Towards Foundation Database Models

Systems Research @ Google

Carsten Binnig & Johannes Wehrstein
(work done while at Google)

Co-Authors:



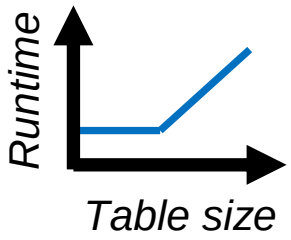


CIDR Conference on Innovative Data Systems Research
Best Paper Award
Google
TECHNISCHE UNIVERSITÄT DARMSTADT

A Prominent Direction: Learning DB Tasks

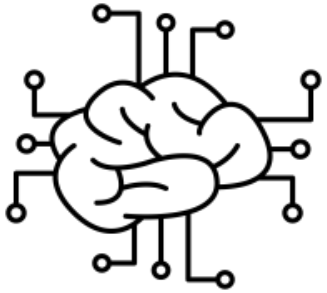
Simplifying assumptions: uniform data, no correlations

Cost Model



Replace

Learned Cost Model



No simplifying assumptions: learn from real characteristics

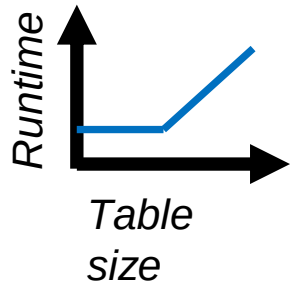
Classical DB Component

Learned DB Component

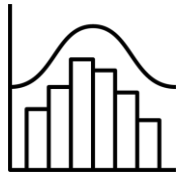
Learned approaches have shown to **significantly improve** Database Performance

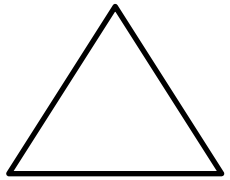
Learned DB Tasks: Wide Applicability

Cost Models



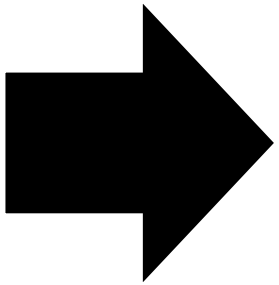
Card. Estimation

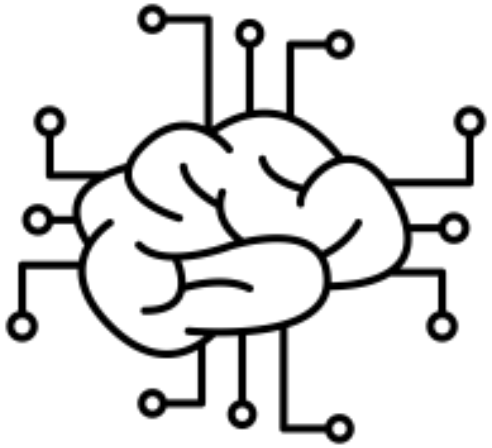




Indexes

Schedulers





ML Models

Learned approaches have been used successfully for a **large spectrum of database tasks**

Initial Approach for Learned DB Tasks

Instance-specific Learning: Learn a Model for a specific Dataset & Task

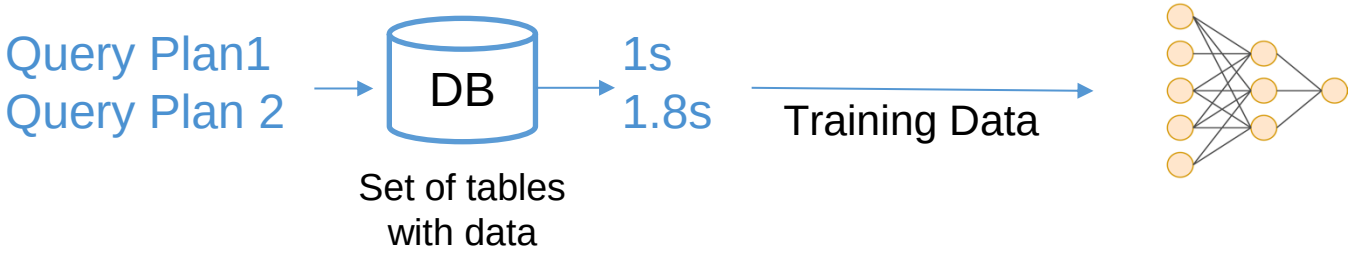
Example: Learn an instance-specific Cost Model

