arXiv:2603.01570v1 [cs.DB] 2 Mar 2026

通过贝叶斯优化进行对抗性查询合成

Jeffrey Tao* **Yimeng Zeng*** **Haydn T. Jones Natalie Maus**
Osbert Bastani Jacob R. Gardner Ryan Marcus宾夕法尼亚大学

摘要

基准工作负载对于数据库管理研究领域至关重要，尤其是随着越来越多的机器学习组件被集成到数据库系统中。本文提出一种贝叶斯优化技术，可自动搜索难度较高的基准查询，大幅减少通常所需的人工投入。在初步实验中，我们证明，与现有基准相比，该方法生成的查询可将优化空间提升至两倍以上。

1 引言

数据库研究人员致力于构建能够高效处理 SQL 查询的系统。为了评估相关工作，数据库领域依赖基准测试来近似模拟具有挑战性的查询。传统上，TPC 系列等合成基准（即从随机分布中抽取数据） [14, 2] 占据主导地位，但也有一些基准采用“真实世界”数据，例如基于互联网电影数据库（IMDb）的连接顺序基准（JOB） [8]。近期，运用机器学习技术构建数据库系统的研究（例如， [5]）进一步凸显了“真实世界”基准的重要性，因为合成数据分布通常过于简单（例如，均匀键列），或完全无法学习（例如，噪声） [10]。举例来说，AWS Redshift 最近的一份报告指出，与其观察到的工作负载相比，传统基准相对“容易” [23]。

这一差距引发了对更大、更具挑战性的基准测试的争相追逐（或许也可以说是一场军备竞赛），应对方案包括：将合成工作负载与真实追踪数据“匹配” [7], 手动基于公开数据构建新的查询集 [11], 使用大语言模型合成查询 [19],或向现有数据集添加更多查询 [25]。除 [19], 外，所有这些方法都需要投入大量人工。并且，除 [25], 外，所有这些方法都没有针对基准测试的难度进行优化（即相较于现有系统的潜在改进空间，我们称之为性能余量）。
在本研究中，我们提出了一个初步系统，可自动合成具有显著优化空间（即难度较高）的简单查询。我们的方法使用贝叶斯优化来发现查询与执行计划对，使发现的计划性能显著优于基线系统。除了完全自动化之外，这种方法还有三个主要优势：首先，通过直接针对查询的优化空间进行优化，我们表明可以找到优化空间超过现有基准测试两倍的查询。其次，除了查询之外，我们还搜索计划，从而生成性能显著优于基线系统的见证执行计划，这可能有助于调试性能缺陷。第三，由于我们将搜索空间限制为简单查询（不含别名连接的合取查询），我们构建的基准查询可在作者目前所知的所有关系数据库系统上运行。

*共同贡献。联系邮箱：jefftao@seas.upenn.edu

NeurIPS 2025 机器学习系统研讨会。

总体而言，我们将基准测试构建问题表述为一个优化问题，并采用结合了 SQL 查询和计划潜在空间表示的贝叶斯优化。为处理 SQL 查询结构化、离散的特性，我们使用复合变分自编码器，将查询和计划编码到共享的连续潜在空间中，从而实现高效搜索。查询通过大型文本嵌入模型进行嵌入，并由经过微调的紧凑型大语言模型使用语法约束解码进行解码 [3], ，以确保语法有效性；而计划则采用封闭词汇表表示，以保证格式良好。这种组合使系统能够高效地提出并评估候选对，迭代优化模型，从而发现具有挑战性且优化空间大的查询。

2 方法

我们要解决的核心问题是自动发现对抗对（Q, P），其中 Q 是 SQL 查询，P 是 Q 的执行计划。所谓对抗对，是指计划 P 的延迟（记为$L(P)$）显著低于数据库默认查询优化器生成的计划的延迟，即 $L(P_{\text{default}})$。我们将此问题建模为黑箱优化问题，目标是最大化**相对加速比** $L(P_{\text{default}})/L(P)$ 或**绝对加速比** $L(P_{\text{default}}) - L(P)$。

