

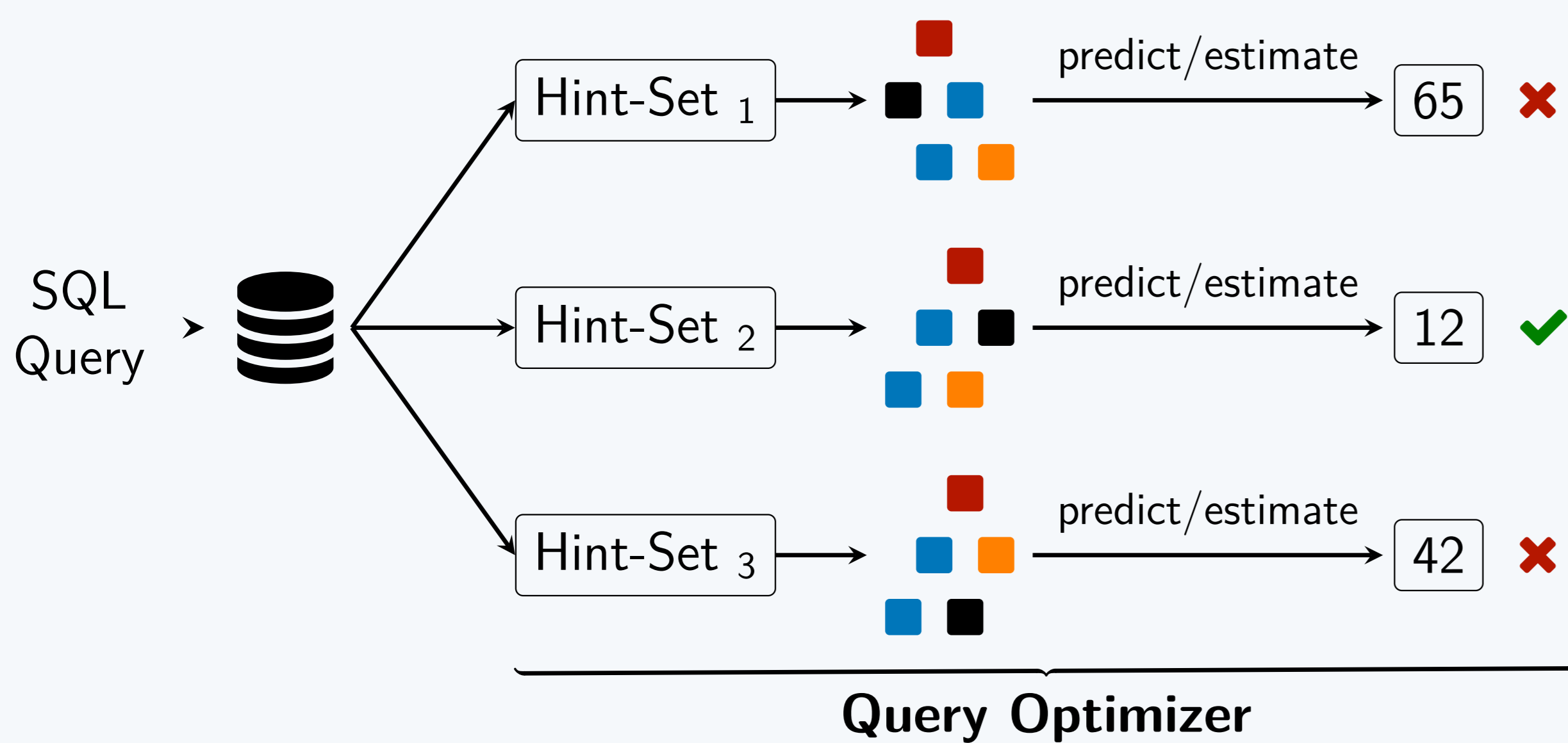
# AutoSteer: Learned Query Optimization for Any SQL Database

Christoph Anneser<sup>†</sup> Nesime Tatbul<sup>‡</sup> David Cohen<sup>°</sup> Zhenggang Xu<sup>∞</sup>

Prithviraj Pandian<sup>∞</sup> Nikolay Laptev<sup>∞</sup> Ryan Marcus<sup>◇</sup>

