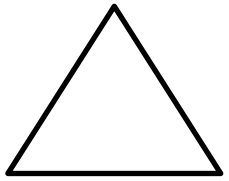
# The Case for Multi-Task Zero-Shot Learning

Johannes Wehrstein, Benjamin Hilprecht, Benjamin Olt,
Manisha Luthra, Carsten Binnig

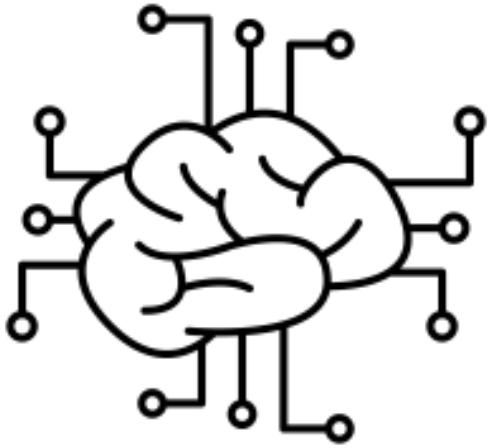
Technical University Darmstadt, Germany

DATA MANAGEMENT
TU DARMSTADT

# Learned DBMS Components



Replace



**Classical (Hand-tuned) DBMS Components**

**Learned DBMS Components**

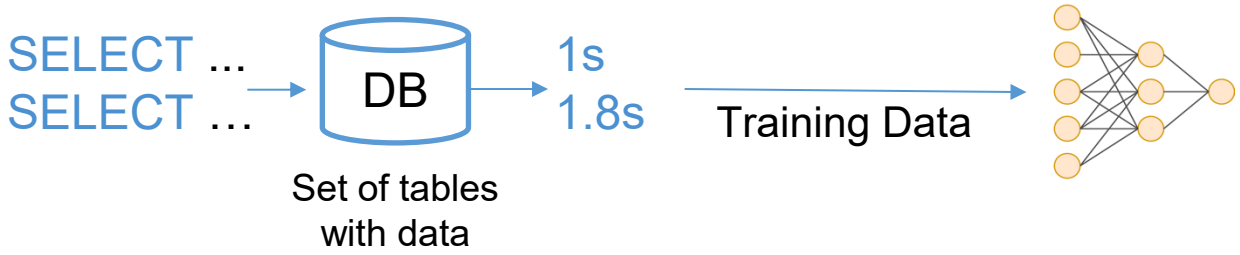
**High Potentials:** Learned models can handle complexity very well and reduce manual engineering/adaption effort at the same time

# Learned DBMS Components: State-of-the-Art

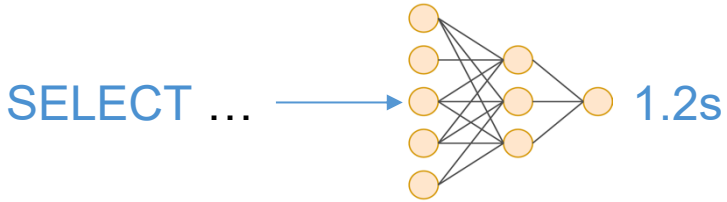
**The Predominant Approach:** Workload-Driven Learning
(e.g. MSCN (Kipf et al.) / NEO (Marcus et al.))

**Example:** Cost Estimation – estimate the cost of a query execution

**Training** 1) Run Workload: 2) Observe Behavior (e.g., Runtime of Queries) 3) Train Model:




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



# Issues of Workload-Driven Learning

**Workload-Driven Learning is highly unattractive in practice**

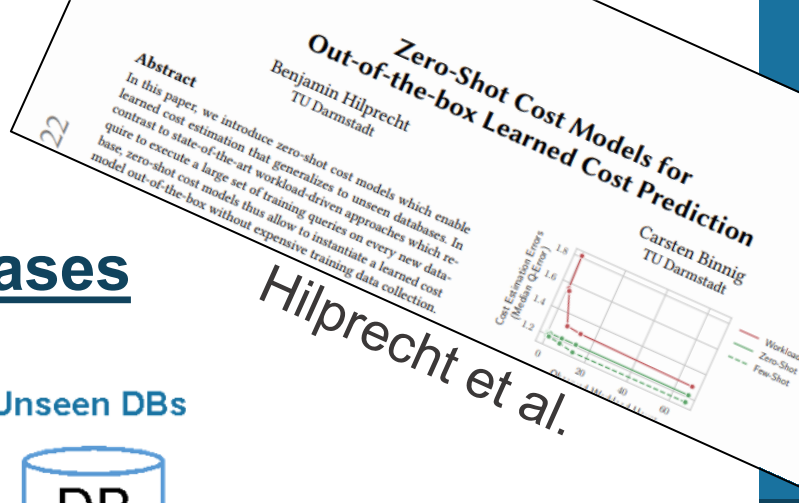
👎 **High efforts for training data collection**> 10's of thousands of training queries needed → might take hours / days to execute