黑箱优化与贝叶斯优化。在黑箱优化中，我们的目标是在候选空间 $\mathbf{x}^* =$ 上优化一个预言机目标函数 $f(\mathbf{x})$，即 $\text{argmax}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$。此类问题的例子包括药物发现中的分子活性最大化 [22, 12], ，以及 DNA 序列或蛋白质的结合亲和力 [1, 4]。通常，我们假设 $f(x)$ 的评估成本高昂，甚至完全未知。

贝叶斯优化是一种样本高效框架，用于解决这类昂贵的基于模型的优化问题 [15, 13, 20]。在 BO 的第 t 次迭代中，我们可以获得观测值 $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^t$，其中 y_i 表示输入 x_i的目标值。通常，会使用高斯过程 [18]作为代理模型，利用这些输入和值来近似目标函数。该代理模型通过采用采集函数来辅助优化，并由采集函数有策略地提出下一个待评估候选项。通过真实预言机查询这些候选项后，代理模型会使用新的观测值进行更新。该过程逐步构建更全面的数据集并不断优化代理模型，从而提高未来迭代中所提出样本的质量。

针对 SQL 查询的贝叶斯优化。贝叶斯优化是一种样本高效技术，用于优化昂贵的黑盒函数。然而，标准 BO 在连续向量空间上运行，而我们的搜索空间由离散的结构化对象（SQL 查询和查询计划）组成。我们通过潜在空间贝叶斯优化（LS-BO）来弥合这一差距；事实证明，该技术在分子设计 [12] 和离线查询规划 [21]等领域中十分有效。

LS-BO 框架需要两个关键组件：(1) 一个编码器 $\mathcal{E}$，将结构化输入映射到连续潜在空间 $\mathcal{Z}$；以及 (2) 一个解码器 $\mathcal{D}$，将 $\mathcal{Z}$ 中的点映射回结构化输入空间。本文提出了一种新颖的复合变分自编码器（VAE）架构，为查询和计划共同创建一个联合潜在空间。

我们构建了一个连续潜在空间 $\mathcal{Z} = [z_q;\ z_p] \in \mathbb{R}^{320}$，将查询VAE潜变量（$z_q \in \mathbb{R}^{256}$）与计划VAE潜变量（$z_p \in \mathbb{R}^{64}$）拼接起来：

$$\mathcal{E}_q : \text{SQL string} \to z_q, \qquad \mathcal{D}_q : z_q \to \text{SQL string},$$
$$\mathcal{E}_p : \text{plan string} \to z_p, \qquad \mathcal{D}_p : z_p \to \text{plan string}.$$

在推理时，解码 z 会得到一个可执行对（Q, P）；计划字符串会转换为数据库专用指令（例如优化器提示字符串 [16]），并与 Q 结合后执行。

计划表示与变分自编码器。我们的计划表示与先前使用贝叶斯优化进行查询规划的工作完全相同 [21]。总体而言，我们将计划编码为整数字符串，再将其解释为算子选择和连接顺序。与 SELFIES [6], 类似，每个字符串都表示一个有效计划，且每个计划至少由一个字符串表示；不过，与 SELFIES 不同的是，有些计划可以由多个字符串表示。这意味着，从 $\mathcal{E}_p$ 解码出的任何序列都可以解释为一个有效计划。

