Introduction

Each database running a different workload, demands different resources and database configuration settings to achieve optimal performance, which prompts us to study workload features in detail.

We define a database workload as

\[ W = \left\{ (p_1, \theta_1), (p_2, \theta_2), \ldots, (p_m, \theta_m) \right\} \]

where \( p_i \) is the database query-plan, and \( \theta_i \) is a normalized weight of importance of \( p_i \) in workload \( W \). For understanding workloads comprehensively it is necessary to perform feature engineering on query plans.

Key Contributions

1. We propose query plan encoder models capturing structure and computational performance resource requisites as distributed feature representations.
2. We keep structure, and computational performance representation separate that enables downstream tasks to weigh each representation independently in their model.
3. We propose a taxonomy for operator types for learning diverse structure of query plans with self-attentive transformers.
4. We find performance of query plans are best characterized by encoders when plan task nodes are classified under scan, join, sort and aggregate; each having an encoder of its type.
5. Latency prediction and query classification downstream tasks performing well with our pretrained encoders suggests efficacy of our modeling strategy.
6. In depth domain adaptation evaluation and ablation studies on various datasets signifies pretrained encoders adapts to new domain quickly, whereas encoders trained from scratch overfits.
7. In this work, we open-sourced an automated workload execution tool for cloud, a crowd-sourced plan dataset and revised two spatial benchmarks.

Plan Encoders

![Fig 2. Structure Plan Encoder Modeling.](image)

![Fig 3. Computational Performance Encoder Modeling.](image)

Downstream Task Modeling

![Fig 4. A bird-view architecture diagram, showing the role of plan encoders for downstream tasks. For example, latency prediction and query classification tasks.](image)

Results

![Fig 5. Blue bars are median query latency, Orange lines are 5th-95th percentile range variations, and mean abs. error marked with black bar for spatial queries. A smaller black bar on a larger orange-line bar means better results.](image)

![Fig 6. Results of structure encoder domain adaptation analysis on TPC-H, TPC-DS, and SPATIAL datasets. Notations: Scratch is Untrained encoder weights initialized; Fixed is Pretrained encoder weights freeze. Fine is Pretrained+Finetuned Encoder.](image)