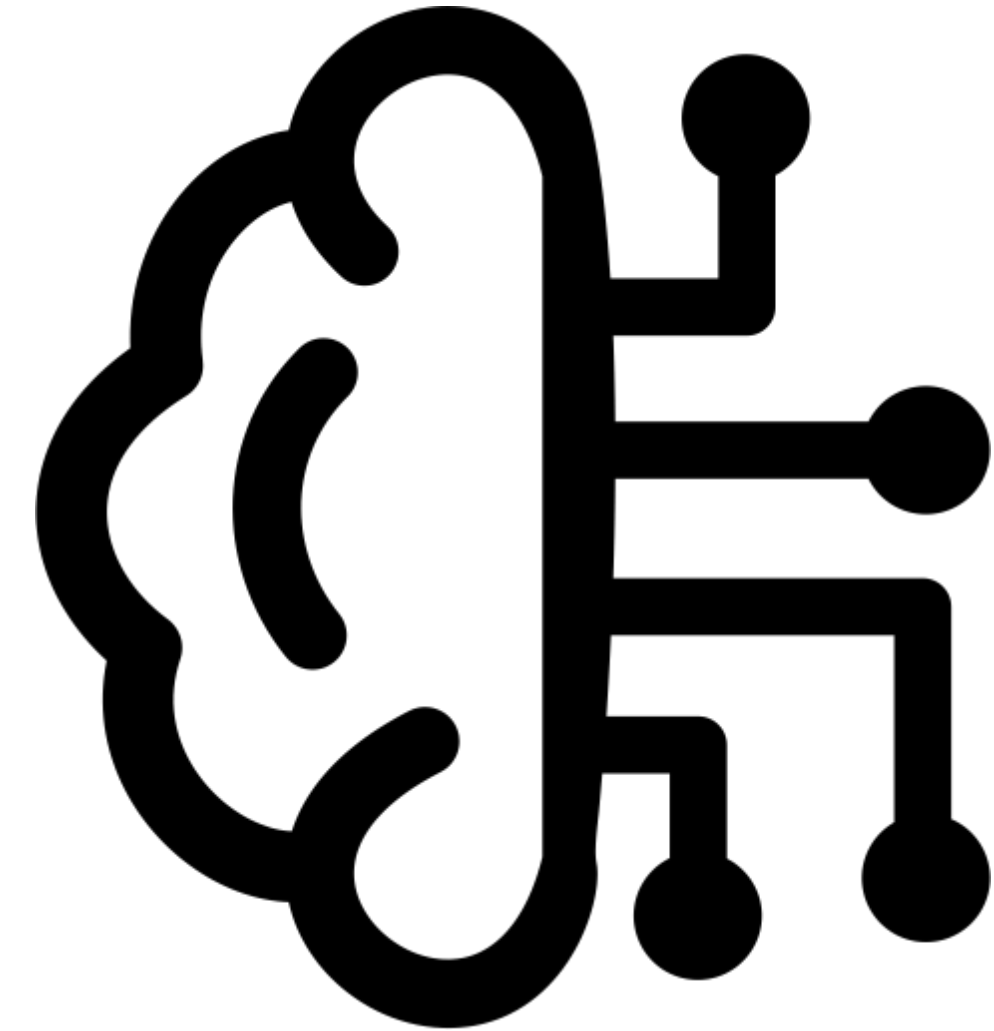
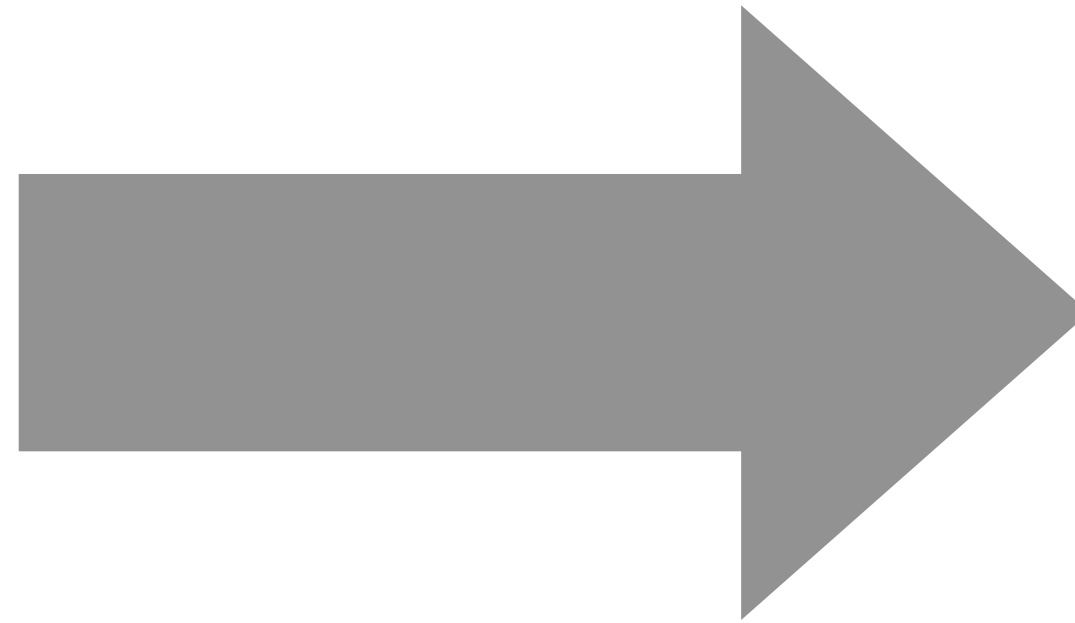


ML opportunity again!