👎 **Needs to be repeated** for every new database (i.e., new set of tables + data) or even if data is updated

**Zero-Shot Learning: Generalize over unseen databases**

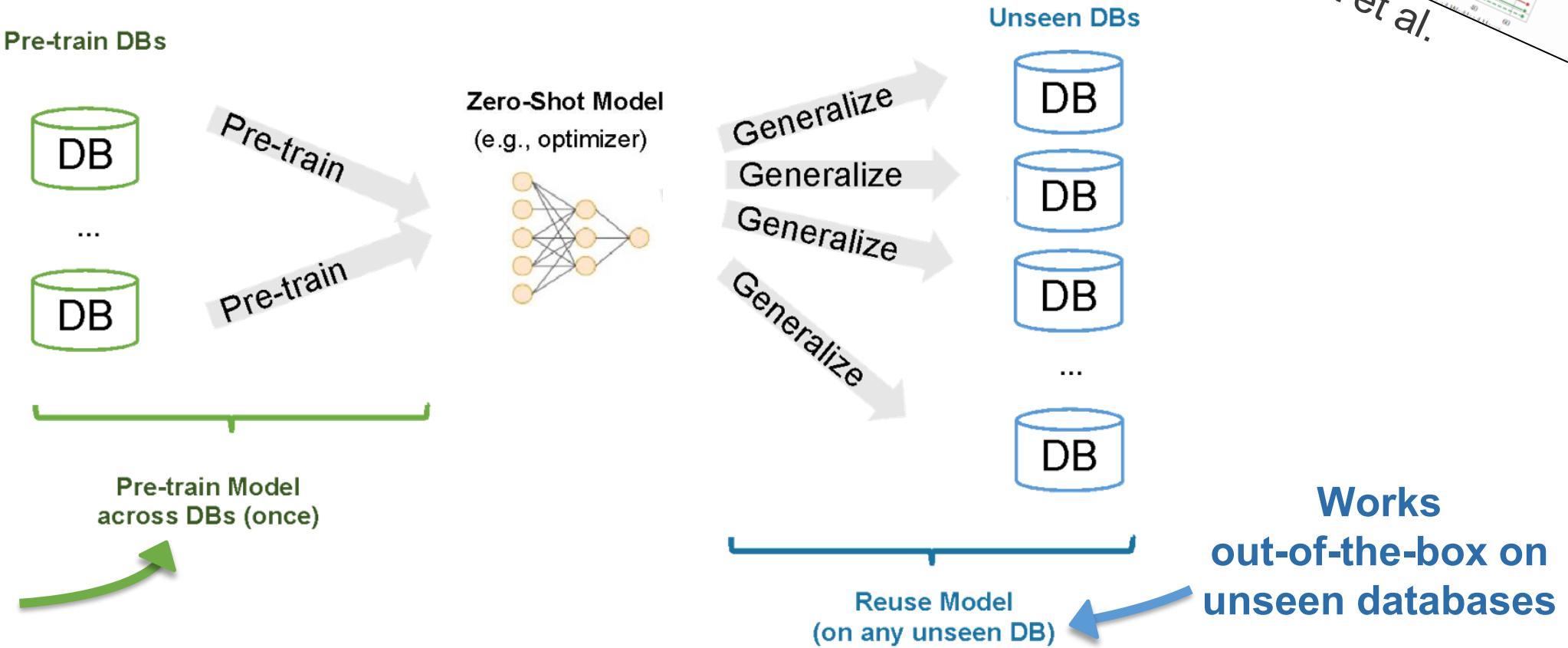
→ Works Out-of-the-Box on unseen databases

# Zero-Shot Cost Estimation




Hilprecht et al.

