软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007695] [CSTR: 32375.14.jos.007695]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

QRCE: 基于查询路由的鲁棒基数估计方法*

余晓婷, 高锦涛

(宁夏大学 信息工程学院, 宁夏 银川 750021)

通信作者: 高锦涛, E-mail: gaojintao@nxu.edu.cn

摘　要: 基数估计是数据库管理系统 (database management system, DBMS) 查询优化器的核心组件, 其准确度直接影响执行计划质量. 现有学习式基数估计方法虽然在特定场景下优于传统方法, 但面对高度异质查询负载 (如表数量、连接形态及谓词差异巨大) 时, 单一模型难以保持稳定的高精度, 同时, 现有多模型方案缺乏精细的查询级选择机制. 因此, 提出基于查询路由的基数估计方法——QRCE, 核心思想包括: 1) 构建异质、可扩展的 Seq2Seq 模型空间, 集成多种具有不同结构归纳偏置的序列建模架构, 形成候选模型集合; 2) 构建门控混合查询路由器 GMQR, 实现细粒度的、基于查询语义的模型路由, 并体现出面向单条查询的自适应决策特征; 3) 提出一种基于 Q-error 的 (1+ε)-近似标签策略, 将模型选择转化为可监督学习任务, 使系统能够根据每条查询的语义特征自适应地选择最适合的模型. 在 STATS、JOB-light 和 TPC-H 这 3 个基准数据集上的实验结果表明, QRCE 的整体与尾部误差表现显著优于多数基线方法, 尤其在结构复杂的 STATS 上, QRCE 将 Q99 误差降低了约 53%, 有效规避了长尾灾难性偏差; 在 JOB-light 上, 其 Q99 与最优单模型相近, 而在 Q50/Q90/Q95 上保持优势; 在 TPC-H 上 Q99 进一步降低约 16%. QRCE 在长尾查询和复杂连接场景下展现出更强的鲁棒性与适应性.

关键词: 基数估计; 查询路由; Seq2Seq 模型; 鲁棒性

中图法分类号: TP311

中文引用格式: 余晓婷, 高锦涛. QRCE: 基于查询路由的鲁棒基数估计方法. 软件学报. http://www.jos.org.cn/1000-9825/7695.htm

英文引用格式: Yu XT, Gao JT. QRCE: Robust Cardinality Estimation Method Based on Query Routing. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7695.htm

QRCE: Robust Cardinality Estimation Method Based on Query Routing

YU Xiao-Ting, GAO Jin-Tao

(School of Information Engineering, Ningxia University, Yinchuan 750021, China)

Abstract: Cardinality estimation is a core component of the query optimizer in database management systems (DBMS), and its accuracy directly affects the quality of execution plans. Although existing learning-based cardinality estimation methods outperform traditional methods in certain scenarios, they struggle to maintain consistently high accuracy on highly heterogeneous query workloads, where queries may differ significantly in the number of tables, join patterns, and predicates. Moreover, existing multi-model approaches lack a fine-grained mechanism for selecting models at the query level. Therefore, this study proposes a query routing-based cardinality estimation method, QRCE, whose core ideas include 1) constructing a heterogeneous and scalable Seq2Seq model space, integrating multiple sequence modeling architectures with different structural inductive biases to form a candidate model set; 2) building a gated mixture query router, GMQR, to achieve fine-grained model routing based on query semantics and enable adaptive decision-making for individual queries; 3) proposing a Q-error-based (1+ε)-approximate label strategy, transforming model selection into a supervised learning task, enabling the system to adaptively select the most suitable model based on the semantic features of each query. Experimental results on three benchmark datasets, STATS, JOB-light, and TPC-H, show that QRCE achieves significantly better overall and tail error performance than most baseline methods. Particularly on the structurally complex STATS dataset, QRCE reduces the Q99 error by approximately 53%,

* 基金项目: 国家自然科学基金 (62462051); 宁夏自然科学基金 (2025AAC020045)
收稿时间: 2026-01-22; 修改时间: 2026-04-02; 采用时间: 2026-05-11; jos 在线出版时间: 2026-08-12

effectively avoiding long-tail catastrophic bias. On JOB-light, its Q99 is close to that of the best single model, while maintaining advantages at Q50/Q90/Q95. On TPC-H, the Q99 error is further reduced by about 16%. QRCE demonstrates stronger robustness and adaptability in long-tail queries and complex join scenarios.

Key words: cardinality estimation; query routing; Seq2Seq model; robustness

基数估计是数据库管理系统 (database management system, DBMS) 查询优化器中的核心组件, 作用是在查询实际执行之前预测各个中间结果的记录数, 为代价模型提供输入, 从而决定连接顺序、访问路径和算子实现等关键选择[1−3]. 传统方法依赖统计数据, 将底层数据压缩为轻量级摘要, 如直方图、采样和多维统计量[4−7], 然后通过解析公式给出近似基数. 这类方法在数据分布简单、相关性较弱的场景下效果尚可, 但面对复杂相关、高度倾斜分布及多表连接等复杂查询场景时, 可能导致优化器产生严重劣化的执行计划[8−11].