Cross-platform



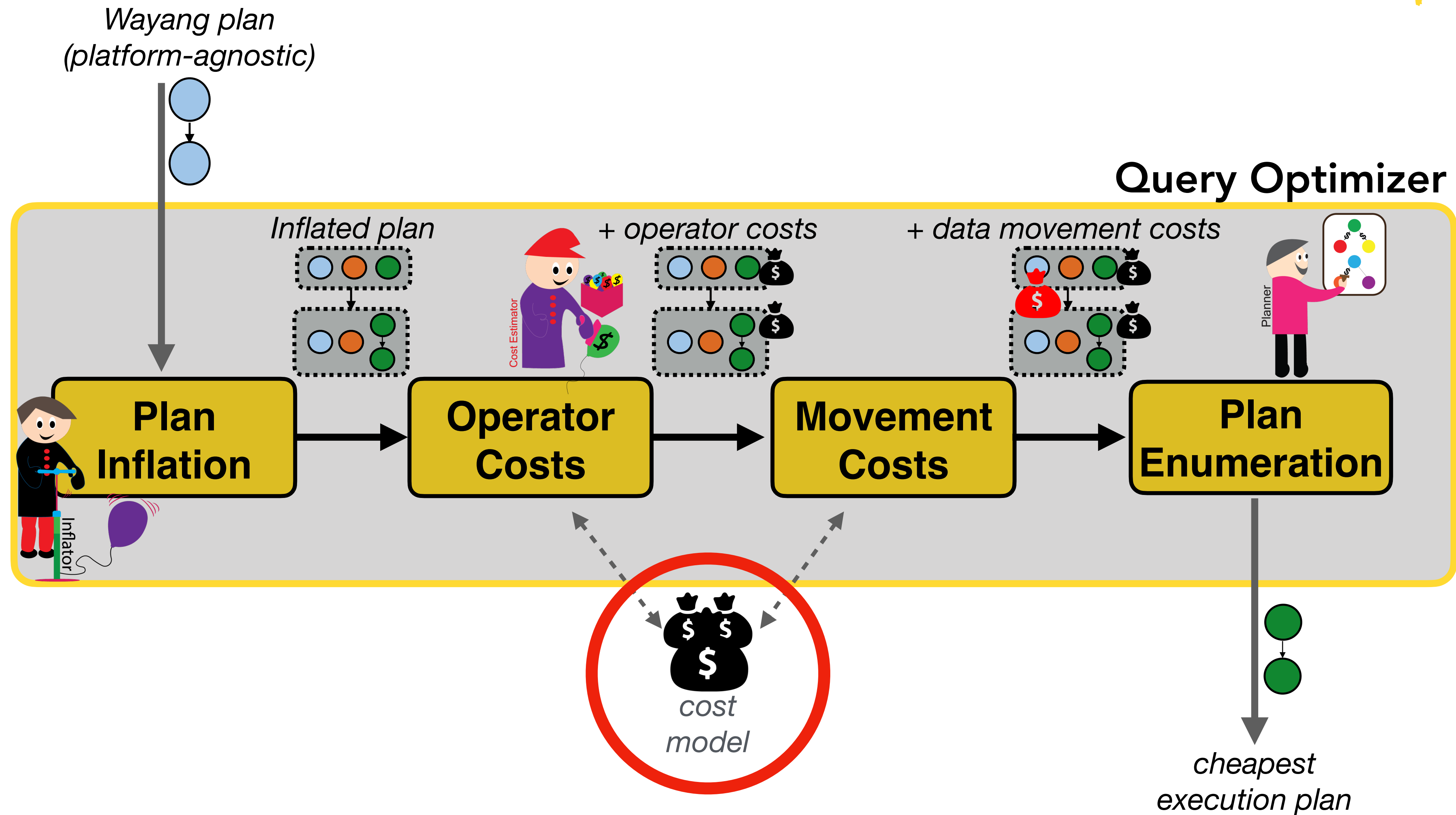
Cost



ML

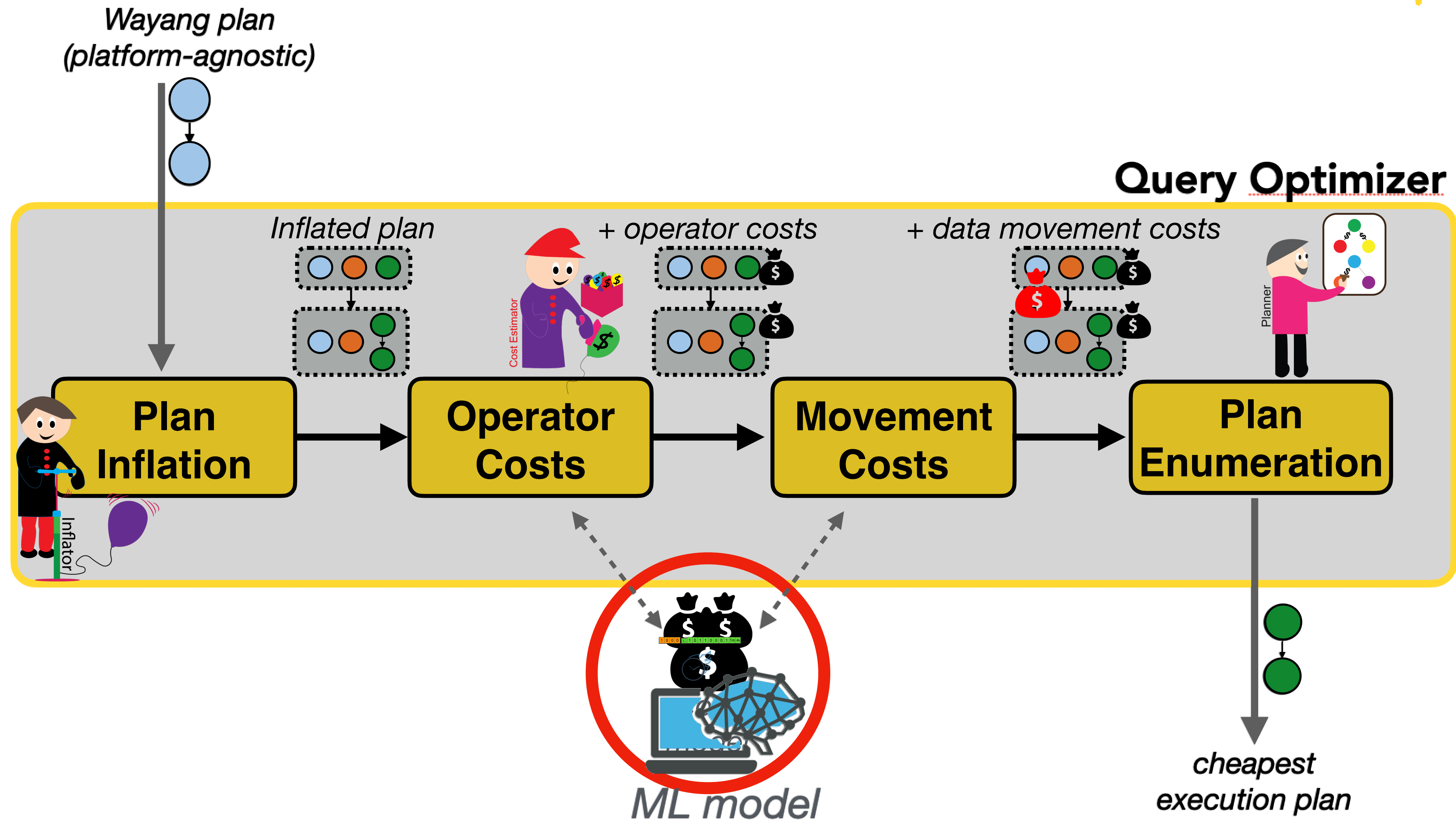
ML-based cross-platform QO

Cross-platform



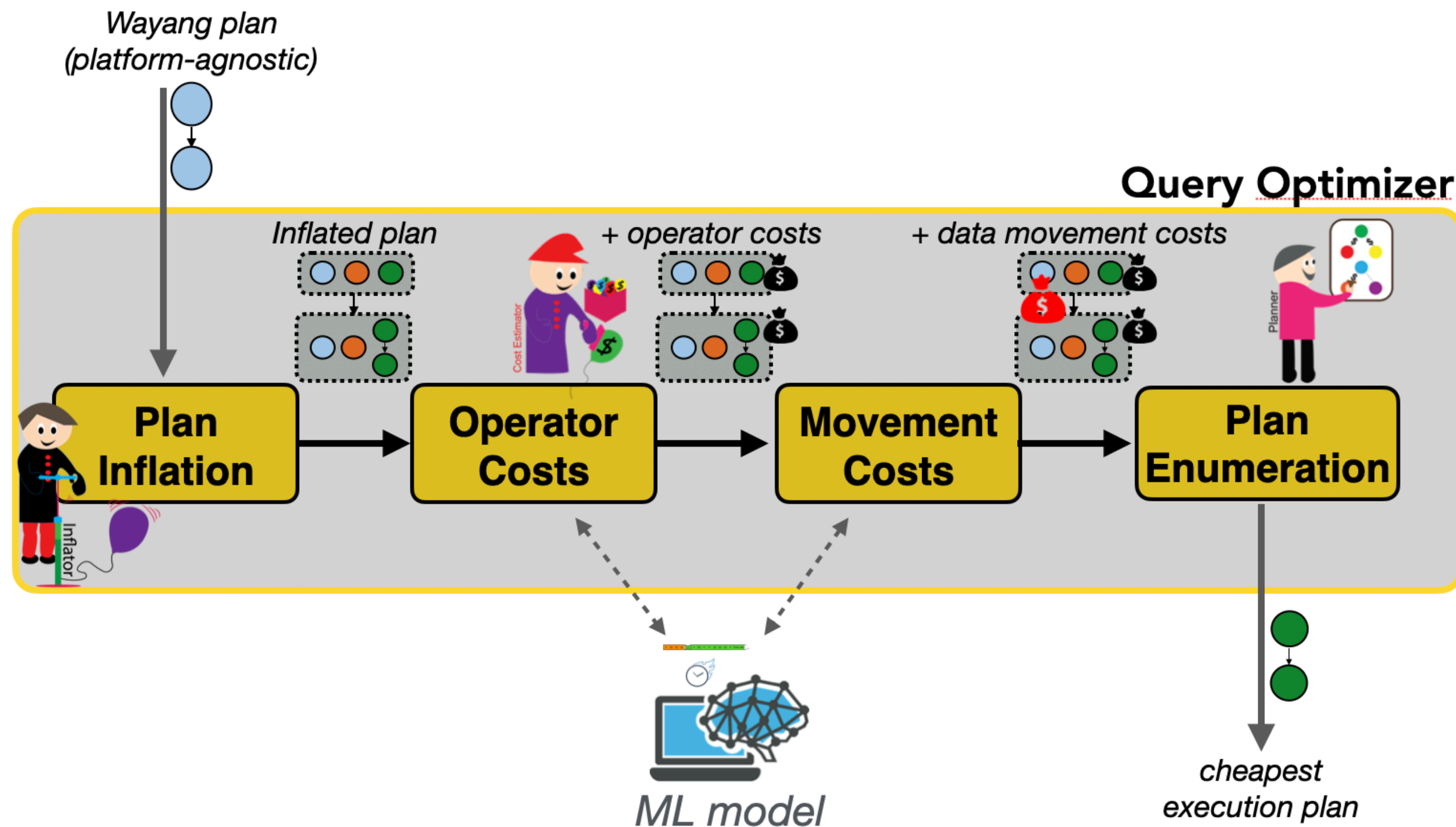
ML-based cross-platform QO

Cross-platform



ML-based cross-platform QO

Cross-platform



Lots of training data