**Goal: Learn models that <u>generalize to unseen databases</u> with no overhead**




**One-time effort**

**Works out-of-the-box on unseen databases**

**Hilprecht et al. (VLDB 2022 - Session 12)**

# Zero-Shot Learning

What has been shown:

- Zero-Shot Learning works well for cost estimation (Hilprecht et al.)

Benefits:

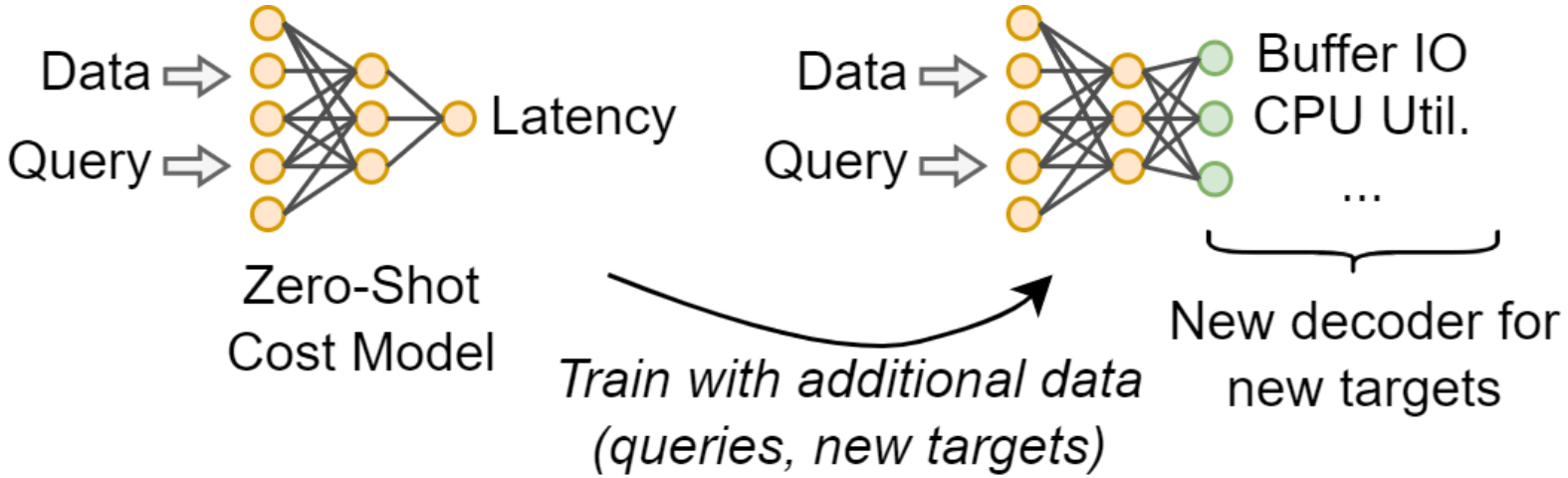
- 👍 High Accuracy
- 👍 Generalization over databases
- 👍 Training is one-time effort (not again for every new database)