<sup>†</sup>Technical University of Munich <sup>°</sup>Intel <sup>‡</sup>MIT <sup>∞</sup>Meta <sup>◇</sup>University of Pennsylvania

## Background – Steered Query Optimizers



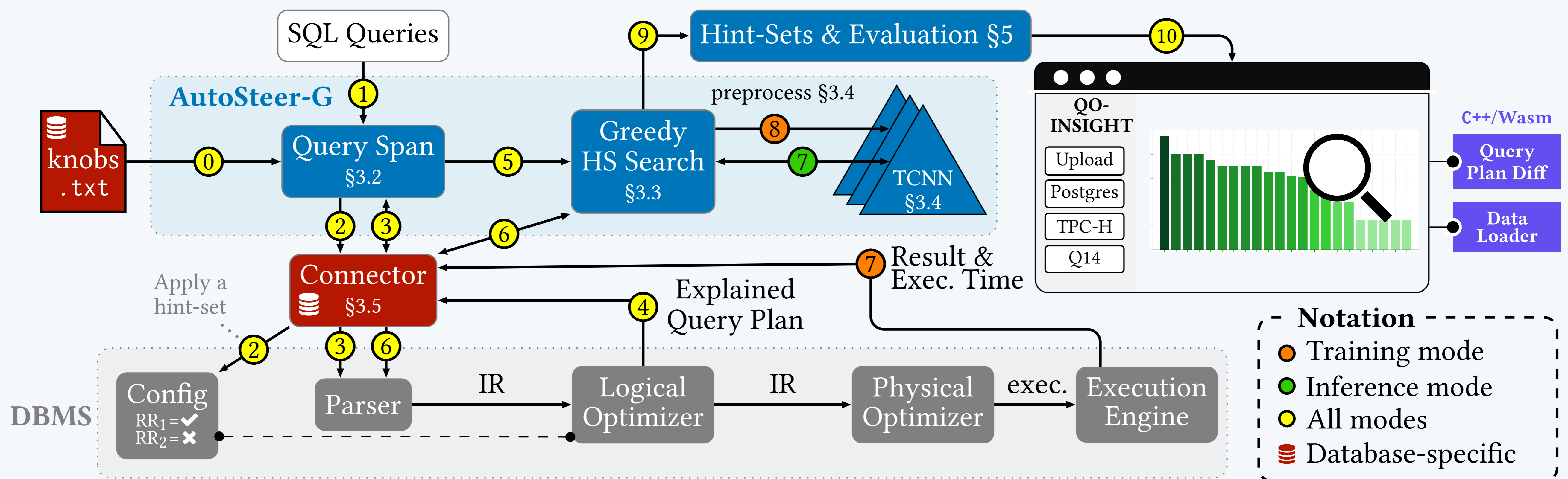
- ▶ Database Systems **expose knobs** that can be used to steer query execution. For example, PostgreSQL has knobs to disable nested loop joins or index scans.
- ▶ **Hint-sets (HS)** can combine multiple knobs. For example: `{indexscan:false, nestloop:false}`
- ▶ Recent work on steered query optimizers either predefines [1] or randomly chooses [2, 3] multiple hint-set, which are used to **generate alternative query plans**.
- ▶ A **deep neural network predicts** the execution time of each plan.

[1] Marcus et al.: “Bao: Making Learned Query Optimization Practical” (SIGMOD’21)

[2] Negi et al.: “Steering Query Optimizers: A Practical Take on Big Data Workloads” (SIGMOD’21)

