Facilitating SQL Query Composition and Analysis

Zainab Zolaktaf, Mostafa Milani, Rachel Pottinger

Department of Computer Science

University of British Columbia

Motivation: Facilitating SQL Query Composition and Analysis

- SQL query composition can be fundamentally difficult for users
  - Requires several cycles of tuning and execution of costly queries

- To write efficient SQL queries users can
  - Gain knowledge of database schema and tuples
  - Use hints or tutorials available on the system
    - E.g., On SDSS users are advised to write a ``Count'' query first!

- Our goal: predict SQL query performance properties - prior to execution

Motivation: Facilitating SQL Query Composition and Analysis

A new SQL query

$\boldsymbol{Q_*}, \boldsymbol{y_*} = ?$

```
SELECT q.name AS qname,
       dbo.fDistanceArcMinEq(q.ra,q.dec,p.ra,p.dec), ...
FROM SpecObj AS s,
     SDSSSQL010.MYDB_670681563.test.QSOQuery1_DR5 AS q, PhotoObj
      AS p
WHERE ((s.bestobjid=p.objid) AND (s.ra BETWEEN 185 AND 190) AND
      ...) ORDER BY q.ra
```

Goal

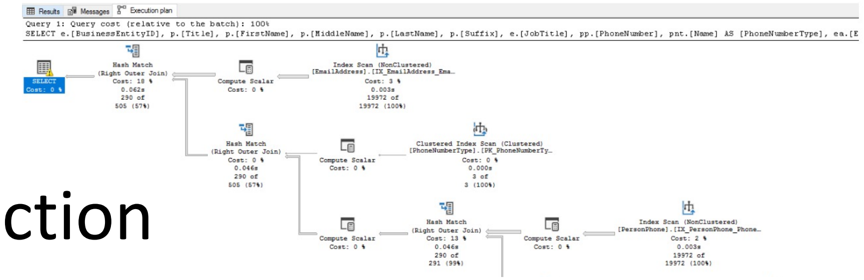
Predict performance properties of Q_*, prior to submitting it the database

Output

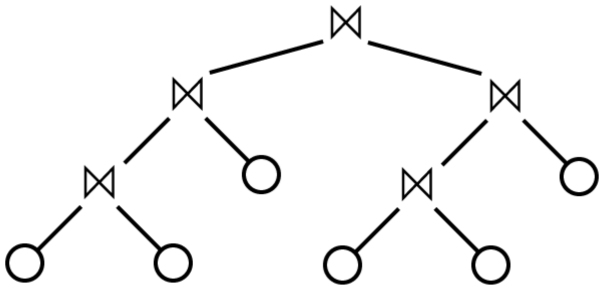
- Answer size (y_*^a) = 304 rows
- CPU time (y_*^c) = 105.37 sec
- Error class (y_*^e) = success
- Session class (y_*^s) = browser

Goal
- Predict performance properties of Q_*, prior to submitting it the database

Challenges: Database Instance



- Existing models for query performance prediction
  - System side applications (e.g., admission control, query optimization [LKNC12])
- Use query execution plan
  - Need database instance and statistics



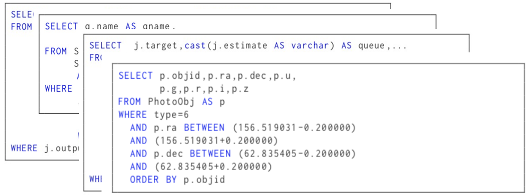
- Problems
  - Query execution plan can be imprecise [LGMB15]
  - Limited access to database instance?
    - Sources on the hidden web
    - Customers of cloud data warehouses
    - Spotify, HSBC use Google BigQuery

Output

- Cardinality estimates?
- Cost estimates?
- Error class (y_*^e) = success
- Session class (y_*^s) = browser

Challenges: Large-scale Query Workloads

$$W = \{(Q_i, y_i)\}_{i=1}^{n}$$



1. Sloan Digital Sky Server (SDSS) [RTS14]
   - Scientific computing domain
   - Extracted ~600K SQL queries
2. SQLShare [JMH16]
   - SQL-as-a-Service platform
   - Users upload data, write queries
   - Contains ~27K SQL queries