**But: Zero-Shot Learning is limited to Cost-Estimation so far**

**Hilprecht et al. (VLDB 2022 - Session 12)**

# Direction #1: Fine-Tuning to New Tasks

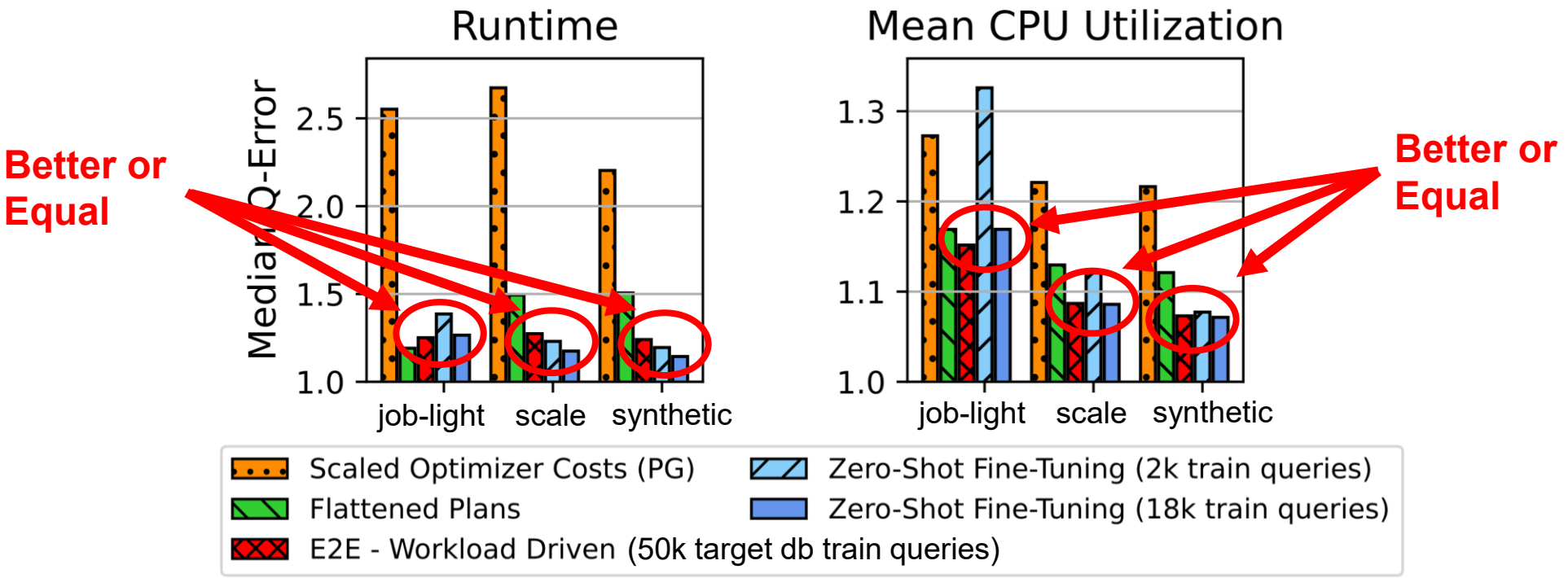
**Key Idea: Further train existing zero-shot model for different task**




- 👍 **Reduced training time:** finetuning ~2x faster than training from scratch
- 👍 **Very simple:** no changes to the model architecture
- 👎 **Limited to related & regression tasks:** due to simplicity cannot be adapted to non-related and non-regression tasks

# Finetuning Case Study: Resource Est.

**Task: Resource Estimation** (Runtime, Mean CPU Utilization)




*Metrics are predicted with very high accuracy on unseen databases.*

- Limited to related regression tasks

# Extending Zero-Shot Learning

**Main directions to extend zero-shot models to further tasks:**

1. **Fine-tuning to new Tasks**
   - Applicable for related regression tasks
   - But what about other tasks?

**2. Combination with Optimization Procedures**

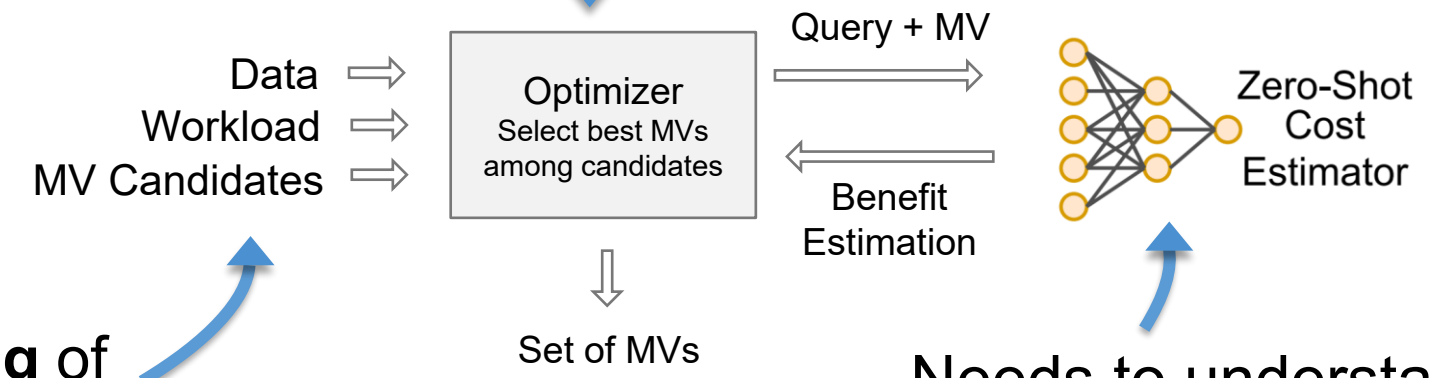
**Optimization Problem:**

- *Example*: Select “best” MVs among all MV candidates
- Requires accurate benefit estimation: *“What-If we create this MV?”*
  → Can be estimated using zero-shot cost estimators