Training

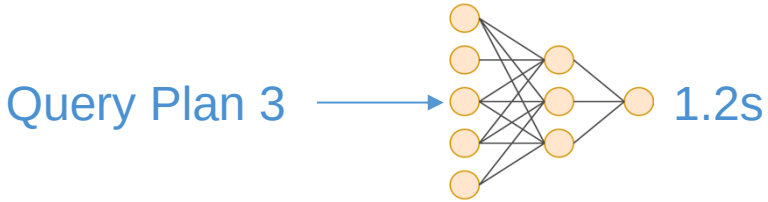
1) Run Workload on a given Dataset:

2) Collect Training Data (i.e., Runtime of Queries)

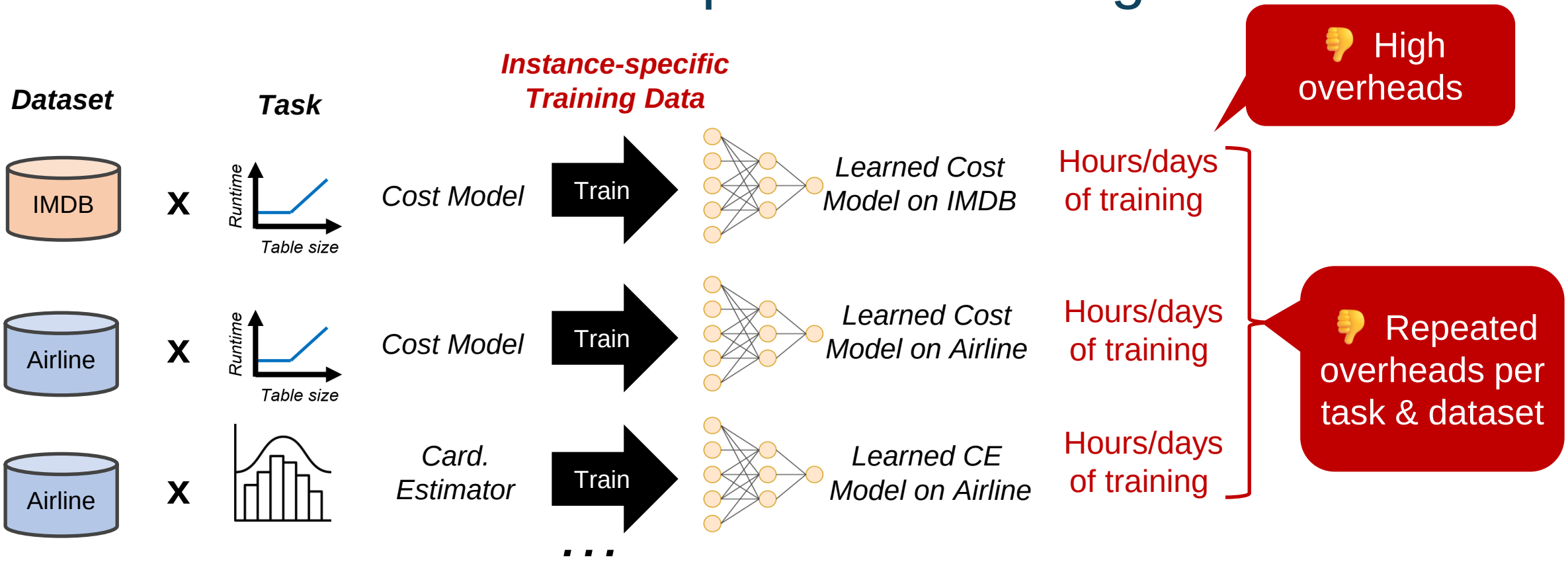
3) Train Model:



Inference Use Model for new Queries (e.g., predict runtime) over same Dataset

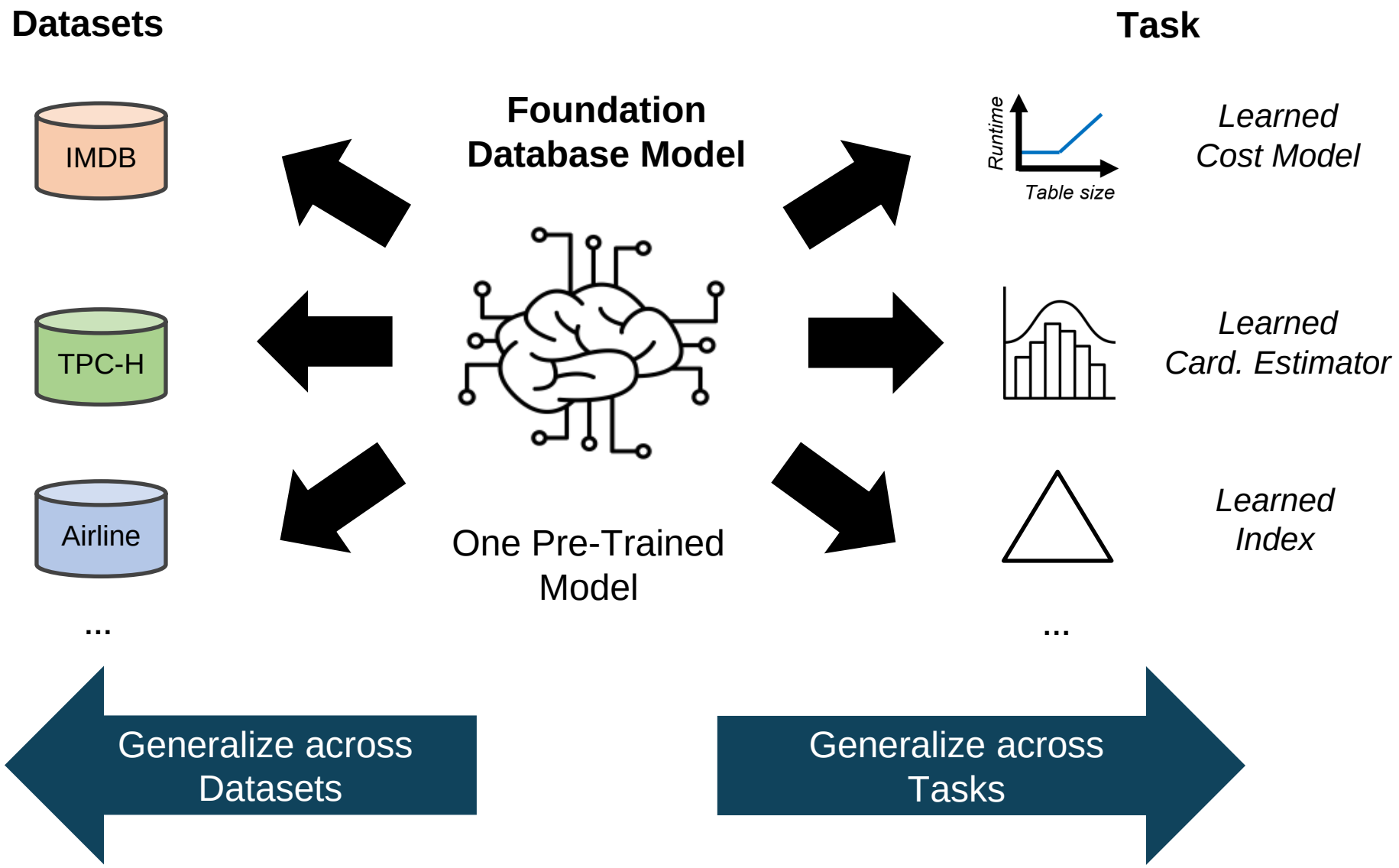


Main Issue of Instance-specific Learning

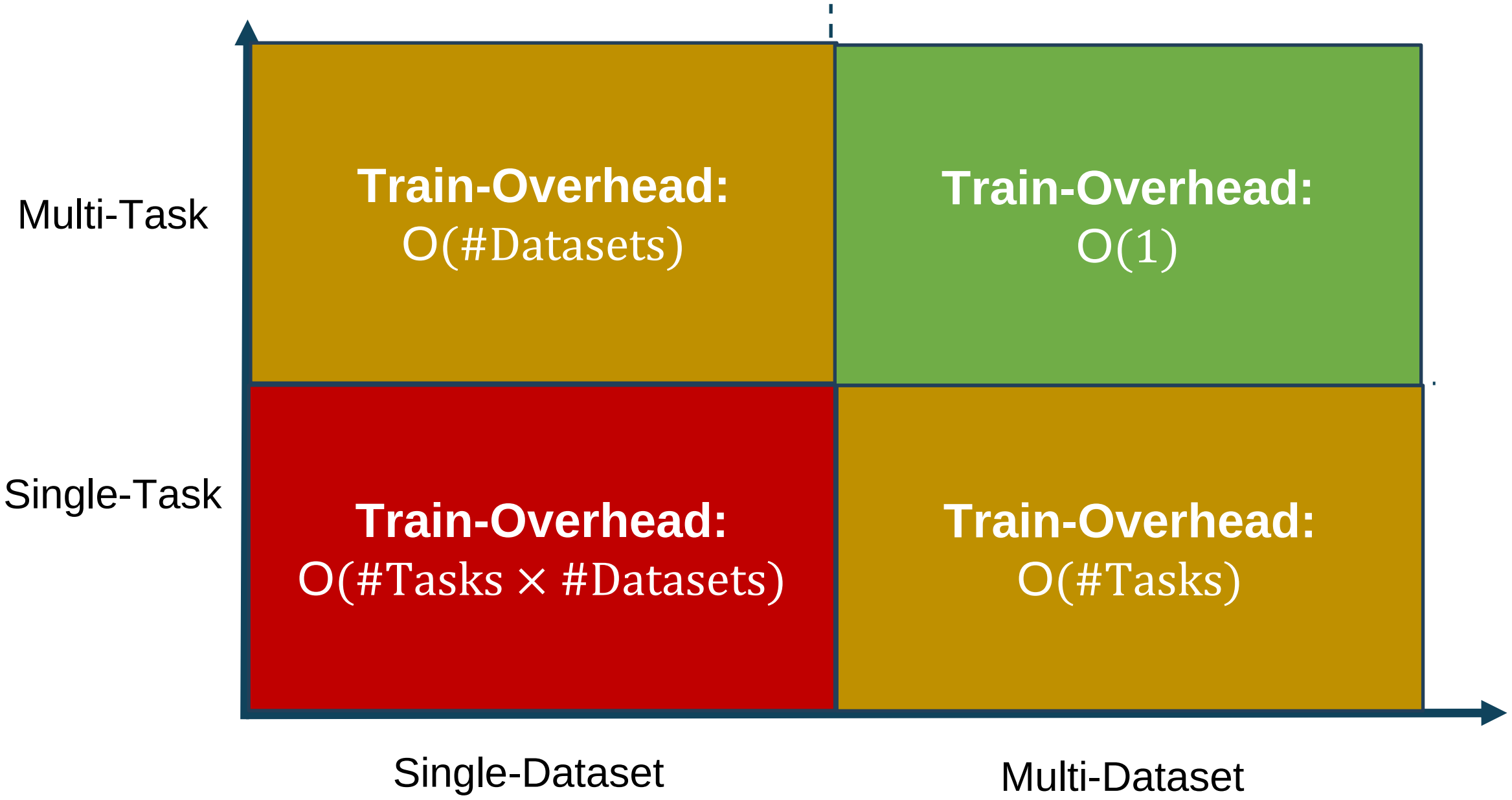


High overheads at the **scale of the cloud DBs** which host **1000's of customer datasets**

Our Vision: Foundation Database Models

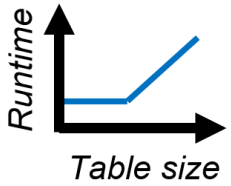
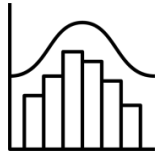
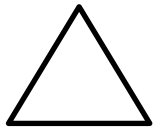


The Learning Landscape & Overheads



How to realize a Foundation Database Model?

What do Models learn from?

Database Task		Input for an ML Model		Prediction
Learned Cost Model		Data	Physical Plan	Runtime
Learned Card. Estimator		Data	Logical Plan	Cardinality
Learned Index		Data		Key Position
...		...		

