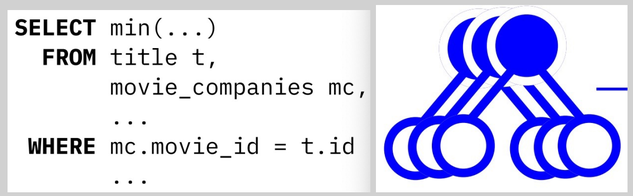
Adversarial Query Synthesis via Bayesian Optimization

Yimeng Zeng, Jeffrey Tao, Haydn T. Jones, Natalie Maus, Osbert Bastani, Jacob R. Gardner, Ryan Marcus
University of Pennsylvania

Adversarial Query Synthesis

As an optimization problem:

$\mathcal{A}$: {  ... }

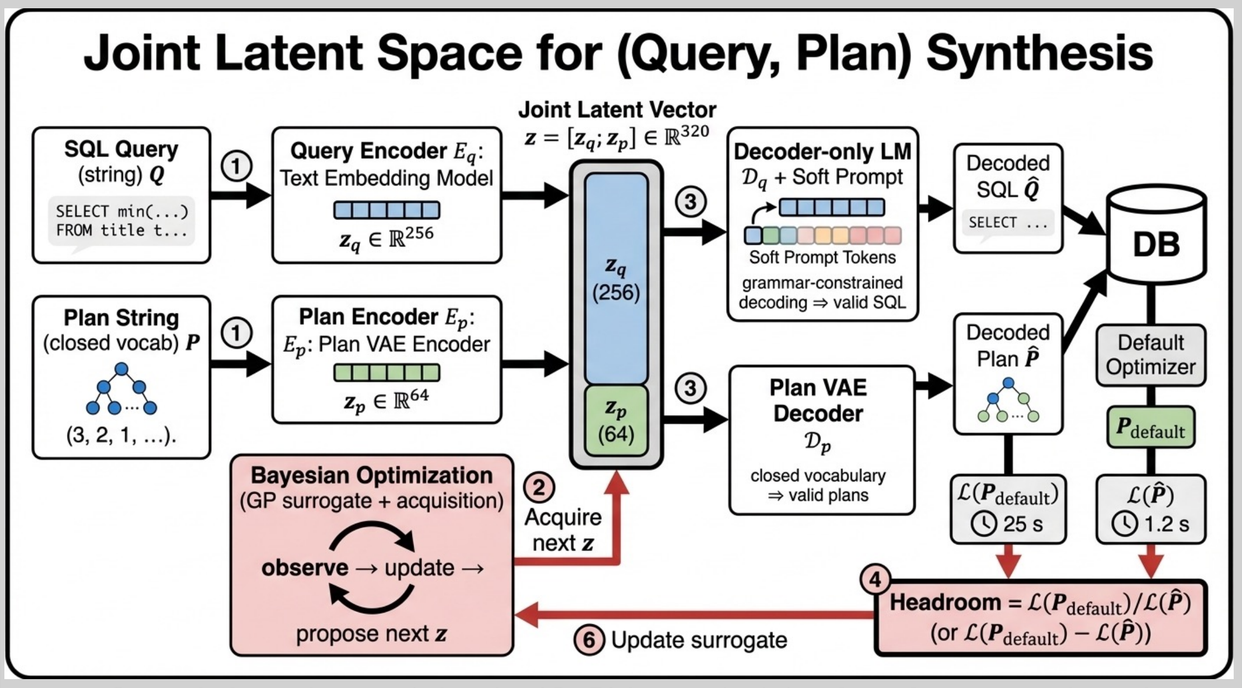
$$\max_{(Q,P)\in\mathcal{A}} \text{Headroom}(Q,P)$$

Maximize optimization headroom

$$\text{Relative}(Q,P) = \frac{L(P_{\text{default}})}{L(P)}$$
$$\text{Absolute}(Q,P) = L(P_{\text{default}}) - L(P)$$