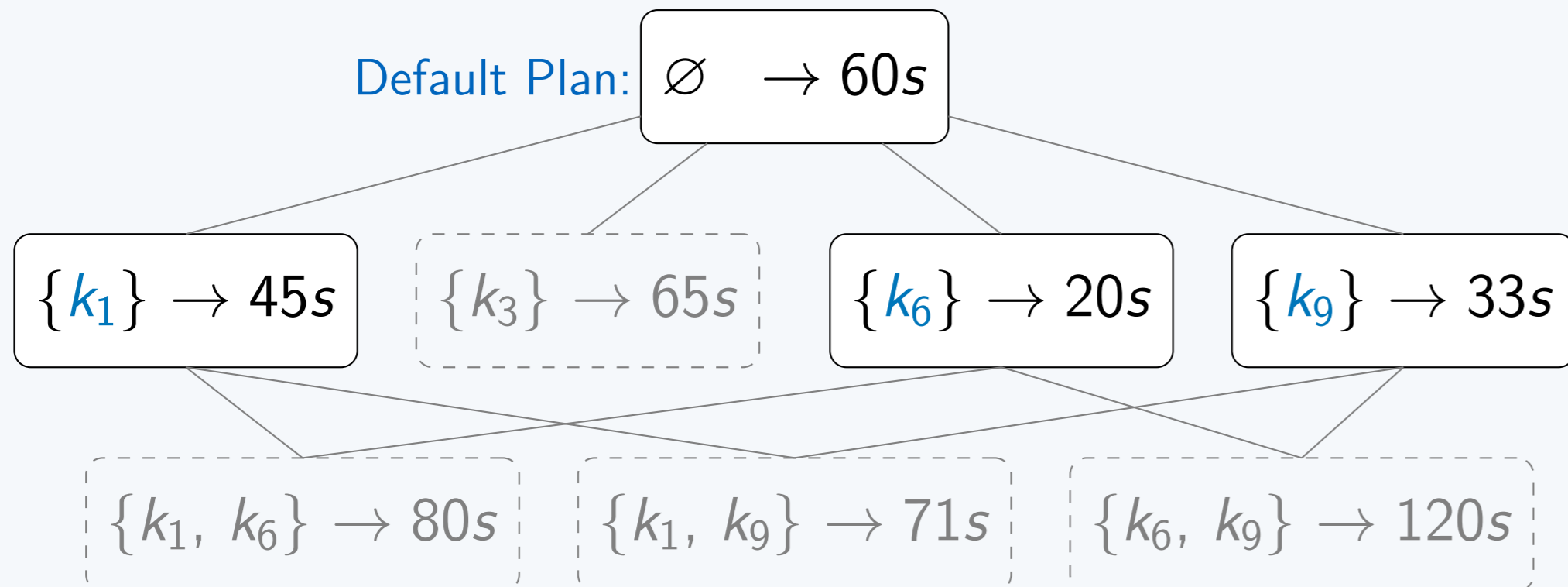
[3] Zhang et al.: “Deploying a Steered Query Optimizer in Production at Microsoft” (SIGMOD’22)