- SQL Query workload (W)
  - Collection of labeled SQL queries submitted in the past
  - Labels are actual observations
    - Eliminate biases e.g., cardinality misestimates
  - Easily logged by DBMS

- Need large-scale and real-world query workloads
  - Reveal usage patterns from a variety of users

Problem Formulation: Facilitating SQL Query Composition and Analysis

Collection of labeled SQL queries

$$W = \{(Q_i, y_i)\}_{i=1}^{n}$$

```sql
SELECT  j.target,cast(j.estimate AS varchar) AS queue,...
```

```sql
SELECT p.objid,p.ra,p.dec,p.u,
       p.g,p.r,p.i,p.z
FROM PhotoObj AS p
WHERE type=6
  AND p.ra BETWEEN (156.519031-0.200000)
  AND (156.519031+0.200000)
  AND p.dec BETWEEN (62.835405-0.200000)
  AND (62.835405+0.200000)
  ORDER BY p.objid
```

A new SQL query

$$Q_*, y_* = ?$$

```sql
SELECT q.name AS qname,
       dbo.fDistanceArcMinEq(q.ra,q.dec,p.ra,p.dec), ...
FROM SpecObj AS s,
     SDSSSQL010.MYDB_670681563.test.QSOQuery1_DR5 AS q, PhotoObj
      AS p
WHERE ((s.bestobjid=p.objid) AND (s.ra BETWEEN 185 AND 190) AND
      ...) ORDER BY q.ra
```

Goal

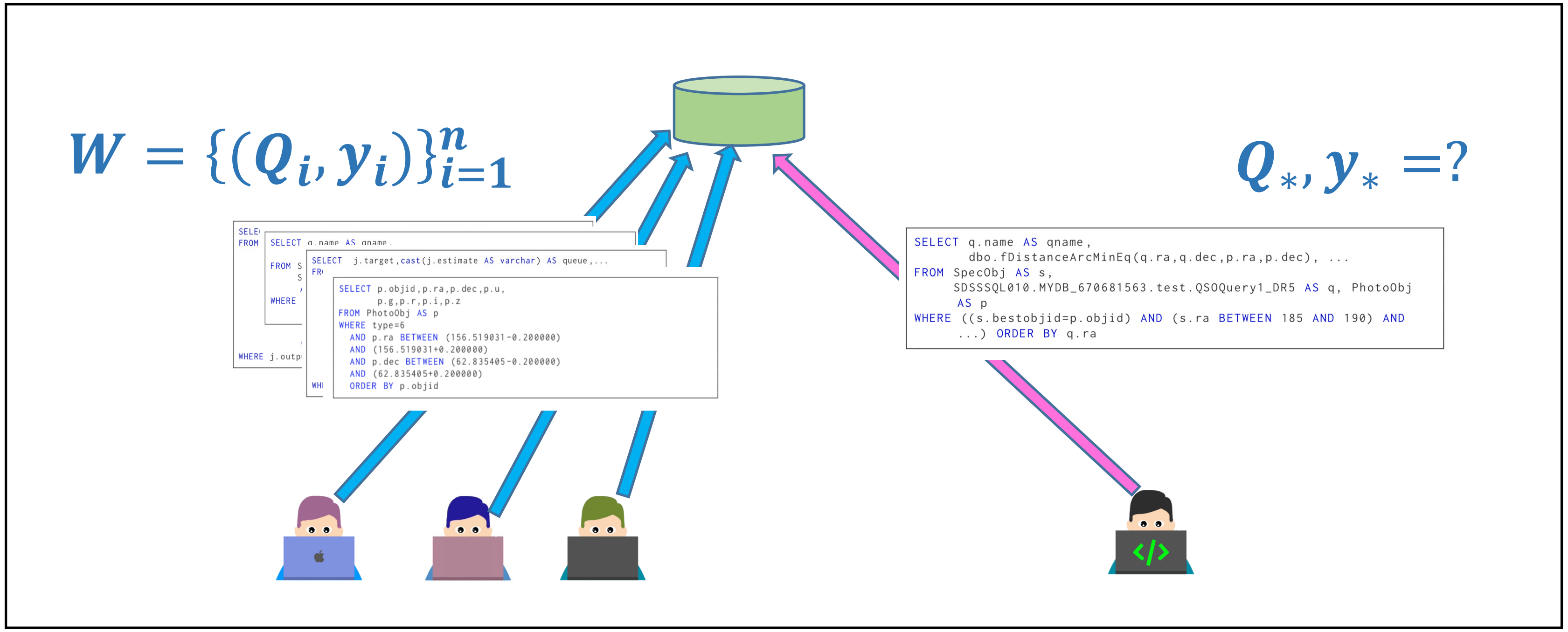
Predict performance properties of Q_*, prior to submitting it the database