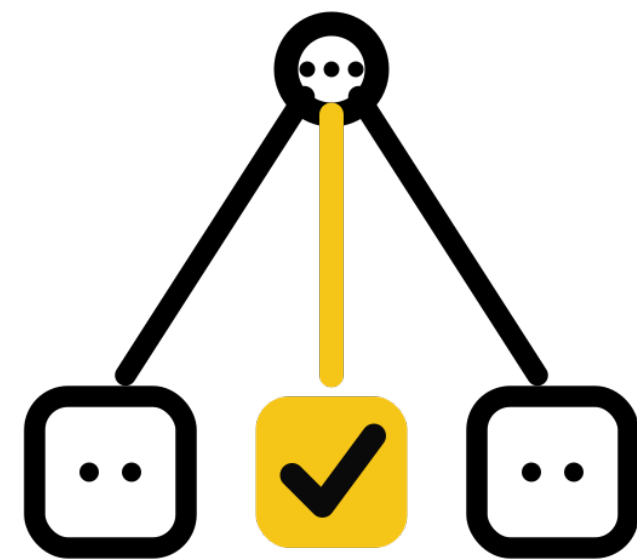
Exponential search space

Inaccurate regression models

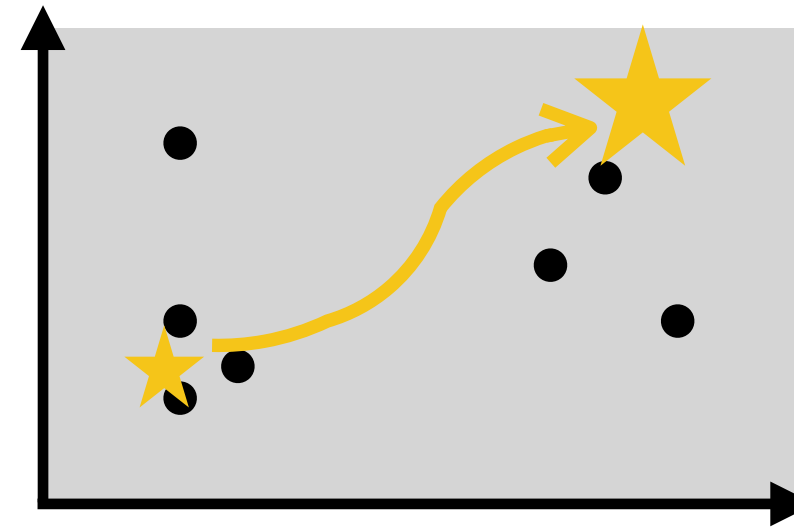
An alternative approach for cross-platform QO

Cross-platform

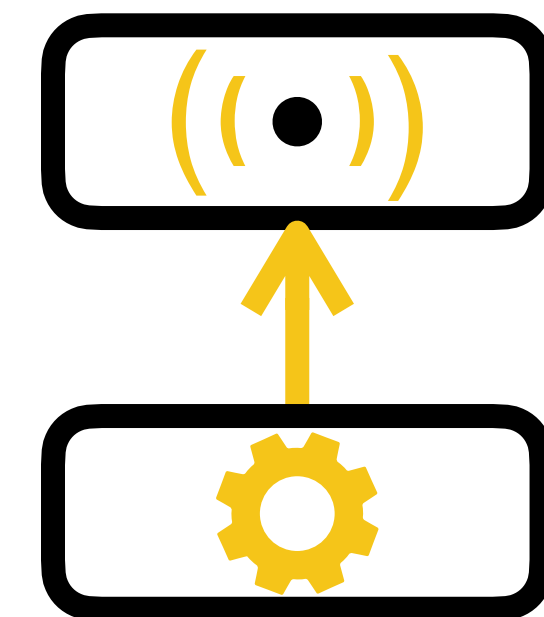
Platform selection as a classification problem