近年来, 学习式基数估计方法在多个基准数据集上展现出优于传统统计方法的性能[3,8]. 现有工作分为 3 类: 1) 数据驱动. 学习原始数据高维联合分布, 通过概率积分、采样或条件查询近似基数[12−14]; 2) 查询驱动. 针对查询空间, 基于历史查询并以真实基数为监督, 学习从查询到基数的映射[15−17]; 3) 混合驱动. 同时利用“数据和查询”两端的信息, 对数据库状态和查询同时进行向量化表示, 并借助注意力等神经结构显式建模二者关系, 在动态工作负载下取得了较好的效果[18−20].

在多模型方面, 近期工作包括: 单一网络架构内部集成多个子模块以提升估计精度与鲁棒性[21,22]; 按连接模板或场景划分模型, 在特定 join 结构或 workload 上训练局部估计器[23]; 从数据库、工作负载粒度出发, 为不同系统推荐合适的学习式基数估计模型[24]. 目前多模型方案聚焦于同构模型内部的集成, 将选择粒度固定在连接模板或数据集这样的较粗层面[23,24].

现有学习式基数估计方法面临 3 个关键挑战: 1) 面对异质查询负载, 单一模型鲁棒性较差, 尾部误差较明显[8,17,25,26]. 2) 现有多模型方案选择粒度较粗, 缺乏查询级精细匹配, 难以针对具体每条查询的语义结构和谓词模式在多种异构模型之间做细粒度的选择从而决定最合适的模型. 现有工作仅在同构模型内部做集成[21], 或按连接模板、场景或工作负载粒度划分模型[23], 在数据库/工作负载级推荐“一个合适的模型”[24]. 3) 缺乏面向查询级模型选择的统一学习范式和稳健监督信号[16,27]. 现有多模型工作大多停留在整体指标层面的模型对比, 或基于启发式规则进行切换[24], 尚未普遍将“给定查询应当选用哪个模型”系统地形式化为一个可监督学习任务.

基于上述问题, 本文提出将基数估计建模为序列学习任务, 以更好地捕捉查询语义的结构化组合特点, 并研究如何在查询级别自动学习“查询-模型”匹配关系, 使每条查询能自适应选择最适合的估计模型, 因此提出基于查询路由的鲁棒基数估计方法 QRCE (robust cardinality estimation method based on query routing). QRCE 针对 SQL 语义设计统一的查询表示, 将每条查询编码为固定维度的向量 $\phi(q)$, 进一步地, 引入共享序列化适配层 S, 将 $\phi(q)$ 转换为定长序列为多模型提供输入, 随后实例化多种具有不同结构归纳偏置的序列模型, 形成一组共享输入与训练目标, 但内部结构各不相同的候选模型集合, 使不同模型适应长连接、极端选择率和长尾查询等子空间.

为驱动查询级模型选择, QRCE 进一步利用候选模型在训练查询上的 Q-error 行为构造显式的“查询-最优/近似最优模型”标签, 将模型选择转化为可监督学习任务. 具体而言, 我们通过比较各候选模型在同一查询上的 Q-error, 识别出严格最优及一组近似最优的模型, 形成多标签数据集. 基于该数据集, 我们构建门控混合查询路由器 GMQR (gated mixture query router), 通过非线性变换与结构感知机制建模查询语义, 并输出对各候选模型的偏好分数. 训练阶段 GMQR 学习查询子空间与模型性能之间的映射; 推理阶段则采用 hard selection 策略, 为每条查询选取得分最高的模型进行基数估计, 从而实现查询级的多模型自适应路由, 并使模型选择过程体现出一定的查询感知动态决策特征.

本工作的主要贡献概括如下.

1) 提出基于查询路由的基数估计方法 QRCE. 该方法将基数估计从传统的“单模型回归问题”重新定义为“多模型集合+查询级模型选择”的两阶段学习. 通过显式引入查询感知的路由机制, 系统性地利用不同结构模型在异质查询子空间上的性能差异, 实现细粒度的模型调度, 并使多模型组合过程由静态集成进一步体现为面向单条查询的自适应决策过程.

2) 设计面向查询级模型选择的监督学习机制. 这包括: (i) 统一的查询语义表示并序列化, 为后续模型提供结构化输入; (ii) 基于 Q-error 的 (1+ε)-近似标签策略, 生成稳健的“查询-最优/近似最优模型”多标签数据集, 从而将模型选择转化为可监督学习任务.

3) 构建门控混合查询路由器 GMQR. 基于多标签数据集进行监督训练, 通过其门控混合结构学习从查询到模型偏好的复杂映射, 从而实现对每条查询的模型选择, 并为查询级模型调度提供统一的决策机制.