Output

- Answer size (y_*^a) = 304 rows
- CPU time (y_*^c) = 105.37 sec
- Error class (y_*^e) = success
- Session class (y_*^s) = browser

Goal
- Predict performance properties of Q_*, prior to submitting it the database

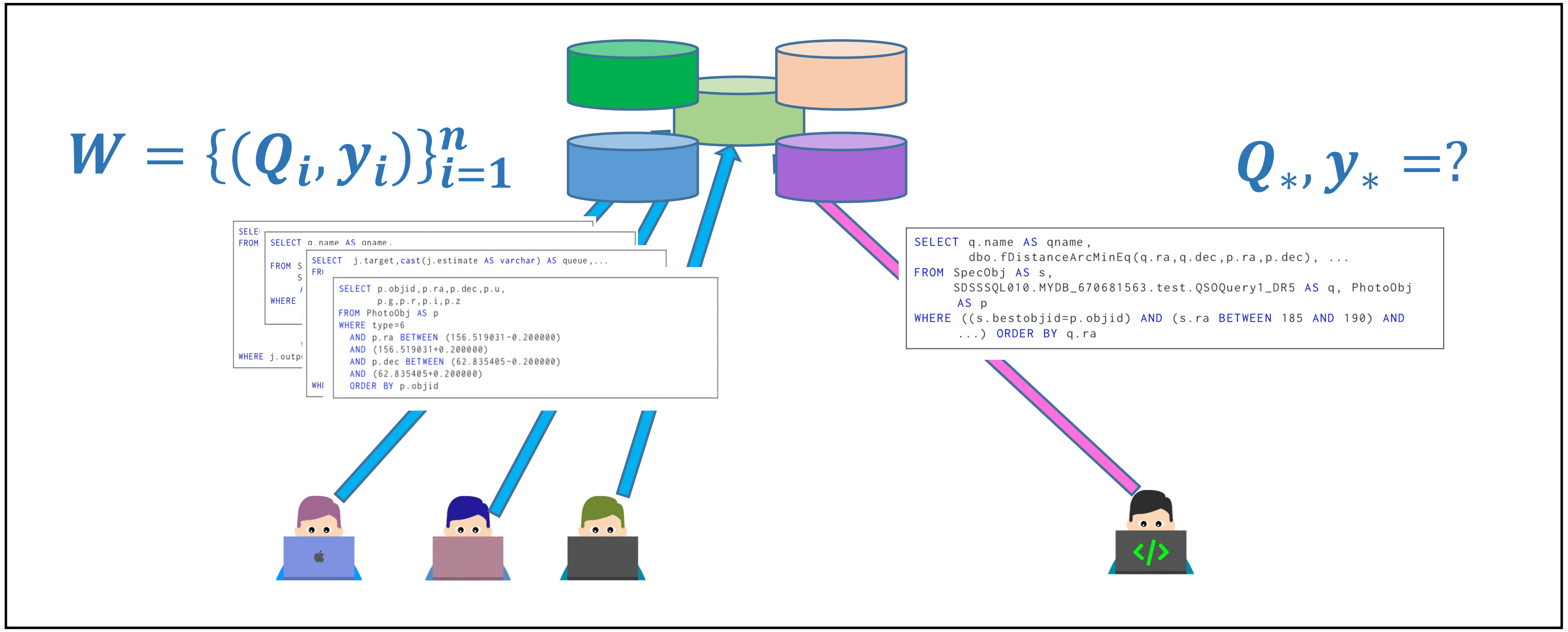
Approach Overview: Different Settings



SDSS