From enumerate & prune to generate & explore

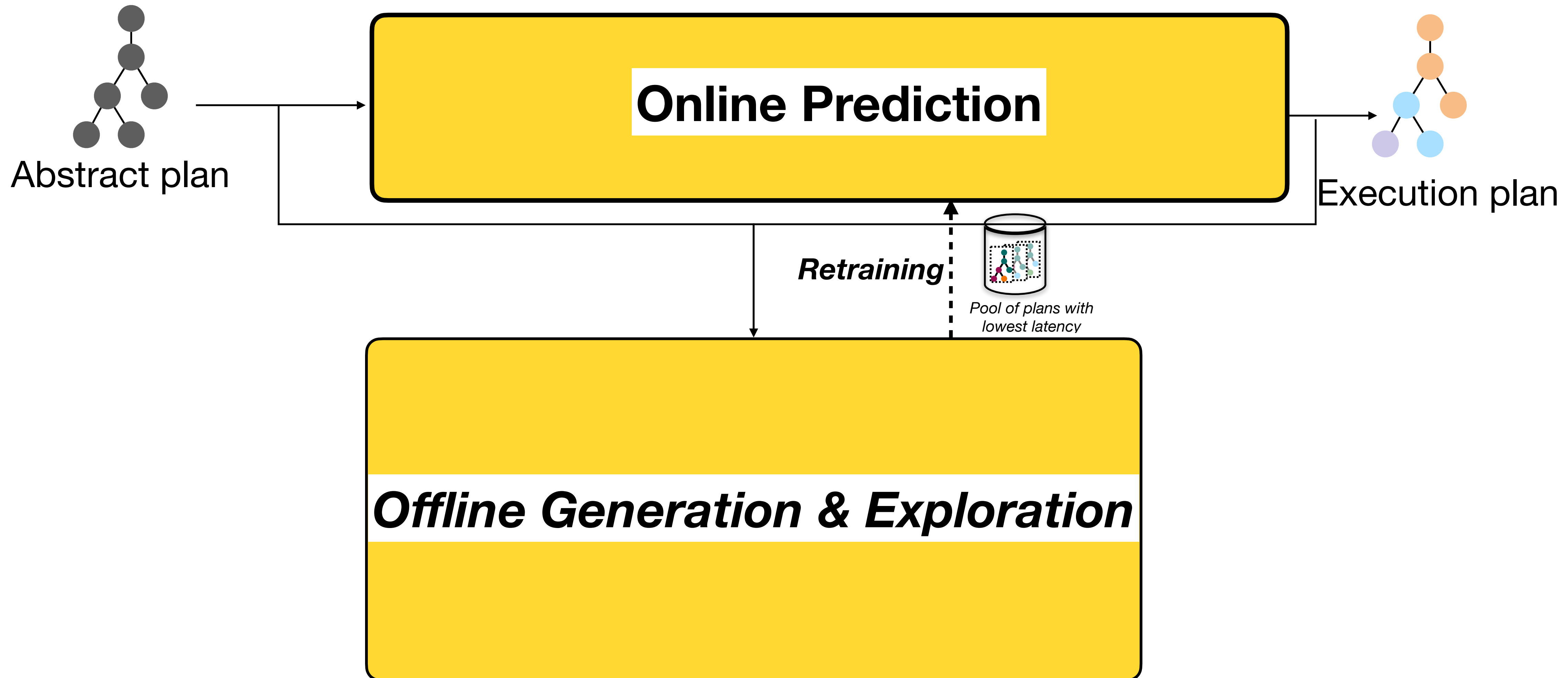


Online / offline architecture



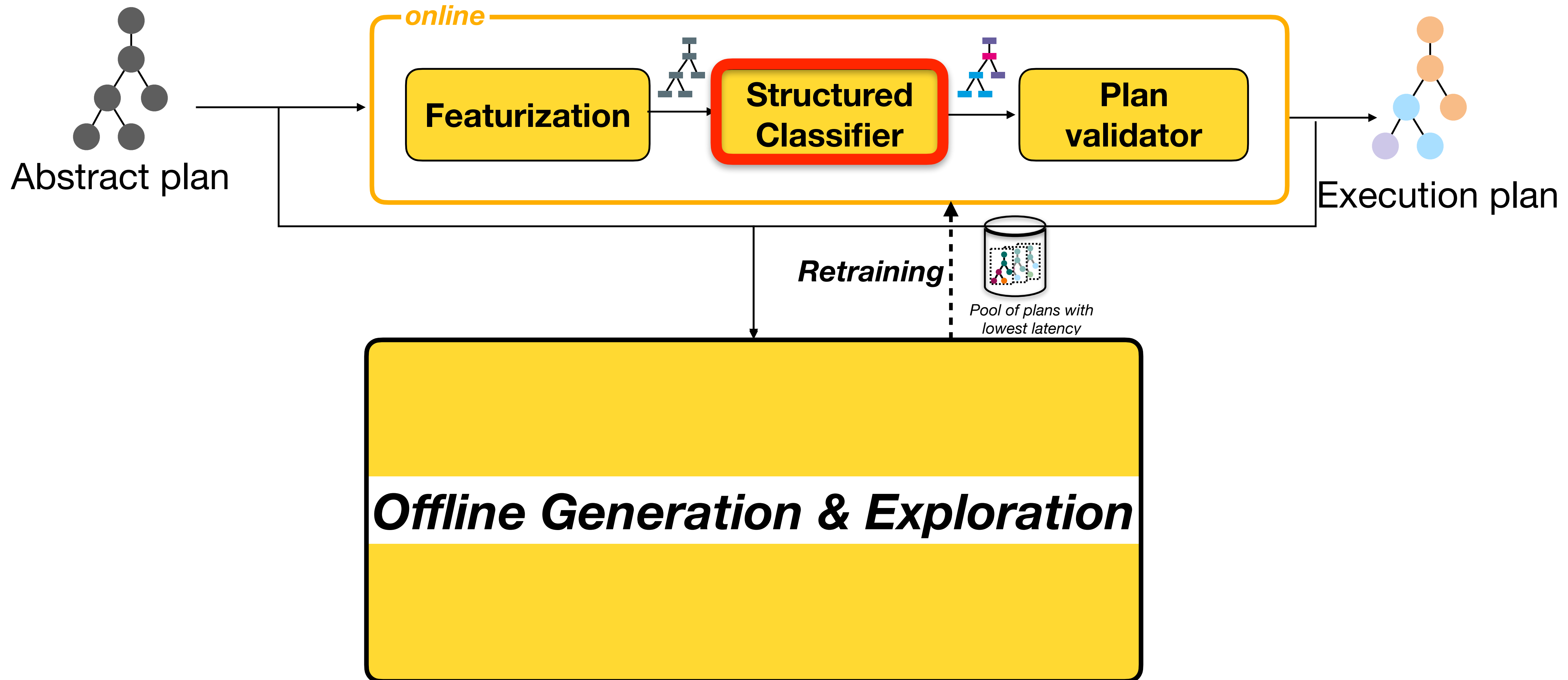
Alternative cross-platform QO architecture