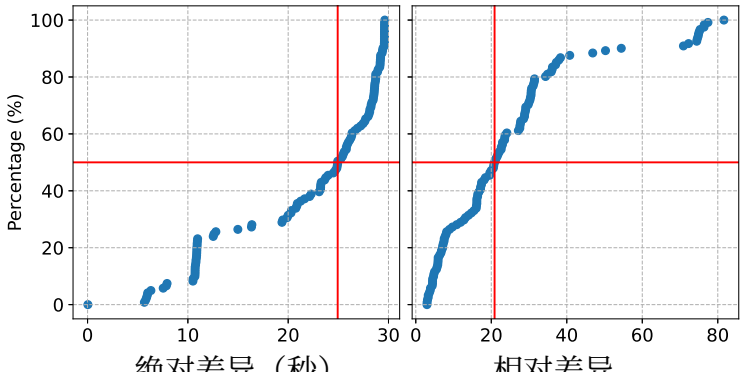


图1：我们生成的基准测试中，基线系统（DuckDB）与见证计划之间绝对差异和相对差异的累积分布函数。红色十字标示中位数。

查询VAE。 与查询计划不同，SQL 缺少类似 SELFIES 的 [6] 封闭词汇表，无法保证任意词元序列都有效。为避开数据稀缺问题（~20k 个查询）和语法复杂性，我们不训练完整的查询编码器—解码器VAE。相反，我们（i）使用文本嵌入模型作为查询编码器 $\mathcal{E}_q$ （OpenAI text-embedding-3-large），然后（ii）微调一个小型大语言模型（Qwen-2.5-0.5B），作为紧凑的查询解码器 $\mathcal{D}_q$，将前 256 个嵌入维度映射回 SQL 查询。这与提示调优 [9]，类似：将嵌入作为软提示传递给大语言模型，并训练大语言模型将嵌入解码回原始查询字符串。推理时，我们使用语法约束解码，以确保重建后的查询在语法上正确。由此，查询的字符串重建准确率达到 ~67%，计划则达到 ~99%。

简单 SQL 子集 为了生成适用于各种数据库的简单查询，我们聚焦于合取查询 [24]。具体而言，查询仅限于：（1）通过等值连接（例如，外键与主键之间的连接）将关系连接起来；（2）对表列使用合取谓词。每个查询都执行不分组的 count(*)聚合。因此，我们的表述不包含子查询、窗口函数、递归等 SQL 特性。尽管合取查询很简单，但现代查询优化器仍难以处理它们 [8]。

SQL 的语法约束解码。 为确保查询解码器 $\mathcal{D}_q$ 始终解码出有效查询，我们为 SQL 子集编写了简洁的扩展巴科斯范式（通过枚举可连接表和可筛选列的幂集），并在 $\mathcal{D}_q$ 解码过程中使用 vLLM 进行语法约束生成来强制执行该范式。这解决了无约束解码带来的主要挑战：（i）语法无效性；（ii）语义上不可能的引用（例如，未定义别名）。语法约束仅适用于 $\mathcal{D}_q$，因为 $\mathcal{D}_p$ 采用封闭词汇表，其中任何解码序列始终有效。

3 初步结果

为检验我们的方法是否可行，我们针对 IMDb数据集生成了一个工作负载（选择该数据集是为了便于与使用同一数据集的先前研究进行比较）。为了与原始连接顺序基准保持一致 [8],，我们生成了 122 个查询计划。这些计划是通过一次为期一周的贝叶斯优化运行，从相对性能余量和绝对性能余量方面选出最佳候选项后收集的。查询在 DuckDB [17] 上执行，所用节点配备 AMD Ryzen 5 3600 处理器和 64GB 内存。整个数据集均已缓存至内存中。
我们在图1中绘制了绝对性能余量和相对性能余量的累积分布函数。在我们生成的工作负载中，中位数查询的绝对性能余量为25秒，这意味着见证计划比DuckDB选择的默认计划快25秒。就相对性能余量而言，我们生成的工作负载中，中位数查询的见证计划比DuckDB选择的默认计划快20倍。发现的最大差异在绝对时间上接近30秒，在相对比例上则接近80倍。我们工作负载中难度最低的查询，绝对性能提升不到1秒，相对性能提升也只有1.5倍（这可能是噪声所致）。我们希望