1. Homogeneous Instance: Q_* and the queries in W are posed to the same database instance

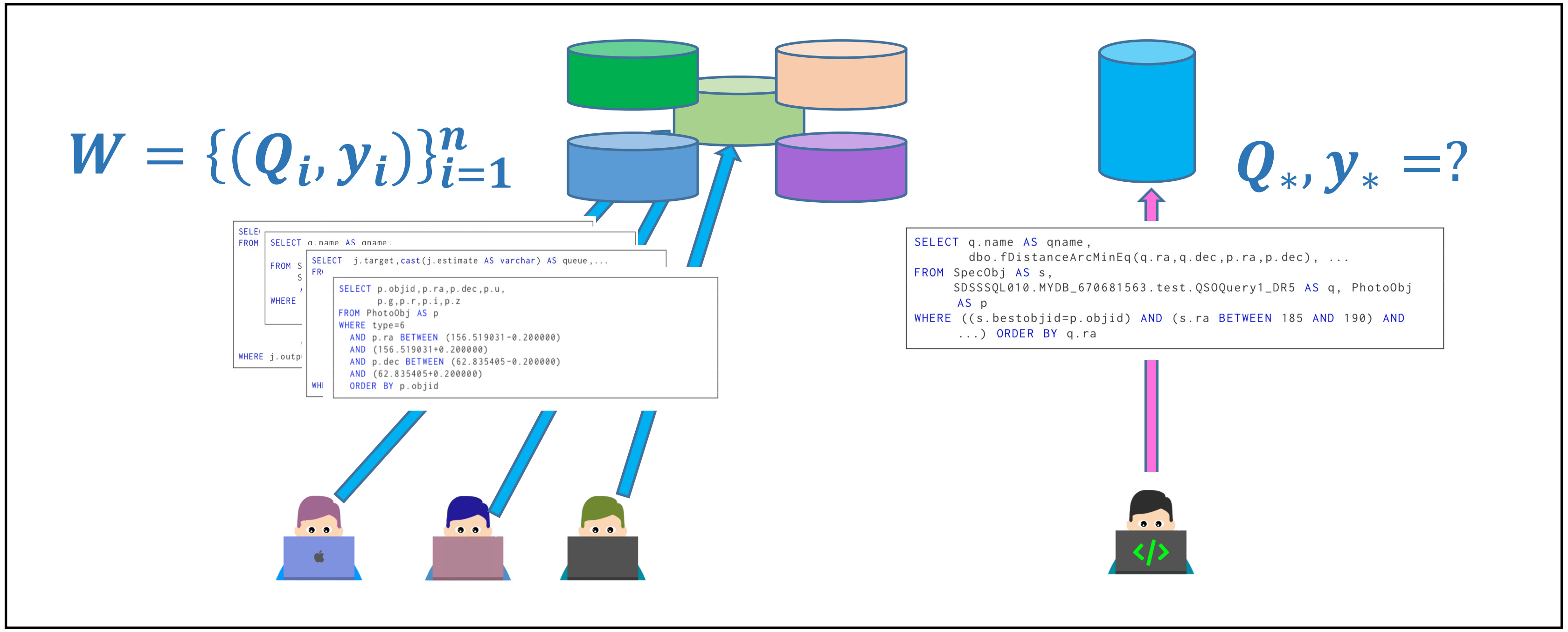
Approach Overview: Different Settings



SQLShare

2. Homogeneous Schema: Q_* and the queries in W are posed to different database instances with the same schema in the same DBMS