Cross-platform



Alternative cross-platform QO architecture

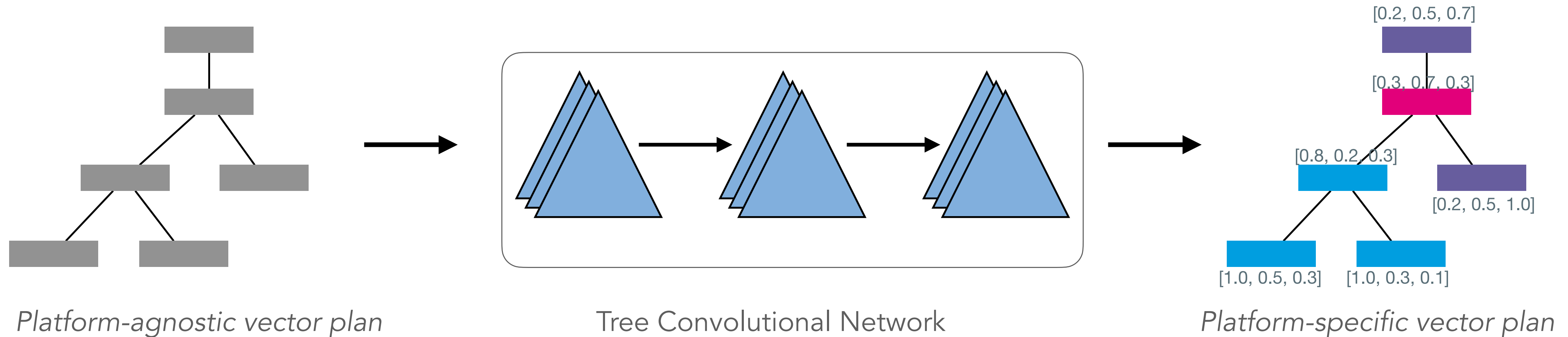
Cross-platform



Structured classifier

online

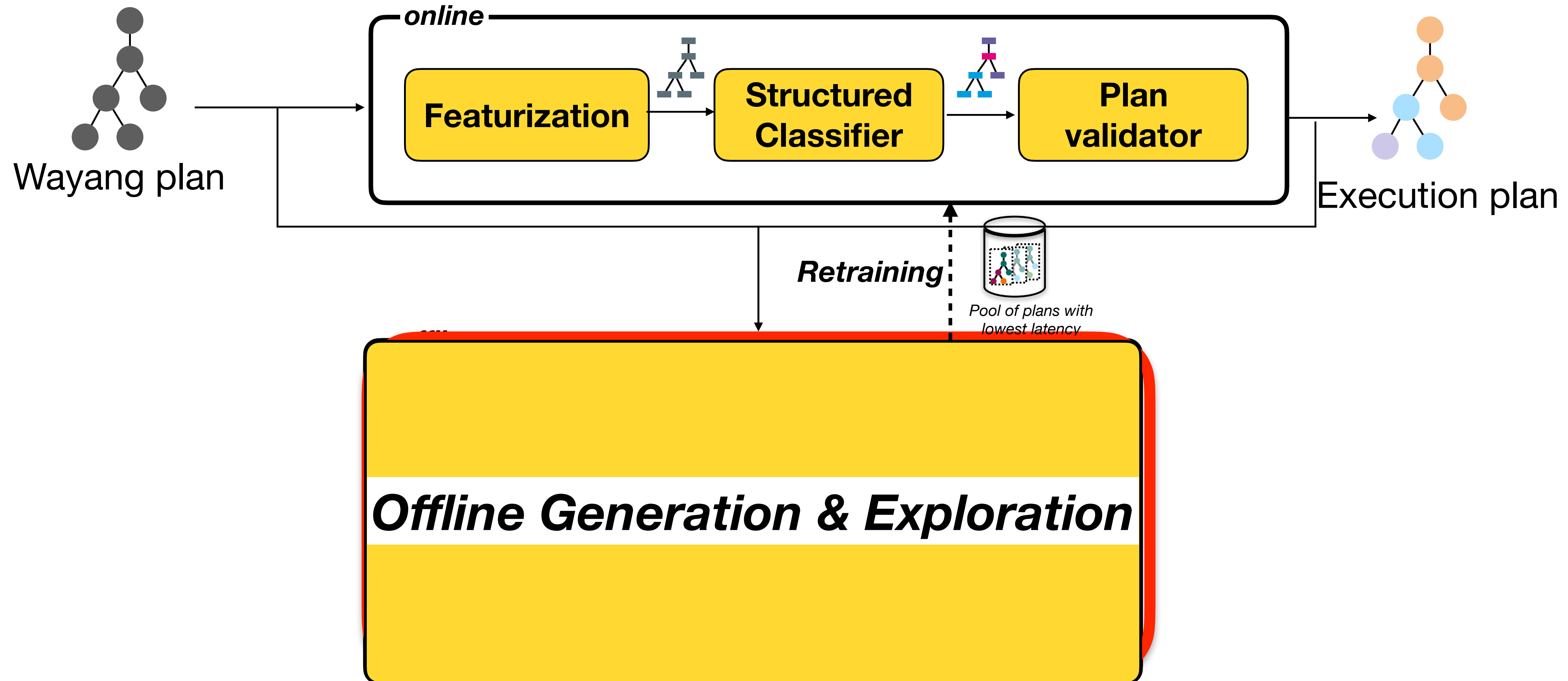
Cross-platform



Training data
No runtime labels
Only (Abstract, Execution) plan pairs

Alternative cross-platform QO architecture

Cross-platform

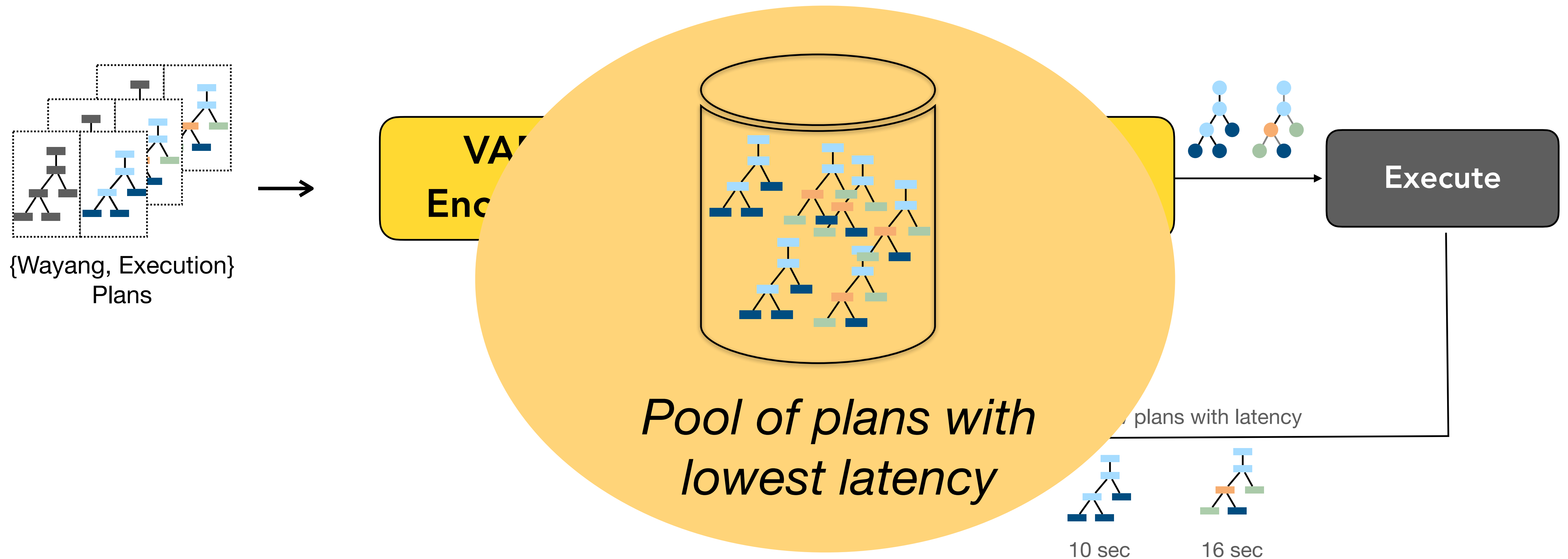


