ReSequel: Robust LLM-assisted Query Rewriting and Optimization using Templatization and Sampling

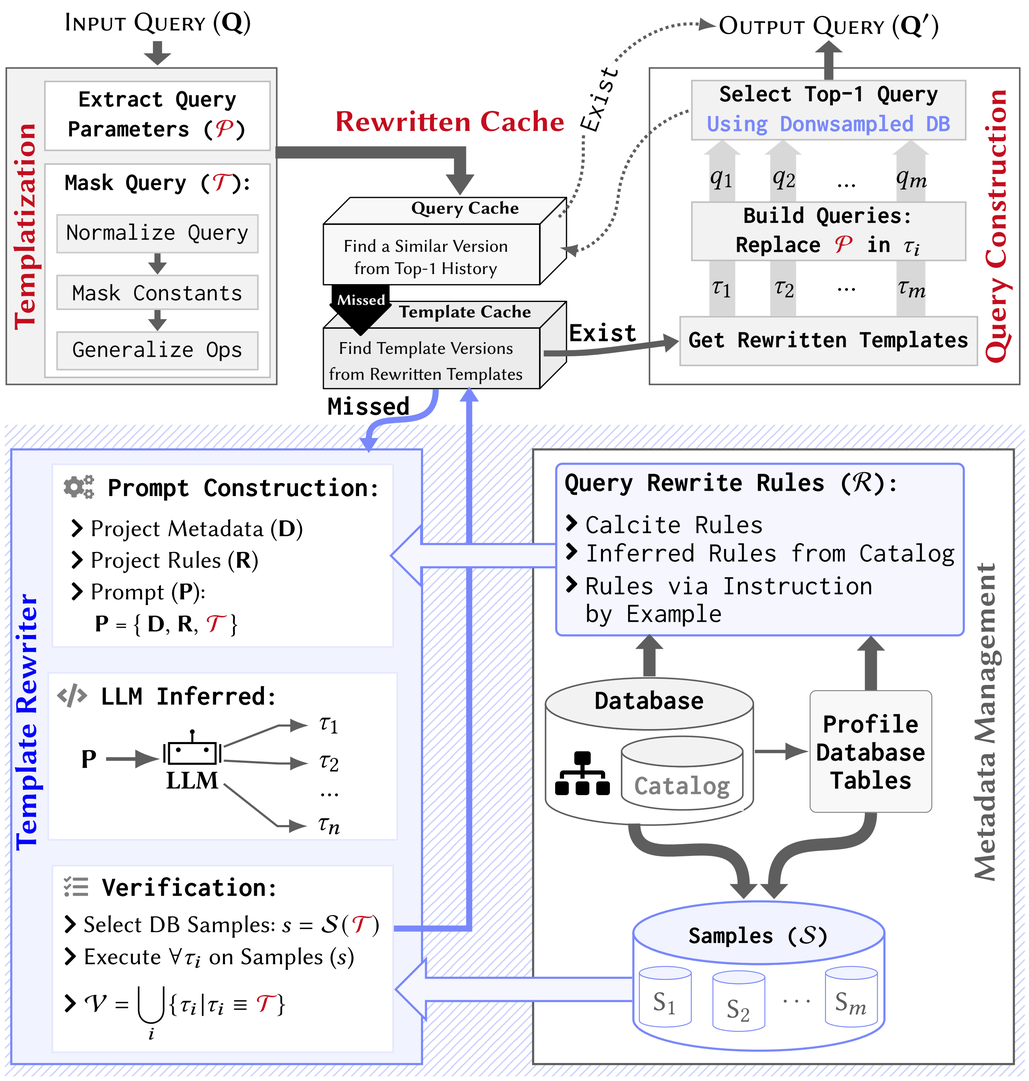
Saeed Fathollahzadeh [1] Essam Mansour [1] Matthias Boehm [2,3]