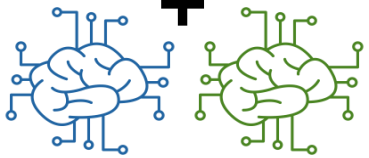
Observation: Models for **different tasks share similar information** (data, logical /physical query plans, …)

Core Idea: Composable Pre-trained Models

Key Idea 2 - Play Lego:
Combine Experts +
A Shallow Model
(Downstream Task)

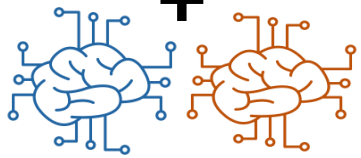
Predicted Cardinality

Simple Regression Model
for CE on TPC-H



Predicted Cost

Simple Regression Model
for Cost Est. on IMDB



Low overhead for down-stream model

Pre-trained experts (no overhead)

How Experts work?

Key Idea 1 – Decompose:
Small Pre-trained Experts
(Task- & Dataset-independent)

Small Pre-trained Experts

Data Expert

Logical Plan Expert

Physical Plan Expert

Other Experts

What do individual Experts learn?

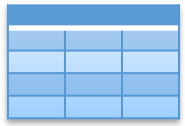
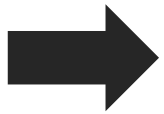


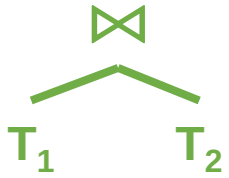
Table Data

Data Expert

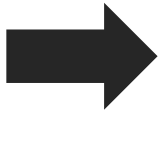


Summarizes Data Character-istics (eg Distr)

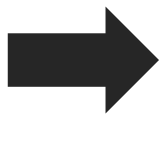
Table Representation (High-dim. vector)



Logical Plan



Logical Plan Expert

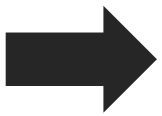


Summarizes how data is transformed

Logical Plan Representation (High-dim.vector)

Physical Plan

Physical Plan Expert



Summarizes Runtime Complexity

Physical Plan Representation (High-dim. vector)

How to pre-train such experts? Let us look at an example!

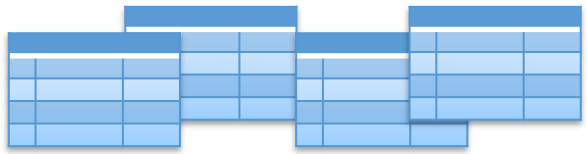
How to Pre-train a Data Expert?

Pre-train Tasks

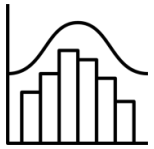
Task- & dataset independent?

Table Repr.

Data Expert

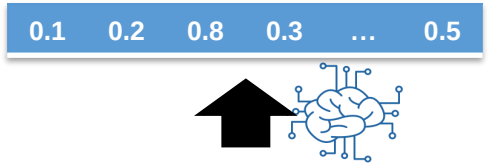


Pre-train on a large & diverse set of tables



Pre-Train Task 1:
Predict Col Distribution From Table Repr.

>100 km²	2
<100 km²	1



Population	Area	Density
0.1	0.248	MASK
0.1	0.60	0.33
1	0.100	0.1
…	…	…

Table (Cities)

Pre-Train Task 2:
Learn Row Corr. From Table Repr.

MASK = 4000

Task-independence?
General Pre-train Tasks

Dataset-independence?
Relative Enc (0-1)

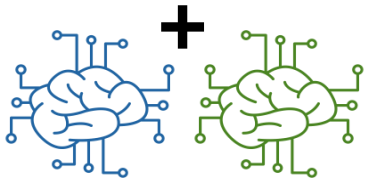
Core Idea: Composable Pre-trained Models

How to play Lego?

Key Idea 2 - Play Lego:
Combine Experts +
A Shallow Model
(Downstream Task)

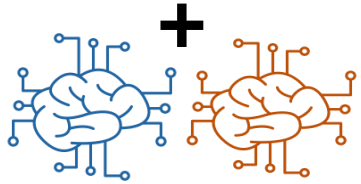
Predicted Cardinality

Simple Regression Model
for CE on TPC-H



Predicted Cost

Simple Regression Model
for Cost Est. on IMDB



Low overhead for down-stream model

Pre-trained experts (no overhead)

Key Idea 1 – Decompose:
Small Pre-trained Experts
(Task- & Dataset-independent)