Generate & explore

offline

Cross-platform

Conditional attribute-regularized β -variational autoencoder

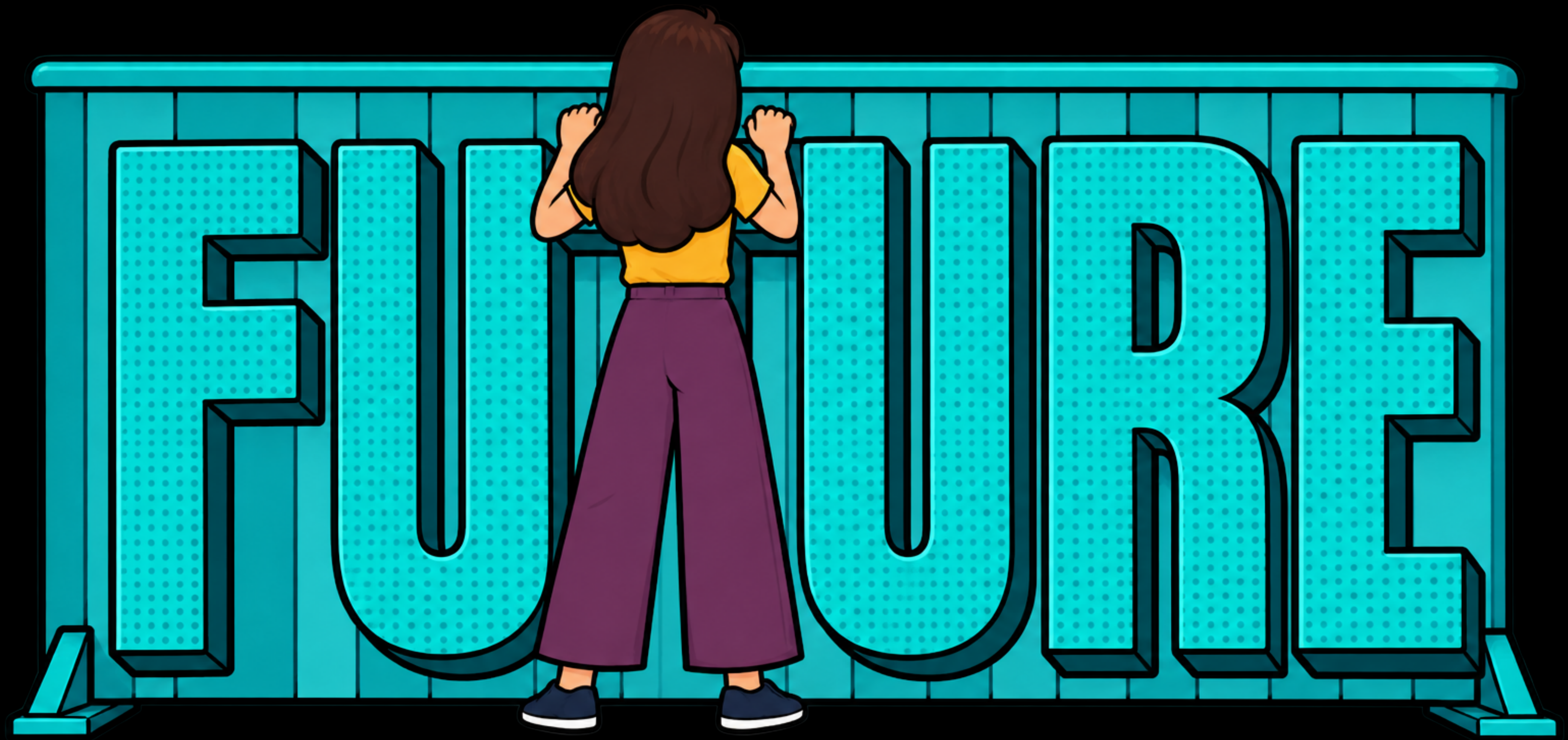


Query Optimization In The Wild

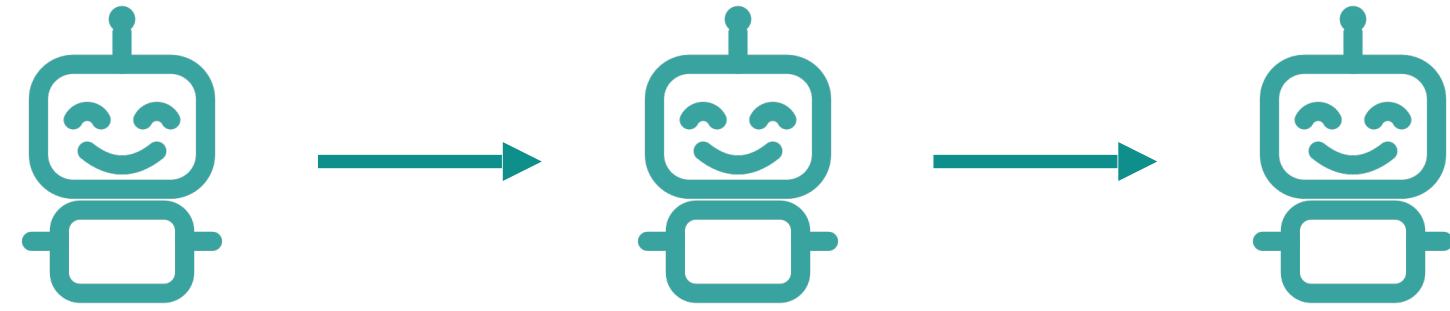
Relational

Cross-platform

What's next?



Multi-agent pipelines



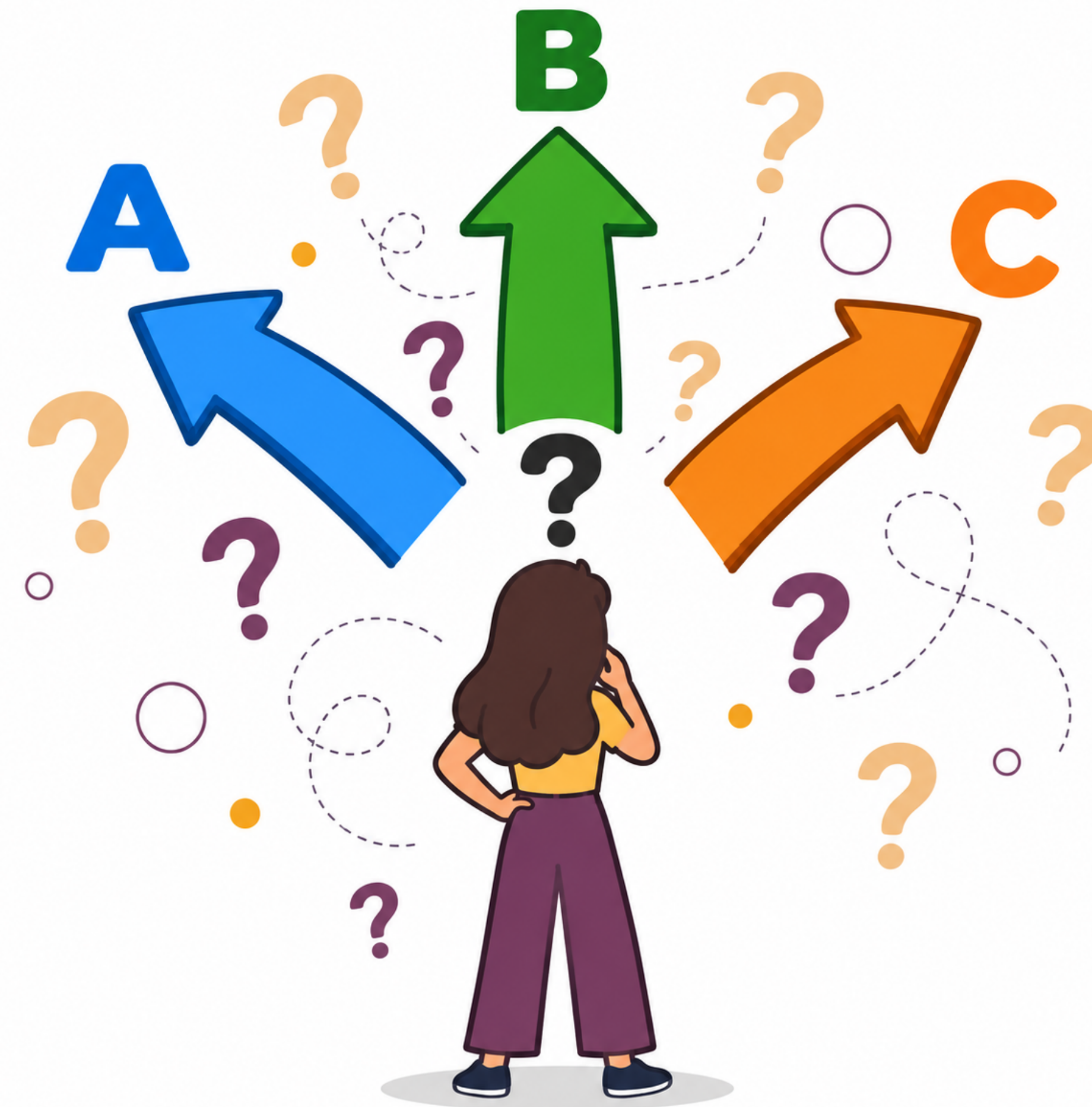
Models, tools, and data sources into one workflow