Approach Overview: Different Settings



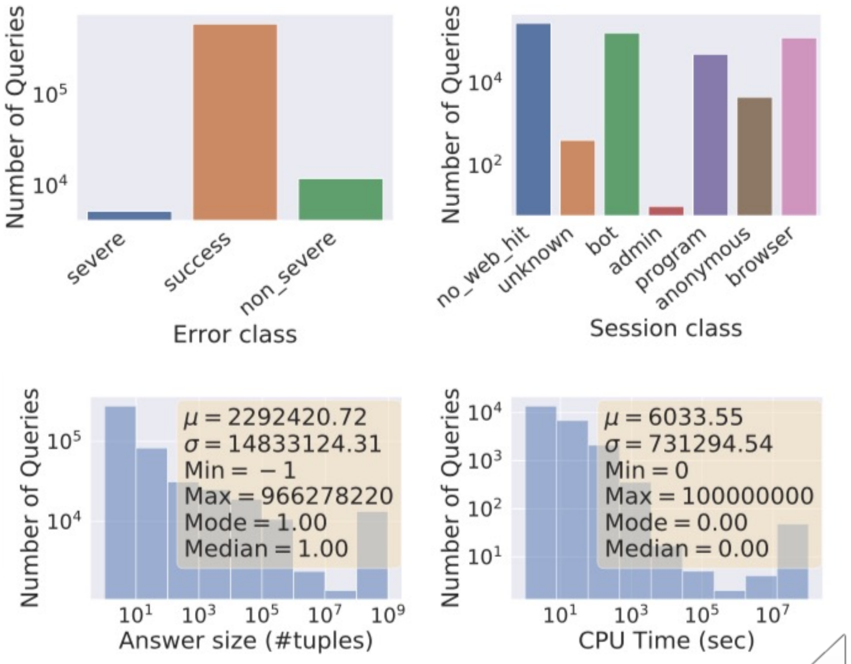
SQLShare

3. Heterogeneous Schema: Q_* and the queries in W are posed to different databases with different schemas that run in the same DBMS

Approach Overview: Workload Analysis

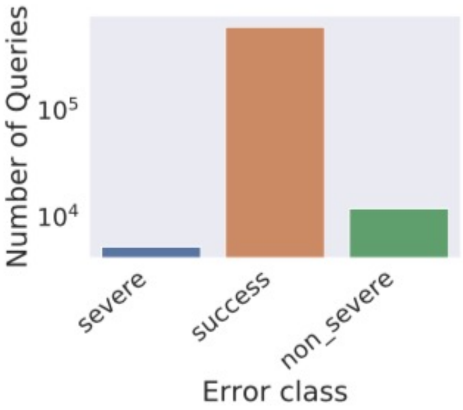
- Perform workload analysis for
  - Better model selection
  - Better model evaluation
- SQL query statements
  - Digits and mathematical equations in statements
    - Affect query performance, e.g., answer size
  - Range in complexity w.r.t. length, #joins
- SQL query labels
  - Classification labels are imbalanced
  - Regression labels had a wide range

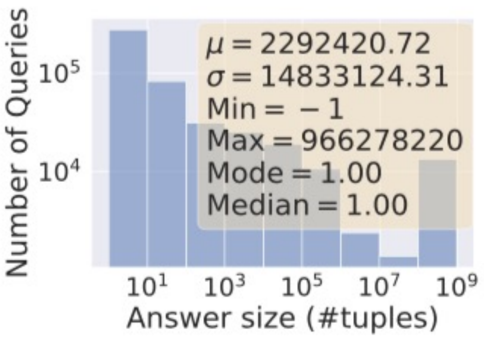
```
SELECT q.name AS qname,
       dbo.fDistanceArcMinEq(q.ra,q.dec,p.ra,p.dec), ...
FROM SpecObj AS s,
     SDSSSQL010.MYDB_670681563.test.QSOQuery1_DR5 AS q, PhotoObj
      AS p
WHERE ((s.bestobjid=p.objid) AND (s.ra BETWEEN 185 AND 190) AND
      ...) ORDER BY q.ra
```