Small Pre-trained Experts

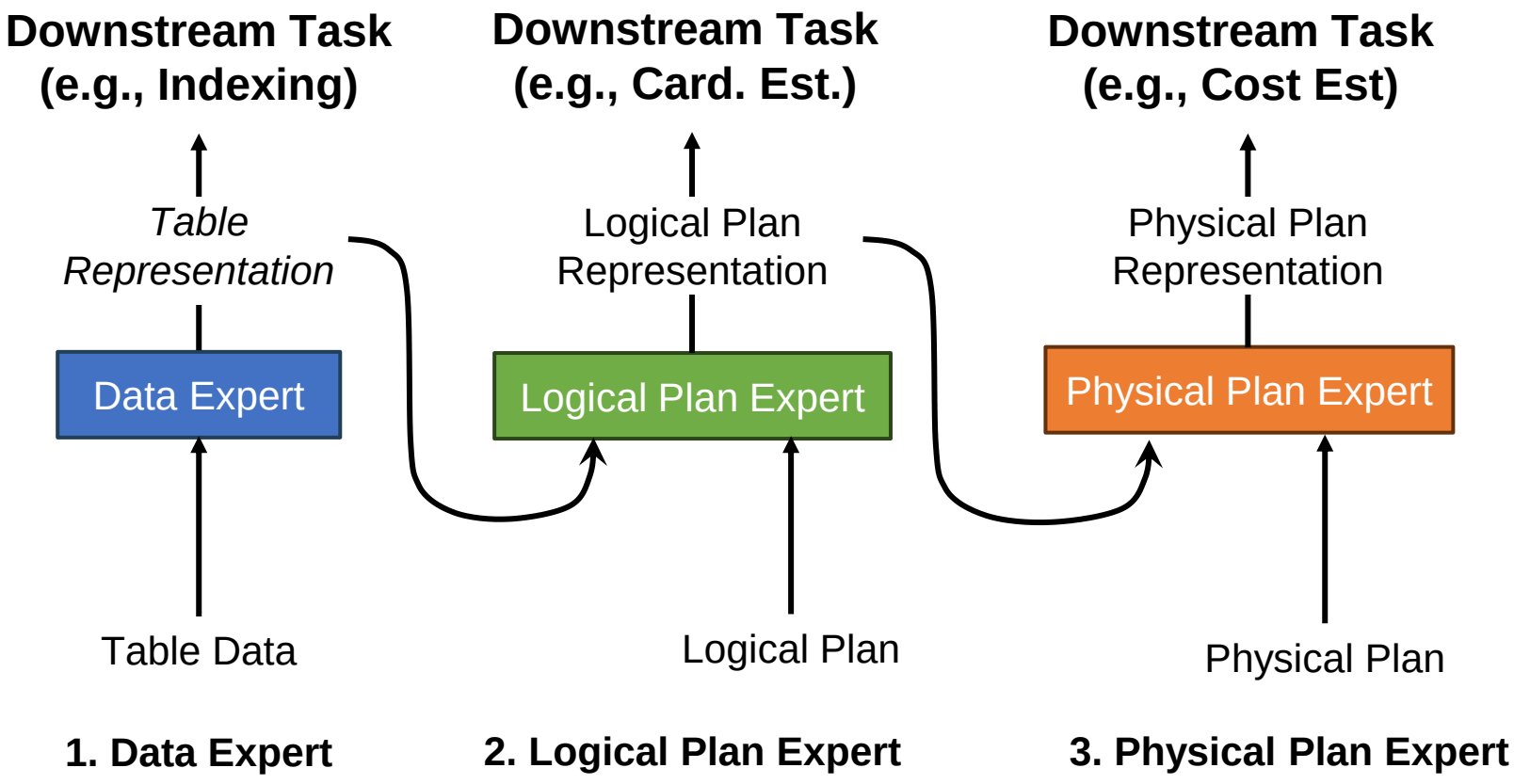
Data Expert

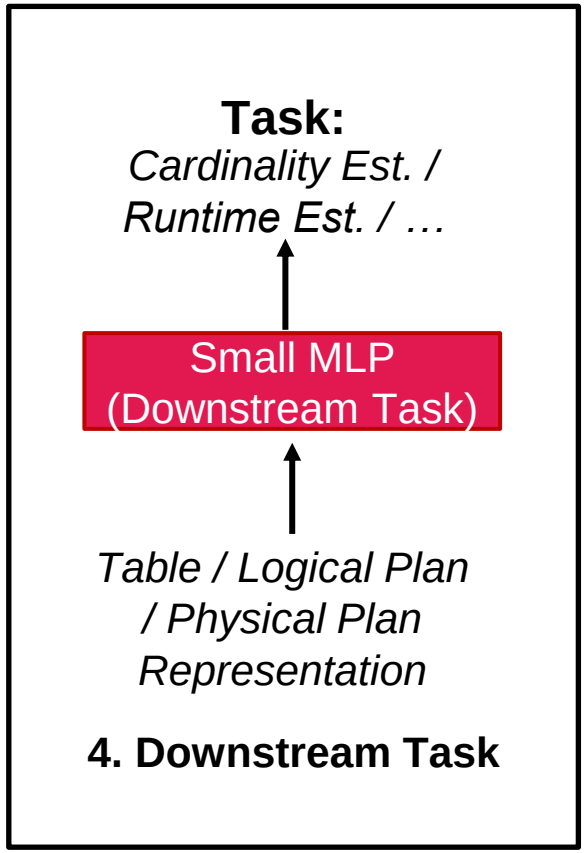
Logical Plan Expert

Physical Plan Expert

Other Experts

How to combine Experts for Tasks?





Experts are trained & used in a "stacked" manner
to solve downstream tasks

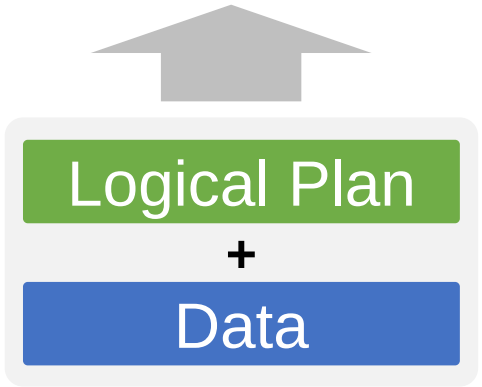
Initial Evaluation & What's Next?

Does it work? Initial Evaluation

… more tasks in the paper (e.g., AQP)

Evaluation on 19 real-world datasets

Task 1: Card. Est (CE)



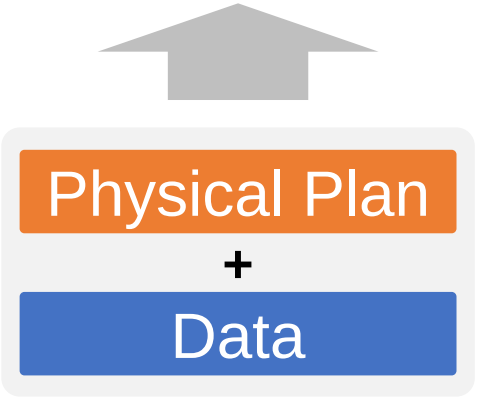
CE across data sets

FT = Fine-tuned on one data set

Instance-Specific Models

	Ours	Ours (FT)	PG	DeepDB	MSCN