Fixed structure

Models out of habit

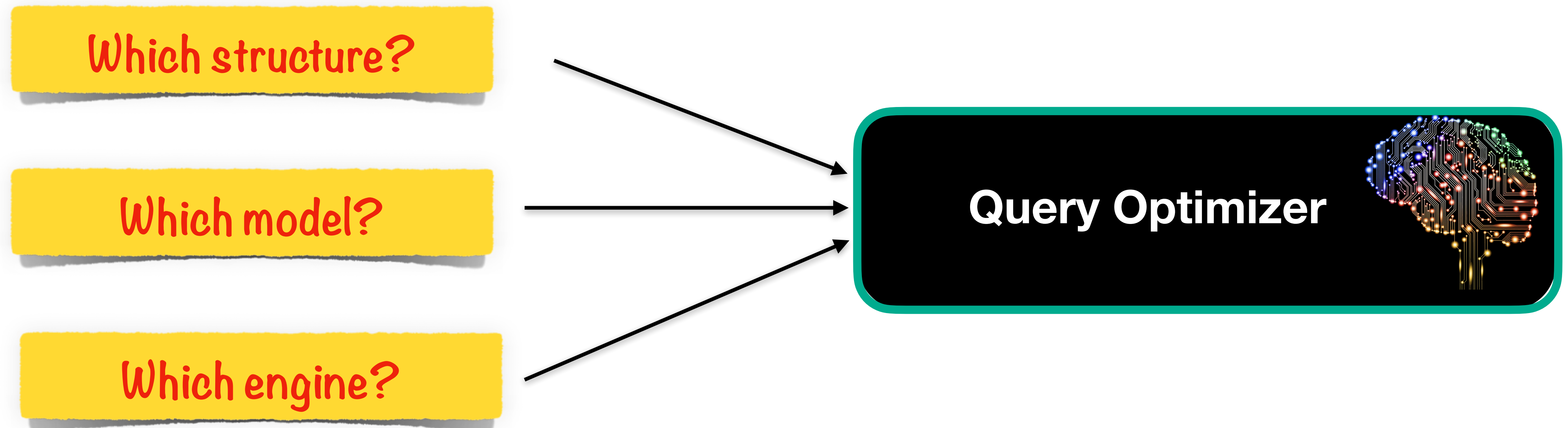
Hardcoded engine

Multi-Agent



Query optimization for multi-agent pipelines

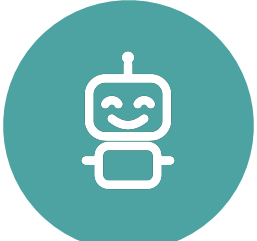
Multi-Agent



Query optimization for multi-agent pipelines

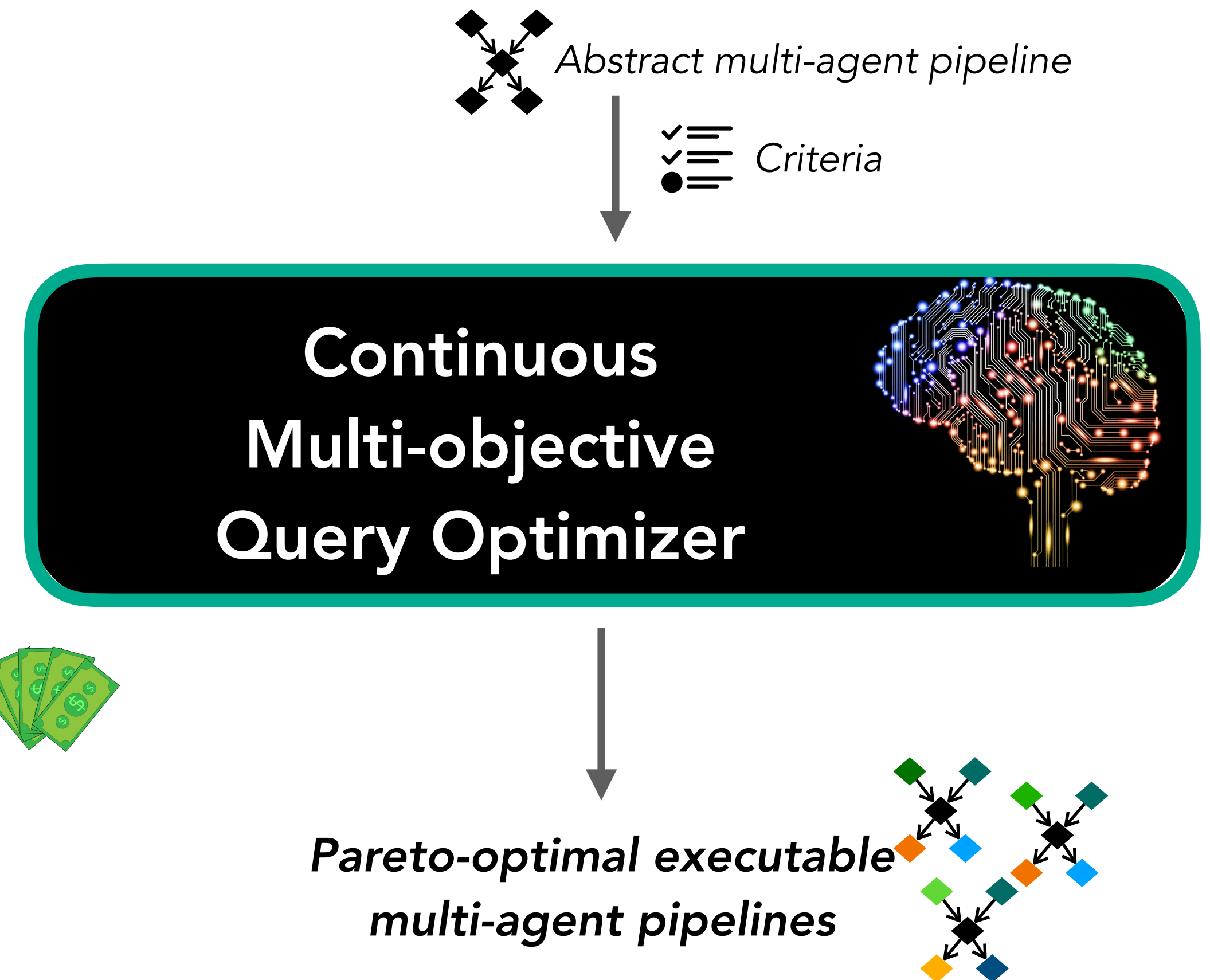
Multi-Agent

 3 coupled dimensions

 Stochastic operators

 Multiple objectives

- Latency 
- Monetary cost 
- Accuracy 
- Energy 



Query Optimization In The Wild

Relational