Challenge: Optimizing over the space of all SQL queries and plans is hard

Joint Latent Space BO

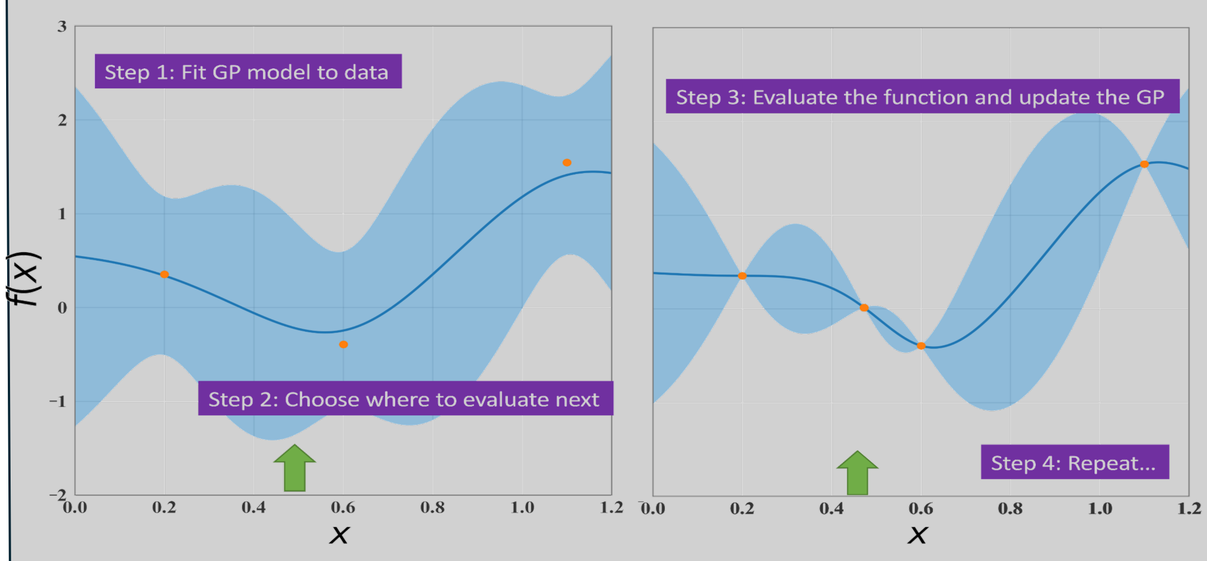


- Search for *adversarial pairs* (Q, P) by optimizing **headroom** against the DB's default plan
- Embed both query + plan into a **joint continuous latent** $z = [z_q; z_p]$ and run **Bayesian optimization** in z

Output is a benchmark entry: **decoded SQL** $\hat{Q}$ **+ witness plan** $\hat{P}$

Bayesian Optimization (BO)

BO: A model-based global optimization technique for black-box optimization.

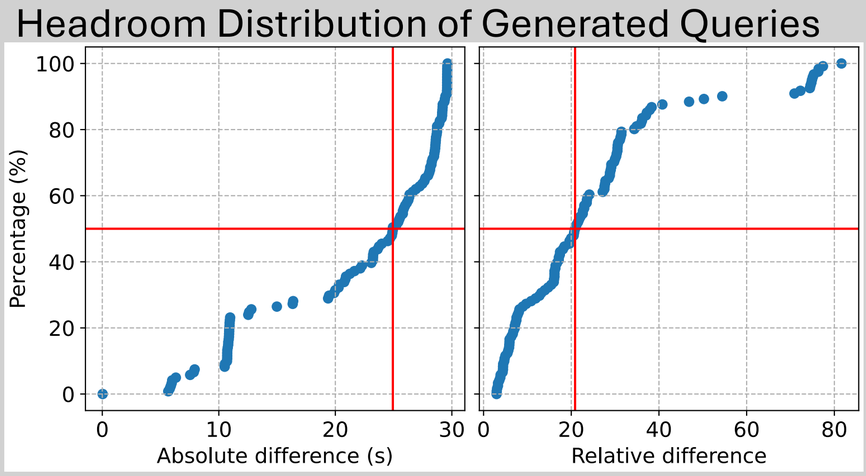


The literature supports: constraints, multi objective, diverse solutions, and more!

