Adversarial Benchmark Generation

Jeffrey Tao, Yimeng Zheng, Natalie Maus, Haydn Jones, Jacob Gardner, Ryan Marcus | DB@Penn

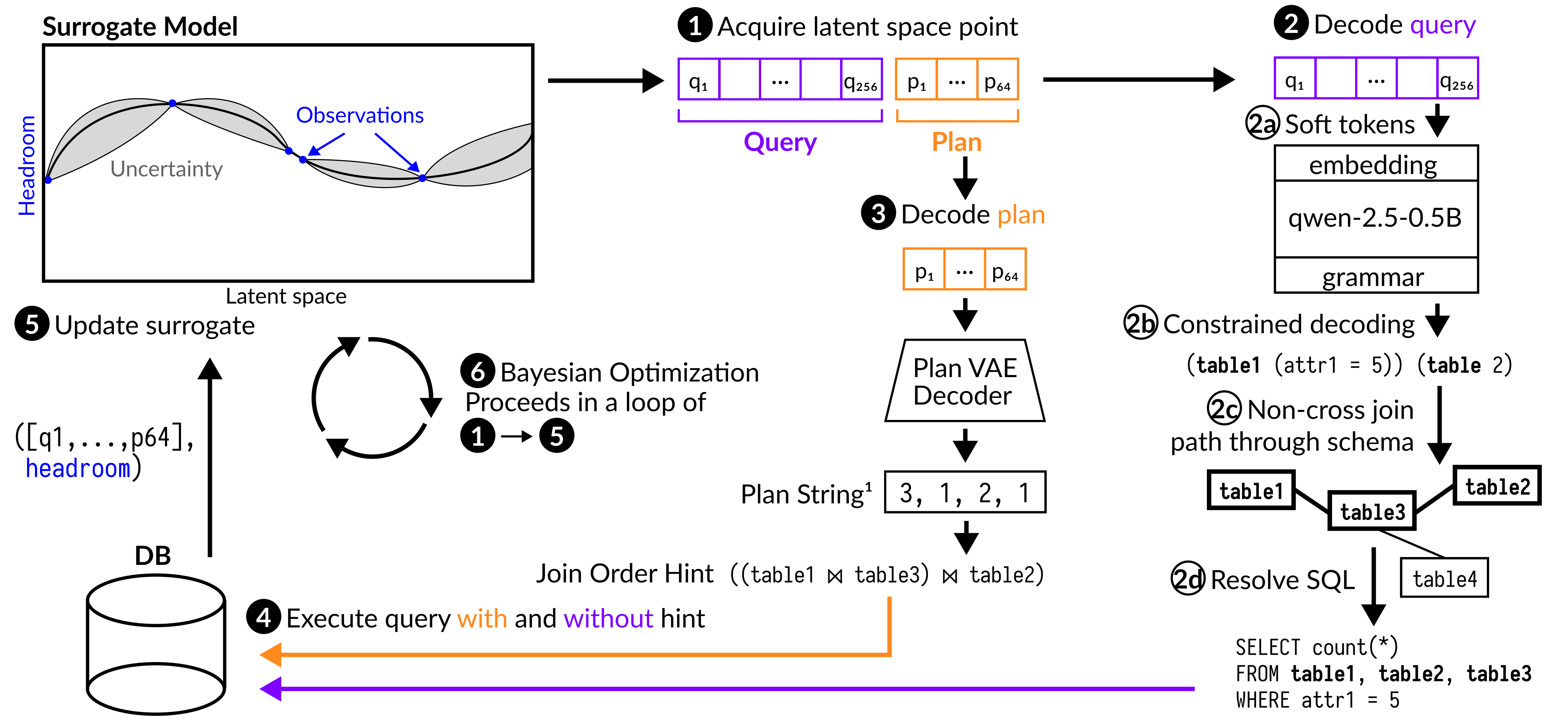

Figure 1. Our system generates a benchmark by searching the joint space of queries and plans using Bayesian Optimization.