1 Concordia University
2 Technische Universität Berlin
3 BIFOLD

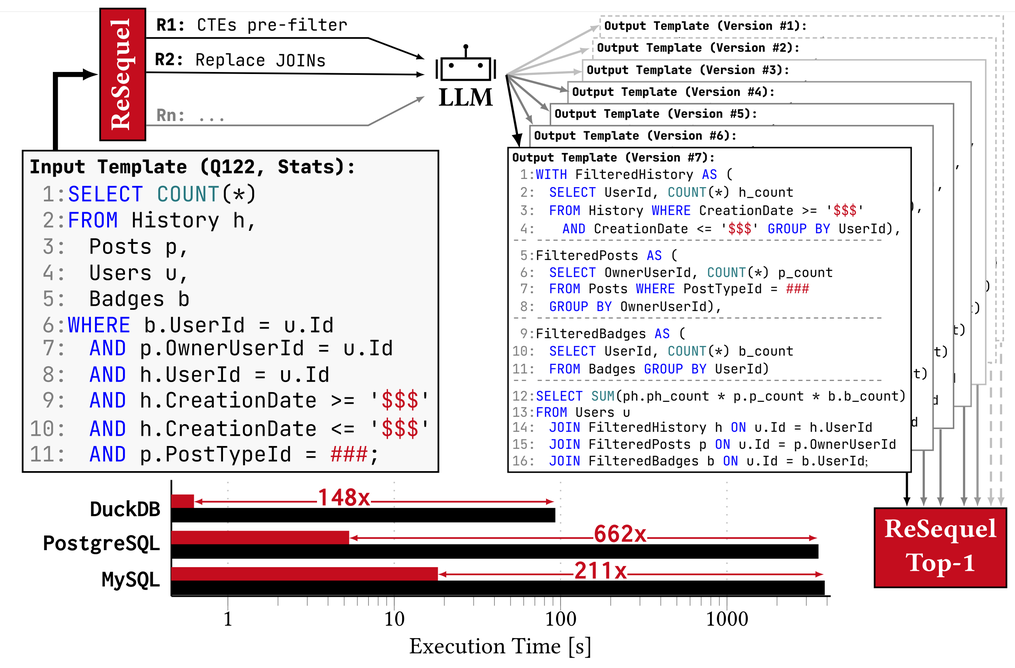
Paper Code

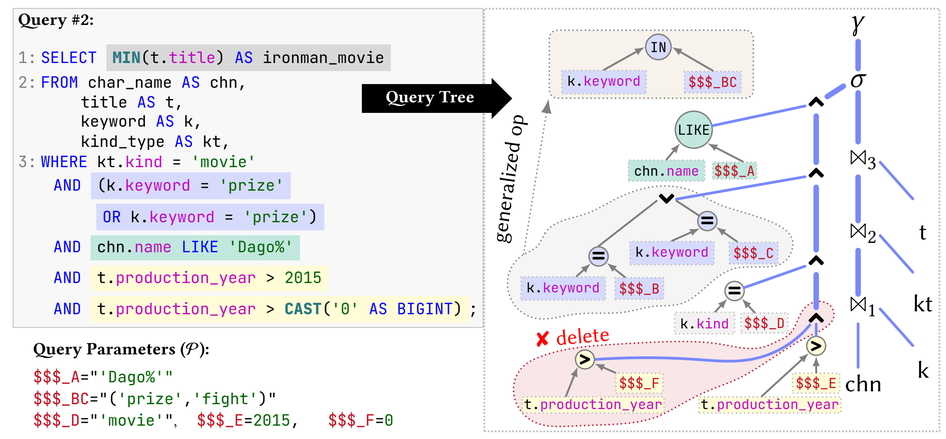
1. ReSequel: An Outer-Layer Optimizer

1. **Templatization** ➜ rewrite the query **shape** — paid **once per template**.
2. **Metadata-guided** ➜ catalog, statistics, and Calcite rules steer the LLM.
3. **Verified, not trusted** ➜ variants are checked on a **downsampled** DB; **Top-1** is cached.
4. **DBMS-agnostic** ➜ an **outer layer** — no engine changes on PostgreSQL, MySQL, DuckDB.