Approach Overview: Models Evaluated

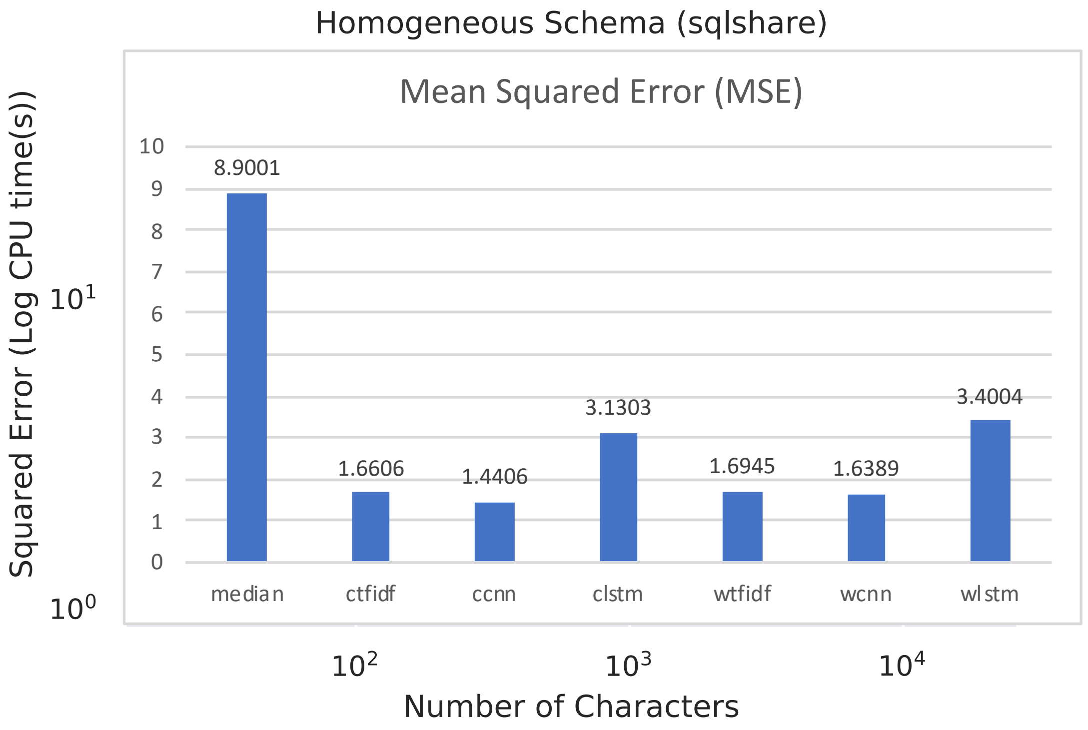
- To establish baselines we examined a broad set of models
  - Models that do not consider SQL query statement
    - Most frequent class (mfreq) classifier
    - Median of distribution for regression
  - Models that do consider SQL query statement
  - Query statement representation?
    - Bag-of-n-grams + TFIDF
    - Shallow Convolutional Neural Network (CNN)
    - 3-Layer Long Short-Term Memory (LSTM)
  - Applied at character and word level