Q-error (Median)	2.12	1.69	1.98	1.83	1.68
Q-error (95th)	92.92	26.08	294.15	152.23	3120

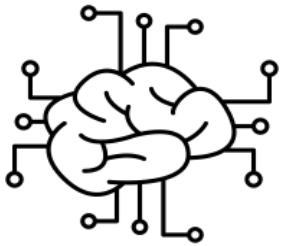
Task 2: Runtime Est



Multi-Dataset Model

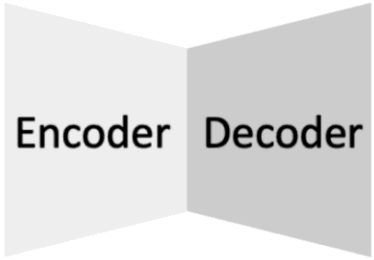
	Ours	Ours (FT)	ZS Cost	PG Cost
Q-error (Median)	1.87	1.5	1.08	6.44
Q-error (95th)	10.76	6.83	1.7	19.73

Future Directions: This is just the beginning



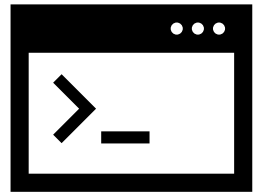
More Tasks / Experts

Hardware Expert (e.g. to enable cost prediction across hardware)



Self-Supervised Expert Training

Learning representations with Auto-Encoder model



Beyond Database Systems

Foundation *System* Models (OSs, ML Systems e.g. Tensorflow)