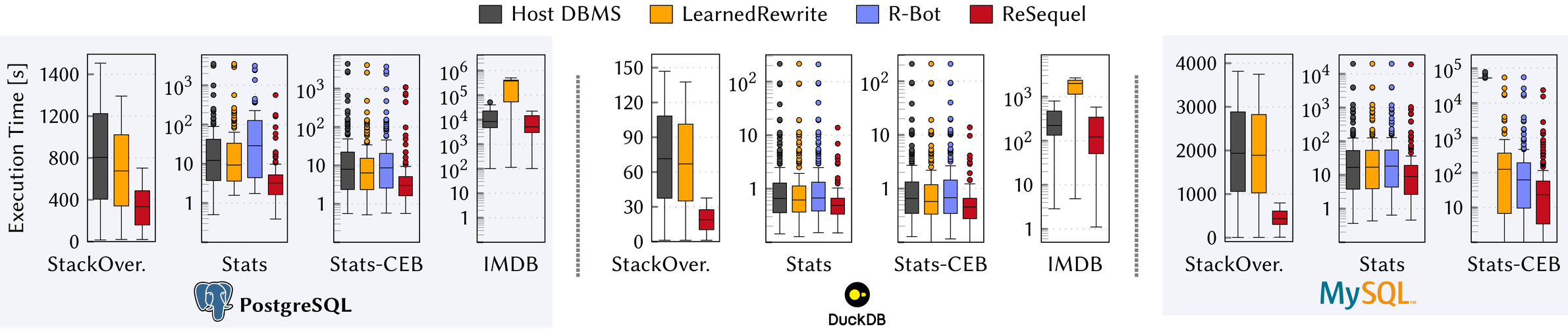
2. Rewriting Workflow Examples



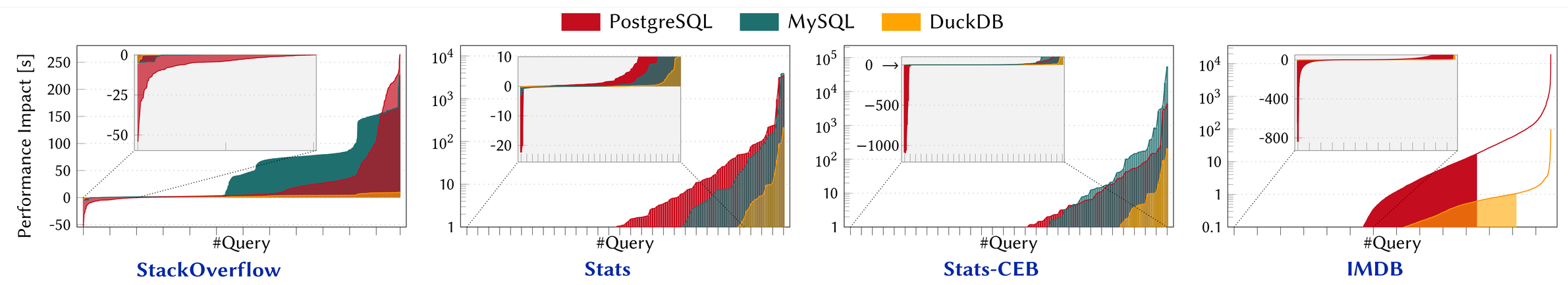


Q2: a query with a generalized operator and a redundant filter

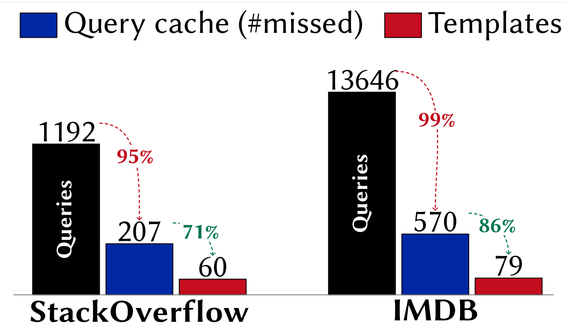
3. End-to-End Performance (LLM = Gemini-2.5-pro, 10% Downsampling)



4. Individual Query Performance Impact



5. Caching: One LLM Call per Template



- **Query cache** ➜ a similar query reuses its cached $Top\text{-}1$ ➜ **no verification**.
- **Template cache** ➜ a masked query matching a rewritten template reuses its versions ➜ **no LLM call**.
- **Miss path** ➜ only a template miss calls the LLM; verified once, then cached.

6. Conclusions

- **Outer-layer workflow** ➜ catalog metadata, DBMS statistics, and LLM reasoning over **unmodified** engines.
- **Template generalization** ➜ masked predicates collapse similar queries into one reusable template.
- **Sample-based verification** ➜ equivalence checked on a downsampled DB; $Top\text{-}1$ ranked by measured runtime.
- **Rewriting and caching** ➜ structured prompts, query reconstruction, and template/query caches.
- **Messy-query coverage** ➜ ReSequel addresses **88%** of the messy rewrite objectives, against **24%** for DBMS optimizers and **38%** for rewrite tools.
- **Low cost, low LLM dependency** ➜ one call **per template**, not per query; a cache hit needs no LLM, no sample runs, and no ranking.
- **Flexible** ➜ unseen queries matching a template are rewritten for free, and masked parameters keep templates valid as the database is updated.
- **Workload speedup** ➜ up to **16×** over native DBMSs and **22×** over baselines; single queries beyond **600×**.

saeed.fathollahzadeh@concordia.ca