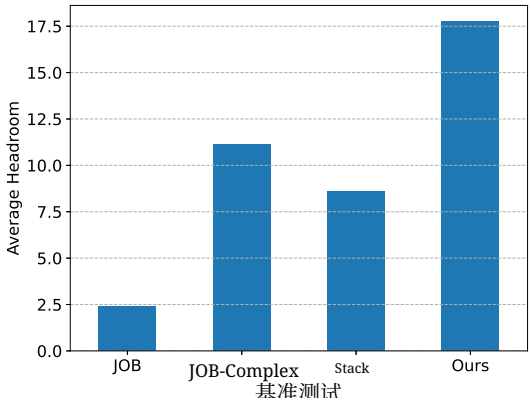


图2：四个基准测试的几何平均相对优化空间对比。JOB 和 JOB-Complex 的数值来自 [25]。Stack 的数值根据 [11]近似得出。我们的方法的数值是相对于见证计划计算的。

未来更长时间的贝叶斯优化运行能继续使这两种分布向右移动；生成的查询中最弱查询相对“容易”，可能是因为搜索时间不足。

为了将我们生成的基准测试与先前工作进行比较，图2展示了不同基准测试的平均（几何）相对性能余量。由于计算基准测试的性能余量代价高昂（即，每个查询都必须执行数百万个查询计划），我们依赖先前工作报告的结果，因此只比较平均相对性能余量（因为绝对性能余量的报告并不一致）。

4 结论与未来工作

在这篇扩展摘要中，我们提出了一种全自动方法，通过将基准测试创建视为优化问题，并在涵盖 SQL 查询与执行计划的联合潜在空间中使用贝叶斯优化，来构建具有挑战性的数据库基准测试。我们结合使用变分自编码器解码器，为查询和计划生成联合嵌入；同时，语法约束解码确保生成的 SQL 在一个简单且易于移植的子集中（包含等值连接和谓词的合取查询）具备语法有效性。这些选择共同使系统能够高效地找到具有显著优化空间的对抗性（查询、计划）对。

在 IMDb 上使用 DuckDB 进行的初步实验显示出令人鼓舞的迹象：经过为期一周的贝叶斯优化运行，我们选出了 122 个候选项；由此形成的工作负载中，默认计划与发现的见证计划之间的绝对性能余量中位数为 25 秒，相对性能余量中位数为 20×（最大差距分别接近 30 秒和 80×）。这些结果表明，我们的自动化流程能够找到比常用基准测试中的查询难得多的查询。

未来的工作中，若能通过更大规模的扩展运行，收集更多难解的（查询，计划）对，我们便可以微调模式条件化大语言模型，使其直接批量生成这些配对，从而在大多数情况下跳过贝叶斯优化。模型将从（SQL、计划、性能余量）三元组中学习，并使用语法约束解码来确保有效性，同时搭配快速验证器（解析 + 快速执行/成本检查）来筛选输出。通过简单的奖励模型或偏好更高性能余量的 DPO式目标函数，可以引导模型生成更多对抗性区域的样本，而无需显式搜索。实践中，我们可以批量生成候选项，通过验证器筛选出前 k 个；仅在置信度较低时才回退到优化，从而缩短实际运行时间，并通过定期开展主动学习会话来保持质量。