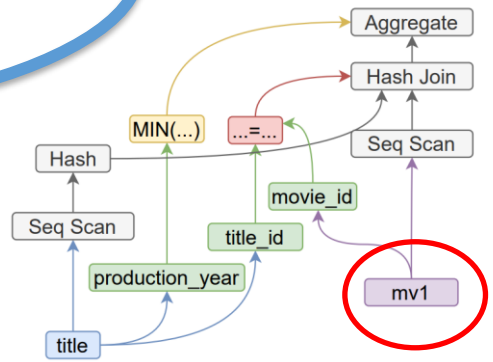
# Direction #2: Combination with Optimization

**Key Idea: Include zero-shot cost models in optimization approach to effectively guide optimizer to pick best MVs.**

**ILP Solver** guided by benefit estimates (works out-of-the-box on any db)




ZS model extended to predict cost of **plans which include MVs**




**Encoding** of search space (i.e. MV Candidates)

Needs to understand the **effects** of MVs on **different databases**

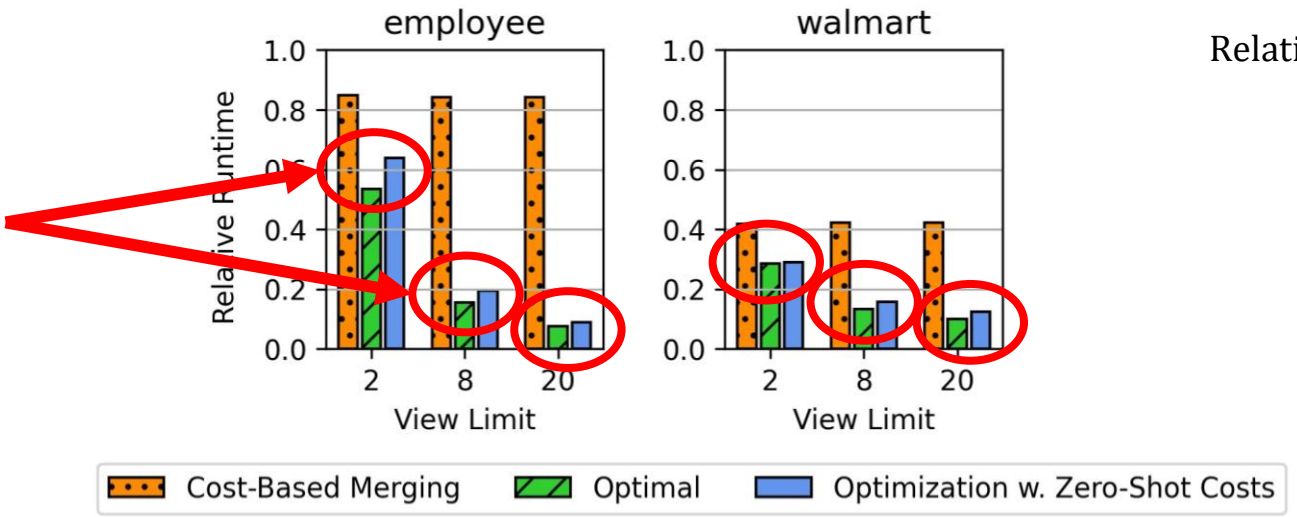
# Direction #2: Case Study – MV Selection

## Task: Materialized View Selection

*(View limit = max. number of created MVs, ensures certain level of generalization)*

$$\text{Relative runtime} = \frac{runtime\ w.MVs}{runtime\ wo.MVs}$$

**Better than cost-based merging + close to optimal performance**




*(non-learned approach used today in a commercial DBMS)*

*Approach shows <u>robust performance</u> over all databases clearly outperforming cost-based merging approach.*

# Direction #2: Combination with Optimization

**Summary:**

- 👍 **Applicability for optimization tasks:** many of most complex DBMS tasks can be cast as optimization problems
- 👍 **Transfers across databases:** works out-of-the-box on unseen databases
- 👎 **Scalability limited by optimization problem solver**: usual issue for optimization problems
- 👎 **Optimization Problem solvers often prune search space:** e.g. using heuristics to cut down search space → leads to errors
- 👎 **Errors add up**: errors in the cost estimation model + optimizer not totally accurate

