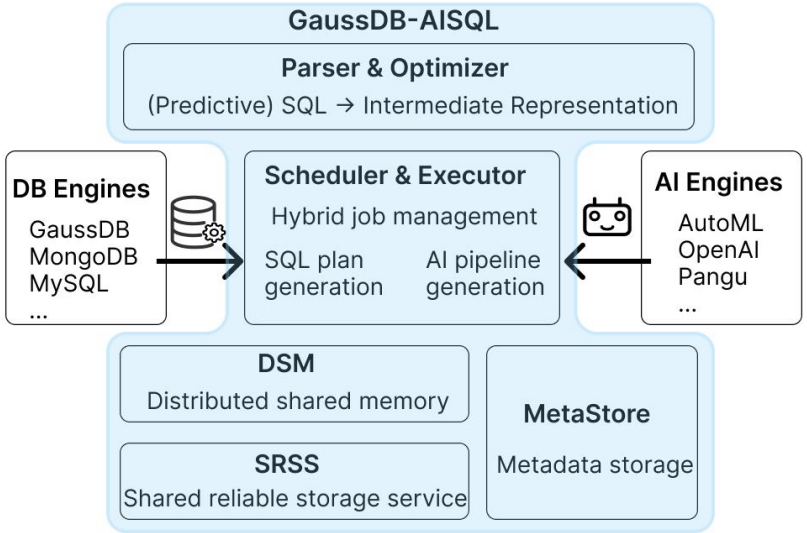
GaussDB-AISQL: A Composable Cloud-Native SQL System with AI Capabilities

Cheng CHEN, Wenlong MA, Congli GAO, Wenliang ZHANG, Kai ZENG, Tao YE, Yueguo CHEN, Xiaoyong DU

Frontiers of Computer Science, DOI: 10.1007/s11704-024-40624-2

Problems & Ideas

- Problems of supporting AI queries in SQL:
  - Bridging the gap between AI and SQL achieves predictive analytics within data warehouse.
  - Existing approaches are often limited by a complex syntax or slow data transportation across engines.
- Ideas: In this paper, we present GaussDB-AISQL, a composable SQL system with AI capabilities.

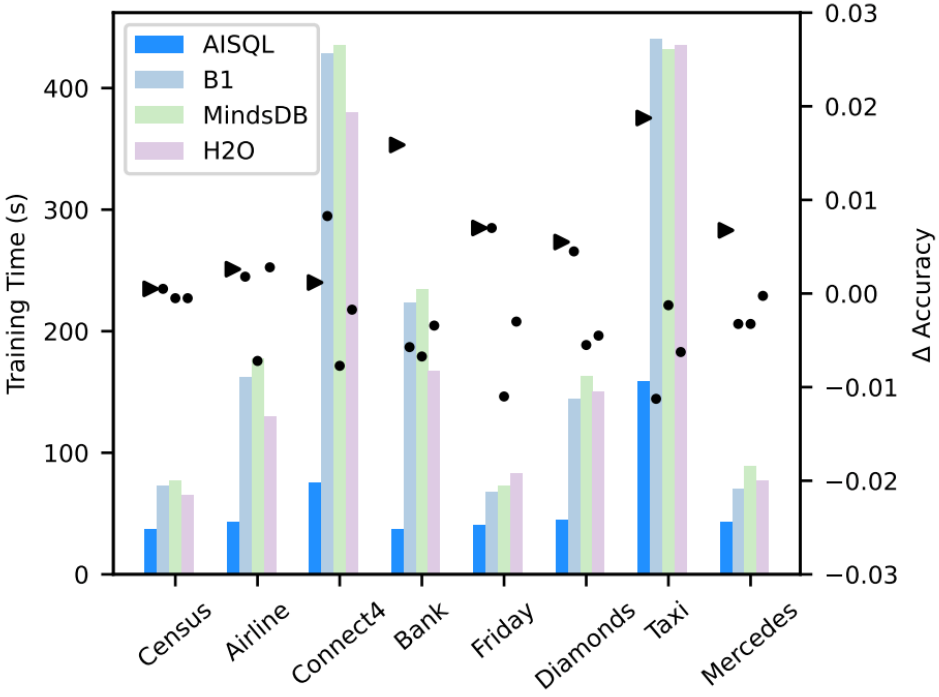


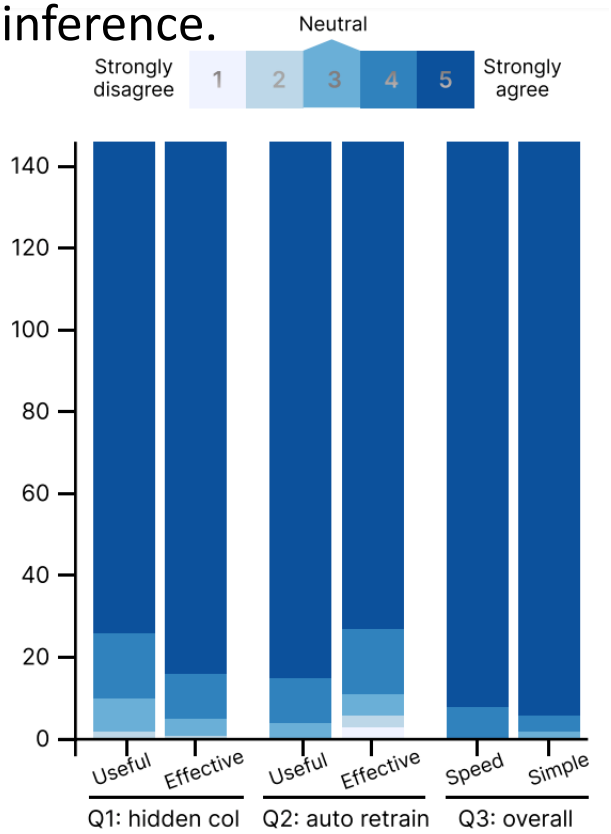

```
-- Transparent model training
CREATE TABLE iris (
  sepal_length FLOAT, sepal_width FLOAT,
  petal_length FLOAT, petal_width FLOAT,
  species char(20) PREDICT KEY);

-- Transparent model inference
-- Predicted column naming: <col_name>_p
SELECT species_p FROM iris;

-- Using the predicted column
SELECT sepal_length, species_p FROM iris WHERE
sepal_length > 5 AND species_p = 'setosa';
```

Left: GaussDB-AISQL extends the parser of the DB engine, and supports a wide range of AI engines. It connects all the sub-systems in a composable way. Right: We support model-transparent predictive analytics. Users don't need to manually create the model so it's extremely easy-to-use. To the best of our knowledge, AISQL is the first system to provide such kind of transparent SQL interface for running AI in SQL.

Main Contributions

- Contributions:
  - A composable system design and job-level resource management.
  - A complete set of model management functionalities with extremely simple-to-use triggers.
  - Efficient, data-centric model training and inference.





Left: The training time (s) and accuracy results for ML tasks. AISQL outperforms all the other baseline system by a large margin, w.r.t. training time and accuracy. Right: The results of the qualitative evaluation. Users think our novel syntax is useful and effective, and enjoys the overall speed and simplicity of AISQL.