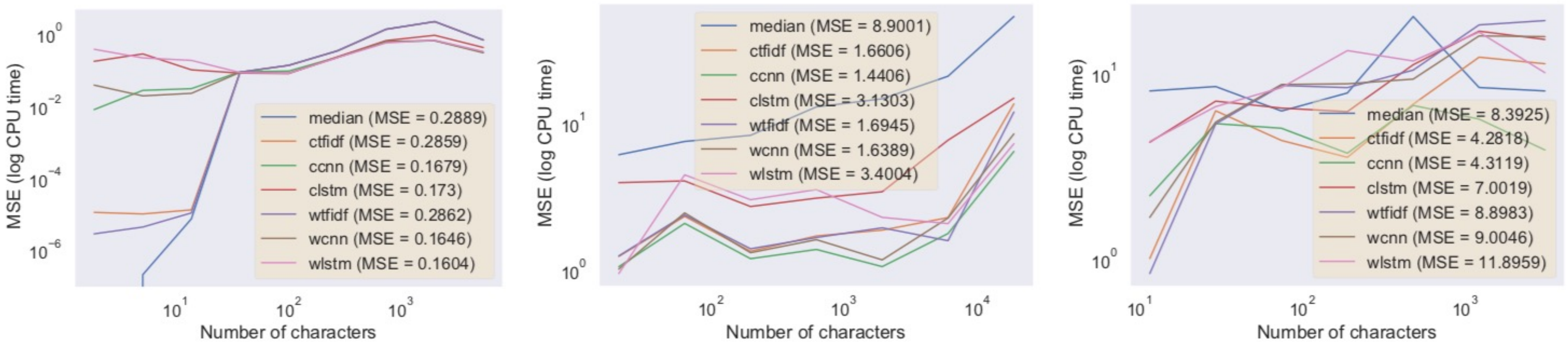


Results: CPU Time Prediction

- Goal: Predict logarithm of CPU time
- Measure mean squared error (MSE)
- Lowest MSE obtained by character-level models
    - ccnn is a shallow character-level cnn
- MSE of these models increases as statement complexity (Number of Characters) increases



Results: CPU Time Prediction in Different Settings



(a) Homogeneous Instance (b) Homogeneous Schema (c) Heterogeneous Schema

- From left to right, the range of MSE values increases as the problem setting complexity increases
- In each figure, the MSE of models increases as statement complexity increases
- Character-level models obtain lowest MSE and test loss value

Results: Answer Size Prediction

- Goal: predict answer size
- Report qerrors in different percentiles of the test data
- qerror: shows the factor by which a prediction differs from its true value

Answer size prediction qerror in **SDSS**

Model	50%	75%	80%	85%	90%	95%
median	1	36	50	144	1885	50000
ctfidf	1.13	4.86	10	25	88	727
ccnn	1.36	2.60	3.75	**6.79**	**18**	174
clstm	1.07	**2.38**	**3.50**	**6.79**	19	**172**
wtfidf	1.00	5.37	11.04	31.98	100	879
wcnn	1.33	3.42	5.14	10.93	36	295
wlstm	1.12	2.62	4.27	10.43	30	292

Highlights:

- For 50% of queries, it is easy to predict and for top 10% prediction is very difficult
- NN models outperform traditional models which have fixed features
- Character-levels obtain the lowest qerror

Selected Related Work

- Query Performance Prediction
  - [LGMB15] Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., & Neumann, T. (2015). How good are query optimizers, really?. Proceedings of the VLDB Endowment, 9(3), 204-215.
  - [LKNC12Li] Jiexing, Arnd Christian König, Vivek Narasayya, and Surajit Chaudhuri. "Robust estimation of resource consumption for sql queries using statistical techniques." Proceedings of the VLDB Endowment 5, no. 11 (2012):
  - [BDM19] Bailu Ding, Sudipto Das, Ryan Marcus,Wentao Wu, Surajit Chaudhuri, and Vivek Narasayya. 2019. AI Meets AI: Leveraging Query Executions to Improve Index Recommendations. In Proceedings of the 2019 ACM SIGMOD International Conference on Management of data. SIGMOD'19.
- SQL Query Workloads
  - [RTS14] M Jordan Raddick, Ani R Thakar, Alexander S Szalay, and Rafael DC Santos. 2014. Ten Years of SkyServer I: Tracking Web and SQL e- Science Usage. Computing in Science & Engineering 16, 4 (2014), 22–31.
  - [JMH16] Shrainik Jain, Dominik Moritz, Daniel Halperin, Bill Howe, and Ed Lazowska. 2016. Sqlshare: Results from a multi-year sql-as-a-service experiment. In Proceedings of the 2016 International Conference on Management of Data. ACM, 281–293.

Contributions: Facilitating SQL Query Composition and Analysis

- Introduce and address 4 problems for predicting query performance properties - prior to execution
- Approach is based on using large-scale real-world query workloads
- Conduct extensive workload analysis
- Adapt data-driven machine learning models
  - Establish baselines and assess feasibility
- Results show character level models (e.g., ccnn) generalize better under different problem settings