Motivation

Benchmarks help us build high-performance systems. Recent SQL database benchmarks have focused on *realism*. But we may be over-indexing on optimizing what's already fast[2]!

We propose a direct method[3] for generating maximally *challenging* benchmarks:

1. Propose potentially difficult queries
2. Use offline optimization[1] to find faster plans
3. Maximize the DBMS's under-performance

We model this as a black-box optimization problem and leverage Bayesian optimization techniques. This allows us to directly find performance bugs within a given DBMS.

Results

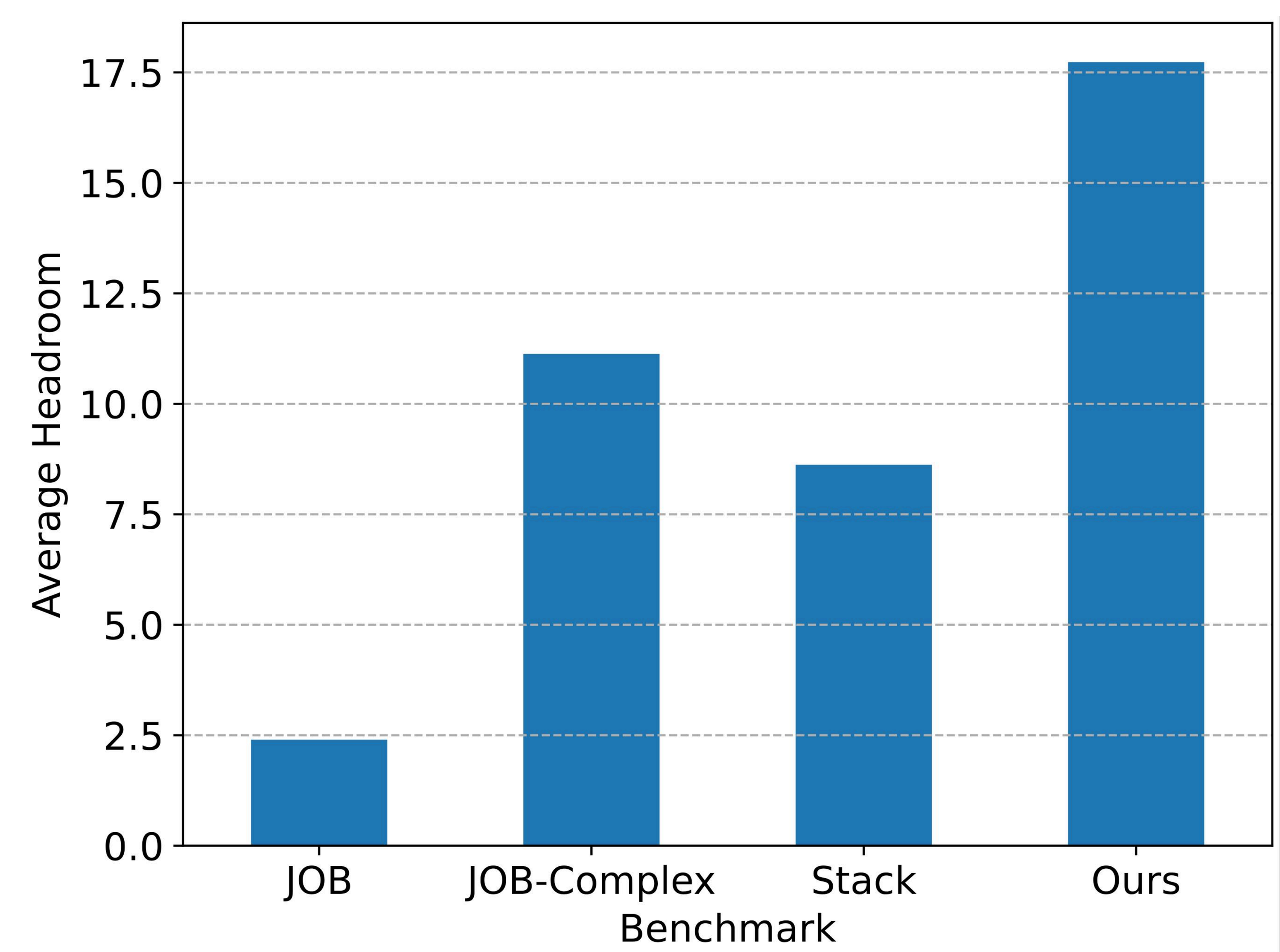

Figure 2. Our method produces more headroom (difference in plan latency) than prior techniques[4,5] because it directly optimizes for difference between the witness plan and the DBMS query optimizer's plan.

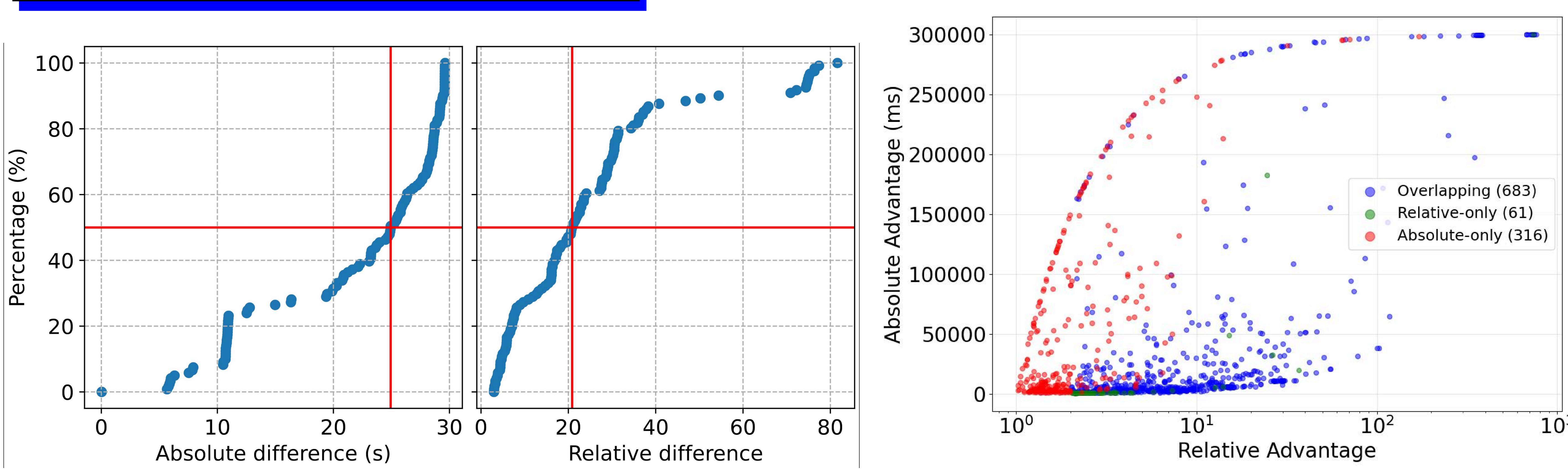

Figure 3. Left: We conduct optimization runs for absolute (DBMS - witness) and relative (DBMS / witness) difference, taking all queries with absolute difference > 1s. Right: Both optimization targets find many overlapping and some unique queries.

Future Work

- Generate benchmarks on other DBMSes to establish generality of our technique
- Compare performance bugs across systems
- Investigate why the DBMS's plan differs from the witness

[1] Tao et al., Learned Offline Query Planning via Bayesian Optimization, SIGMOD '25
[2] Marcus et al., Survivorship Bias in Industrial Database Workloads, CIDR '26
[3] Zeng et al., Adversarial Query Synthesis via Bayesian Optimization, ML for Systems@NeurIPS '25
[4] Wehrstein et al., JOB-Complex: A Challenging Benchmark for Traditional & Learned Query Optimization, AIDB '25
[5] Marcus et al., Bao: Making Learned Query Optimization Practical, SIGMOD '21

speculative.tech/nedb2026