Cross-platform

Multi-Agent

Many thanks



Juri Petersen



Mads Sejer Pedersen



Xiao Li



Ioana Giurgiu

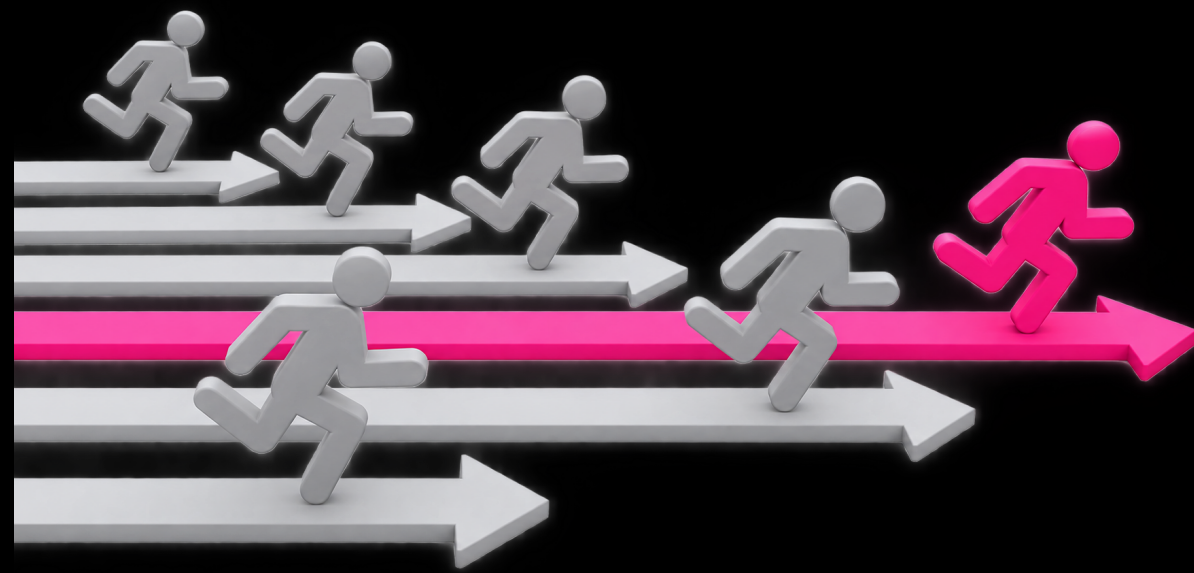
IBM Research

CARLSBERG
FOUNDATION

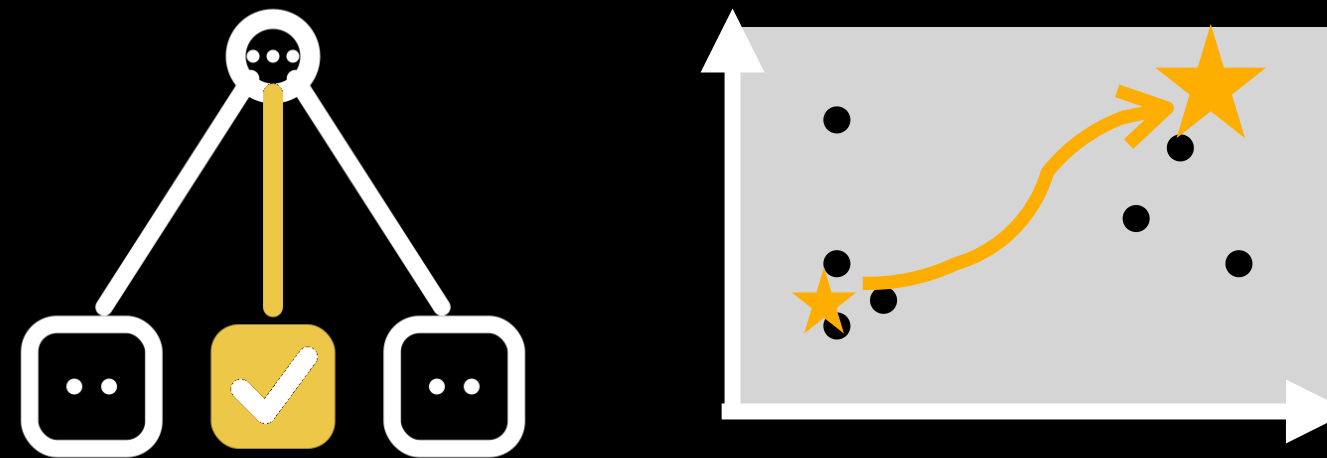


Query Optimization In The Wild

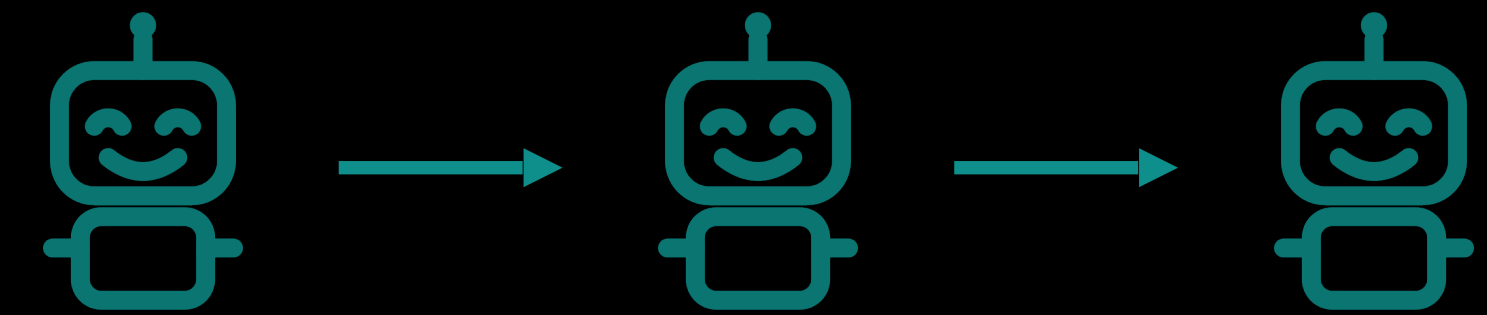
Relational



Cross-platform



Multi-Agent



Learning-to-rank
plan enumeration

Classify-generate-explore

Continuous
multi-objective
optimization

Query Optimizer