总的来说，我们的方法减少了人工工作量，通过性能余量直接针对“难度”，并生成具体的见证执行计划，从而有助于调试和系统改进，同时确保查询能够在各个系统中广泛执行。

致谢与资金披露

N. Maus 获得了美国国家科学基金会研究生研究奖学金的支持；J. R. Gardner 获得了美国国家科学基金会项目 IIS-2145644 和 DBI-2400135 的资助；

参考文献

[1] Luis A. Barrera、Anastasia Vedenko、Jesse V. Kurland、Julia M. Rogers、Stephen S. Gisselbrecht、Elizabeth J. Rossin、Jaie Woodard、Luca Mariani、Kian Hong Kock、Sachi Inukai、Trevor Siggers、Leila Shokri、Raluca Gordân、Nidhi Sahni、Chris Cotsapas、Tong Hao、Song Yi、Manolis Kellis、Mark J. Daly、Marc Vidal、David E. Hill 和 Martha L. Bulyk。《人类转录因子变异调查揭示了普遍存在的 DNA 结合变化》。*Science*，351(6280):1450–1454，2016。doi: 10.1126/science.aad2257。[2] Peter Boncz、Thomas Neumann 和 Orri Erling。《解析 TPC-H：从一项具有影响力的基准测试中发现的隐含信息与经验教训》。载于第 *5* 届 *TPC* 技术会议性能表征与基准测试精选修订论文集——第 *8391* 卷，TPC '14，第 61–76 页，柏林，海德堡，2014。施普林格出版社。ISBN 978-3-319-04935-9。doi: 10.1007/978-3-319-04936-6_5。URL https://doi.org/10.1007/978-3-319-04936-6_5。[3] Yixin Dong、Charlie F. Ruan、Yaxing Cai、Ruihang Lai、Ziyi Xu、Yilong Zhao 和 Tianqi Chen。《XGrammar：面向大语言模型的灵活高效结构化生成引擎》，2025 年 5 月。URL http://arxiv.org/abs/2411.15100。arXiv:2411.15100 [计算机科学]。[4] Nate Gruver、Samuel Stanton、Nathan C. Frey、Tim G. J. Rudner、Isidro Hotzel、Julien Lafrance- Vanasse、Arvind Rajpal、Kyunghyun Cho 和 Andrew Gordon Wilson。通过引导式离散扩散进行蛋白质设计。*arXiv* 预印本，2023。doi: 10.48550/arXiv.2305.20009。[5] Tim Kraska、Alex Beutel、Ed H. Chi、Jeffrey Dean 和 Neoklis Polyzotis。《学习型索引结构的价值》。载于*2018* 年国际数据管理会议论文集，SIGMOD '18，纽约，美国纽约州，2018。美国计算机协会。ISBN 978-1-4503-4703-7。doi: 10.1145/3183713.3196909。URL http://doi.acm.org/10.1145/3183713.3196909。[6] Mario Krenn、Florian Häse、AkshatKumar Nigam、Pascal Friederich 和 Alán Aspuru-Guzik。自引用嵌入字符串（SELFIES）：一种 100% 鲁棒的分子字符串表示，2020 年 3 月。URL http://arxiv.org/abs/1905.13741。arXiv:1905.13741。[7] Skander Krid、Mihail Stoian 和 Andreas Kipf。Redbench：反映真实工作负载的基准测试，2025 年 6 月。URL http://arxiv.org/abs/2506.12488。arXiv:2506.12488 [计算机科学]。[8] Viktor Leis、Andrey Gubichev、Atanas Mirchev、Peter Boncz、Alfons Kemper 和 Thomas Neumann。查询优化器究竟有多好？*PVLDB*，9(3):204–215，2015。ISSN 2150- 8097。doi: 10.14778/2850583.2850594。URL http://dx.doi.org/10.14778/2850583. 2850594。[9] Brian Lester、Rami Al-Rfou 和 Noah Constant。规模对参数高效提示调优的作用，2021。URL https://arxiv.org/abs/2104.08691。

[10] Ryan Marcus、Andreas Kipf、Alexander Van Renen、Mihail Stoian、Sanchit Misra、Alfons Kemper、Thomas Neumann 和 Tim Kraska。《学习索引基准测试》。*PVLDB*，14(1)：1–13，2020年9月。doi：10.14778/3421424.3421425。引文键：benchmark_lis。[11] Ryan Marcus、Parimarjan Negi、Hongzi Mao、Nesime Tatbul、Mohammad Alizadeh 和 Tim Kraska。《Bao：让学习型查询优化切实可行》。载于*2021*年国际数据管理会议论文集，SIGMOD '21，中国，2021年6月。ISBN 978-1-4503-8343-1。doi：10.1145/3448016.3452838。获奖情况：'最佳论文奖'。

[12] Natalie Maus、Haydn Jones、Juston Moore、Matt J Kusner、John Bradshaw 和 Jacob Gardner。针对结构化输入的局部潜在空间贝叶斯优化。载于神经信息处理系统进展，第35卷，第34505–34518页。Curran Associates, Inc.，2022年。doi：10.48550/arXiv.2201.11872。