## System Overview



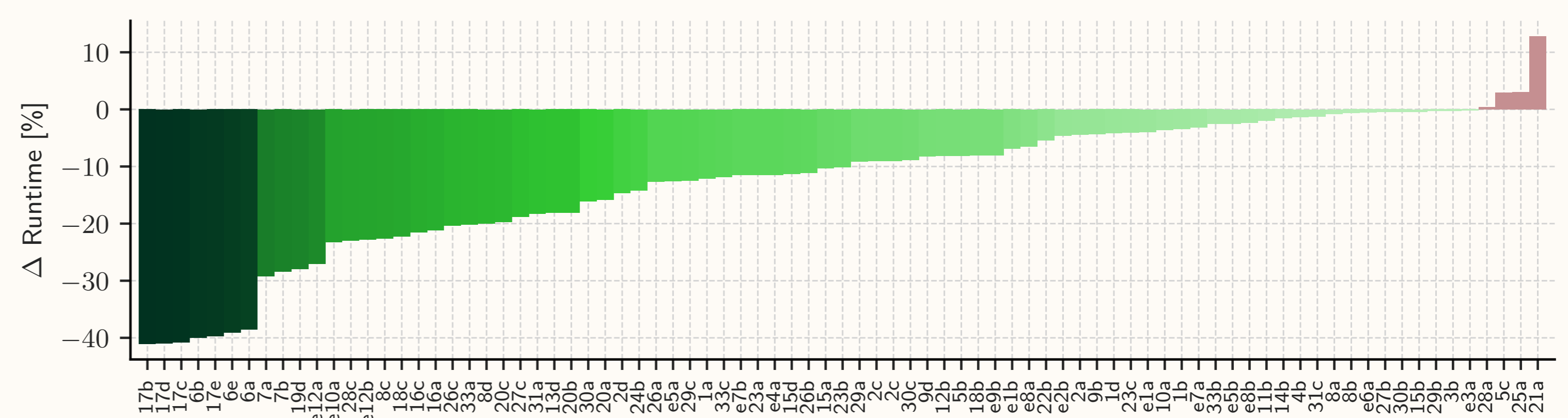
## Greedy Hint-Set Exploration

- ▶  $2^{\#\text{knobs}}$  different hint-sets; However, most yield *bad plans*.
- ▶ Greedy search finds promising hint-sets with reasonable overhead.
- ▶ **Assumption:**  
Larger, beneficial HSs consist of smaller, beneficial HSs.  
Not always, but in many cases true. *Experimentally tested* ✓
- ▶ **Input:** SQL query and query span (example:  $[k_1, k_3, k_6, k_9]$ )



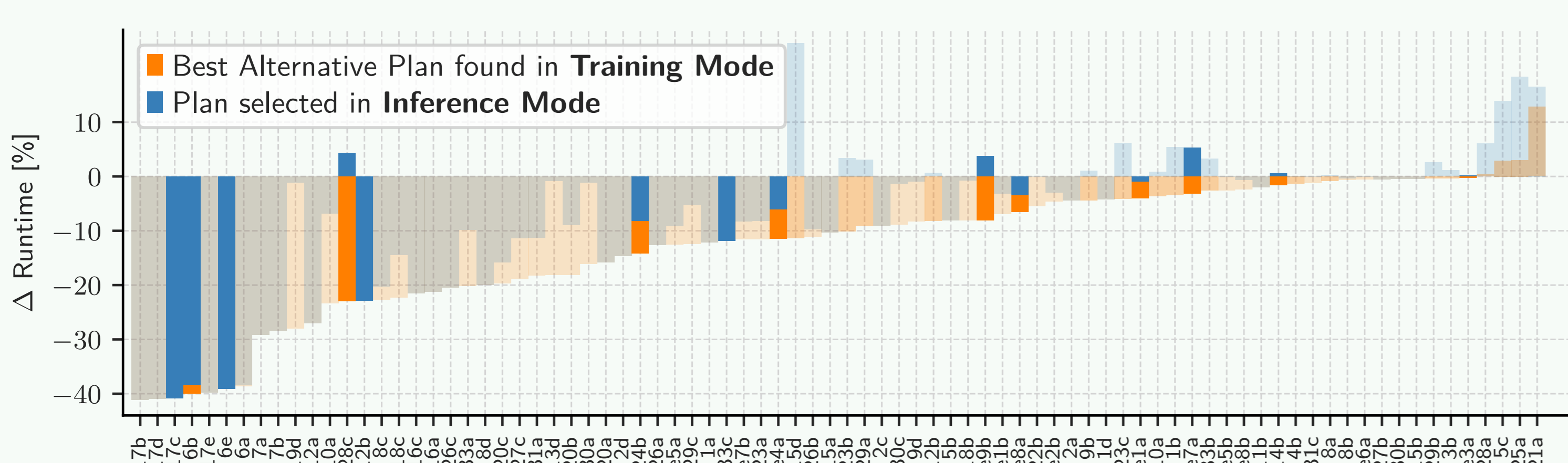
## Training Mode (PrestoDB/Join Order Benchmark)

- ▶ AutoSteer with an integrated database connector for PrestoDB.
- ▶ For 137 queries, AutoSteer explored **1730 different hint-sets**.
- ▶ Found between 8 and 34 different plans per query.
- ▶ Performance improvements of **up to 40%** per query.



## Inference Mode (PrestoDB/Join Order Benchmark)

- ▶ AutoSteer with an integrated database connector for PrestoDB.
- ▶ AutoSteer using a tree convolutional neural network to **infer execution times** at runtime.
- ▶ **Reduces execution times** of unseen queries by **20.6%** (opaque) and seen queries by **26.8%** (transparent).



## Dashboard Application at Meta (PrestoDB)

- ▶ Focus on **tail latencies**.
- ▶ **>3000 queries** scanning **petabytes of data**.
- ▶ Workload runs every day on a PrestoDB cluster having **hundreds of compute nodes**.