# Extending Zero-Shot Learning

**Main directions to extend zero-shot models to further tasks:**

1. **Fine-tuning to new Tasks**

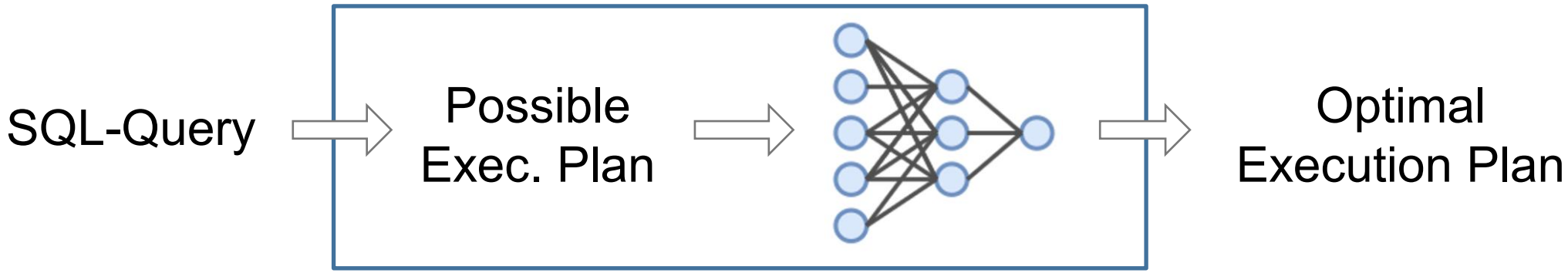
2. **Combination with Optimization Procedures**
   - Limited in scalability
   - Errors add up: errors in the cost estimation model + optimizer not totally accurate
   - Can we directly learn what we aim for?

3. **End-to-End Learning**
   - End-to-End trained model which solves high-level database tasks directly for any database

# End-to-End Learning

Goal: **end-to-end zero-shot model** that works **out-of-the-box** on unseen DBs

Example Task: Query Optimization



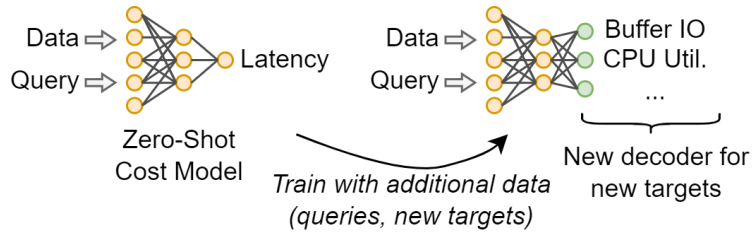

Challenges:

- **Representation of the design space**: How to encode and pass the search space to the model
- **Training objective:** model how efficient a point in the search space is

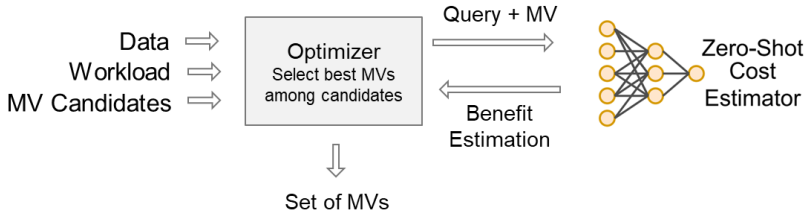
# Conclusion / Summary

*Extension of zero-shot models beyond cost-estimation is possible:*

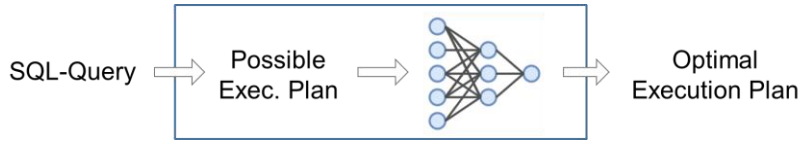
- **Finetuning to new Tasks**
  → *shown for resource estimation*




- **Combination with optimization**
  → *shown for MV selection*




- **End-To-End Learning**
  → *currently working on*



# Thanks for your Attention