[13] Jonas Mockus。《贝叶斯方法》用于全局优化。载于系统建模与优化，第 473–481 页。Springer，1982年。

[14] Raghunath Othayoth Nambiar 和 Meikel Poess。TPC-DS 的构建。载于*VLDB*，VLDB '06，第 1049–1058 页，韩国首尔，2006 年。VLDB 基金会。URL http://dl.acm.org/ citation.cfm?id=1182635.1164217.[15] M. A. Osborne、R. Garnett 和 S. J. Roberts。用于全局优化的高斯过程。载于第 *3* 届学习与智能优化国际会议（*LION3*），第 1–15 页，2009 年。[16] PostgreSQL 开发者。PostgreSQL 提示，https://www.postgresql.org/docs/current/runtime- config-query.html，2024 年。URL https://www.postgresql.org/docs/current/ runtime-config-query.html。tex.key = l.[17] Mark Raasveldt 和 Hannes Mühleisen。DuckDB：可嵌入式分析数据库。载于 *2019* 年国际数据管理会议论文集，SIGMOD '19，第 1981–1984 页，美国纽约州纽约市，2019 年 6 月。计算机协会。ISBN 978-1-4503-5643-5。doi：10.1145/3299869.3320212。URL https://dl.acm.org/ doi/10.1145/3299869.3320212.[18] Carl Edward Rasmussen。机器学习中的高斯过程。载于机器学习暑期学校，第 63–71 页。Springer，2003 年。[19] Tobias Schmidt、Viktor Leis、Peter Boncz 和 Thomas Neumann。SQLStorm：将数据库基准测试带入大语言模型时代。*PVLDB*，18(11):4144–4157，2025。doi：10.14778/3749646. 3749683。[20] Jasper Snoek、Hugo Larochelle 和 Ryan P. Adams。机器学习算法的实用贝叶斯优化。载于*NeurIPS* 论文集，第 25 卷，第 2951–9 页，2012 年。doi：10.48550/arXiv. 1206.2944。[21] Jeffrey Tao、Natalie Maus、Haydn Jones、Yimeng Zeng、Jacob R. Gardner 和 Ryan Marcus。通过贝叶斯优化进行学习型离线查询规划。载于*2025* 年国际数据管理会议论文集，SIGMOD '25，德国柏林，2025 年 6 月。美国计算机协会。[22] Brandon Trabucco、Xinyang Geng、Aviral Kumar 和 Sergey Levine。Design-bench：数据驱动的离线基于模型优化基准测试。载于第 *39* 届国际机器学习会议论文集，PMLR 第 162 卷，第 21658–21676 页。PMLR，2022 年 7 月 17–23 日。doi：10.48550/arXiv.2202.08450。[23] Alexander van Renen、Dominik Horn、Pascal Pfeil、Kapil Eknath Vaidya、Wenjian Dong、Murali Narayanaswamy、Zhengchun Liu、Gaurav Saxena、Andreas Kipf 和 Tim Kraska。为什么 TPC 还不够：Amazon Redshift 集群分析。*VLDB* 基金会会刊，2024 年。URL https://www.amazon.science/publications/ why-tpc-is-not-enough-an-analysis-of-the-amazon-redshift-fleet。[24] Qichen Wang、Bingnan Chen、Binyang Dai、Ke Yi、Feifei Li 和 Liang Lin。Yannakakis+：具有理论保证的实用无环查询求值。*ACM* 数据管理会议录，3 (3):235:1–235:28，2025 年 6 月。doi：10.1145/3725423。URL https://dl.acm.org/doi/10. 1145/3725423。[25] Johannes Wehrstein、Timo Eckmann、Roman Heinrich 和 Carsten Binnig。JOB-Complex：传统与学习式查询优化的挑战性基准测试，2025 年 7 月。URL http://arxiv.org/abs/2507.07471。arXiv:2507.07471 [计算机科学]。