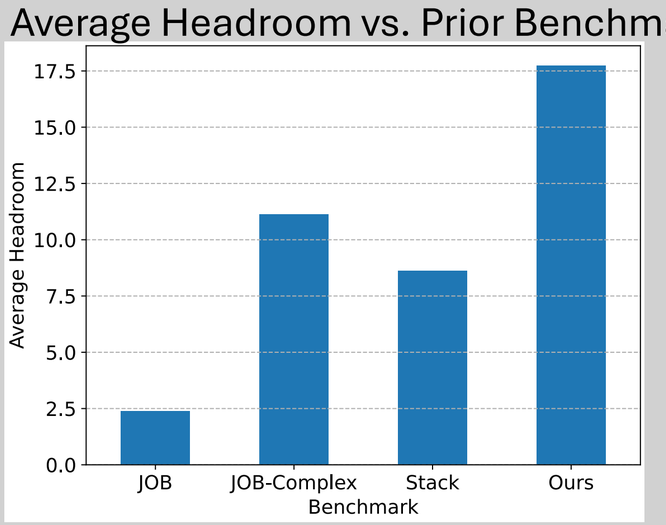
Historically: not very useful for high dimensional optimization, discrete/structured optimization, ...

Results

Headroom Distribution of Generated Queries

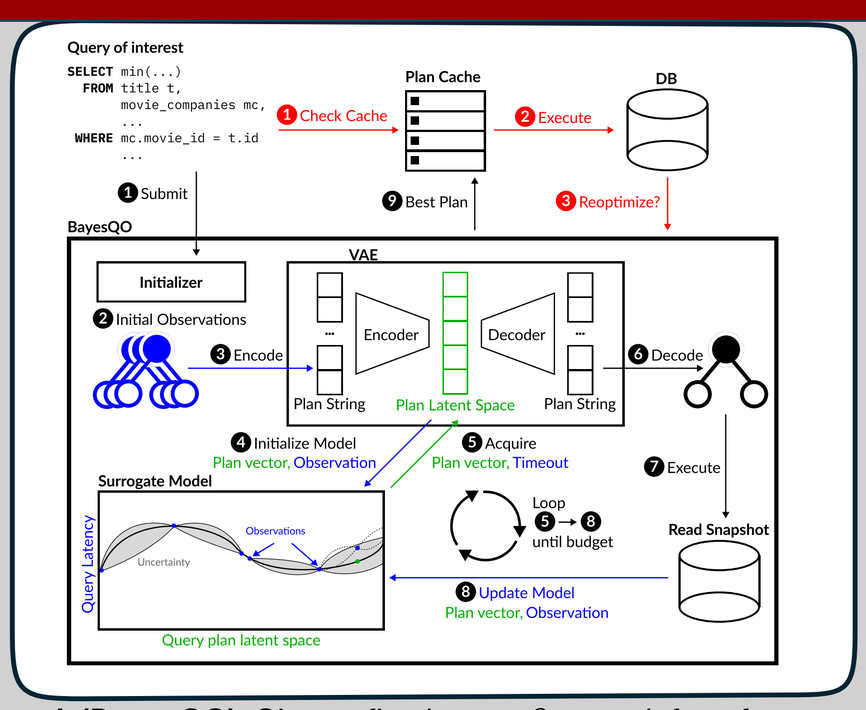


Average Headroom vs. Prior Benchmarks



- **Median** headroom is **~25s** absolute and **~20×** relative (red cross / lines).
- Tail queries are very hard: up to **~30s** absolute gap and **~80×** relative speedup.
- our synthesized queries provide **more optimization "room to improve"**
- Comparison uses reported values for JOB / JOB-Complex / Stack; "Ours" is computed w.r.t. our witness plans.

Query Optimization with BO



Prior work (BayesQO): Given a fixed query Q, search for a faster execution plan P under a limited execution budget.
Key idea: represent discrete plans in a continuous latent space (VAE) and run Bayesian optimization (GP + acquisition) to propose plans, execute, and update.

Difference: BayesQO optimizes plans for one query; we *synthesize (query, plan) pairs* to maximize headroom.

References

For Bayesian optimization:

- David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, Matthias Poloczek. Scalable Global Optimization via Local Bayesian Optimization (Neurips 2019).

- Natalie Maus, Haydn T. Jones, Juston Moore, Matthew J. Kusner, John Bradshaw, Jacob R. Gardner. Local Latent Space Bayesian Optimization over Structured Inputs. (NeurIPS 2022).

For query optimization:

- Jeffrey Tao, Natalie Maus, Haydn Jones, Yimeng Zeng, Jacob R. Gardner, and Ryan Marcus. Learned Offline Query Planning via Bayesian Optimization. (SIGMOD 2025).

- Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. How Good Are Query Optimizers, Really? **(VLDB 2015)**

For structured SQL generation:

- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. XGrammar: Flexible and Efficient Structured Generation Engine for Large Language Models. (Arxiv 2025).
- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. (EMNLP 2021).