4) 基于 STATS、JOB-light 与 TPC-H 基准, 进行系统性实验验证. 结果表明, QRCE 在整体与尾部误差上均优于现有最佳单模型及主流基线, 显著提升了在复杂连接与长尾查询上的鲁棒性, 从而验证了查询级多模型选择机制的有效性.

本文第 1 节概述相关工作. 第 2 节介绍问题定义. 第 3 节介绍 QRCE 方法的查询语义特征并序列化、标签构造与查询路由 GMQR. 第 4 节给出实验设置与评估结果, 并进行消融与行为分析. 最后总结全文.

1 相关工作

● 传统统计方法. 通过构建轻量级数据摘要 (如单列直方图、最常见值列表、抽样统计或有限维度的多维统计量)[5−7,28], 在独立性、均匀性等假设下解析推导选择率[9]与连接中间结果规模. 该类方法实现成熟、在线开销低, 是多数系统的默认方案; 但在复杂相关性、强倾斜分布[8]或多表连接场景下, 摘要对真实联合分布的刻画能力受限, 误差往往在高分位与长尾查询上显著放大[25], 进而影响代价估计与执行计划稳定性.

● 学习式方法. 为突破传统模型的假设限制, 学习式方法尝试从数据或查询中直接学习更灵活的映射关系, 包括: 1) 查询驱动. 基于查询特征学习基数映射, 利用历史反馈对直方图进行校正与自调优[5,29–31], 或更新统计摘要[32]. LW-XGB 和 LW-NN[33]将基数估计视为回归任务. UAE-Q[34]结合深度自回归模型[35]与可微分渐进采样以捕获隐含分布. 基于 KDE 的联接估计器[12]以核密度估计增强多元分布与连接基数估计. NNGP[36,37]进一步刻画不确定性以预测分布的均值与方差. 2) 数据驱动. 侧重建模底层数据分布. 其中一维直方图[38]简单高效但依赖属性独立假设; 多维直方图[4,6,39,40]可建模相关性却仍有分解信息损失, 且难以处理复杂连接. 采样方法[7,28,41,42]支持连接查询, 但在分布复杂或强条件下易出现高方差甚至失效. 基于贝叶斯网络[43–45]用有向无环图表达依赖关系, BayesCard[13]结合概率编程提升推理效率. 近年来深度学习被广泛引入: Naru[46]和 NeuroCard[47]用自回归分解联合分布, DeepDB[12]基于和积网络近似密度, FLAT[48]采用自适应分解的和积网络提升灵活性与精度, FactorJoin[26]结合单表条件分布学习与因子图推断, 提升了连接查询基数估计能力. 近期也出现了跨库预训练/迁移方向的研究[49]. 3) 混合模型则结合查询驱动与数据驱动的优势, 同时利用数据样本与查询负载. Wu 等人[18]以统一深度自回归框架融合无监督数据与监督工作负载. Kipf 等人[41]连接了基本关系信息和查询特征并用多集卷积网络估计. Negi 等人[17]在连接键上构建样本表并从查询中提取信息的技术. ALECE[19]通过联合建模数据特征与查询特征, 提升了 SPJ 查询基数估计的准确性与适应性.

在此基础上, 多模型与模型选择则通过增加“模型层”的结构设计来进一步提升鲁棒性. Fauce[21]在单一架构内部进行深度集成并结合不确定性估计以增强复杂连接场景的稳定性. LocalNN[23]则按连接模板或局部结构对子工作负载进行划分, 并为不同模板训练局部模型以获得更精细的拟合能力. AutoCE[24]则从数据集/场景视角比较多种学习式模型, 并通过“模型顾问”在新数据库或新工作负载到来时推荐合适的单一模型. 总体而言, 现有多模型方案多在数据库级、工作负载级或模板级进行粗粒度选择, 难以在同一数据库的真实负载中针对每条查询的结构差异实现查询级动态匹配; 相比之下, 本文更强调在单条查询层面完成动态模型匹配, 使模型选择过程呈现出更强的查询感知决策特征.

随着深度学习的发展, 序列建模经历了从循环神经网络到注意力机制主导的 Transformer 的重要演进. 循环神经网络类模型通过递归状态在时间维上传播信息, 能够刻画时序依赖, 但在超长序列上易受梯度衰减与计算串行性限制[50,51]. Transformer 以自注意力显式建模任意位置间交互, 显著提升长距离依赖建模能力与并行效率, 逐渐成为主流序列架构[52]. 与此同时, 卷积序列模型以及注意力与状态更新相结合的混合架构也在不同任务中展现出

竞争力, 形成与主线并行的发展方向[53,54]. 为降低长序列下注意力的二次复杂度, 研究者提出线性注意力及低秩/核化近似等方法, 将计算与存储开销近似降为线性, 从而提升可扩展性[55]. 近年来, 状态空间模型及其变体通过递推状态更新与结构化参数化, 在保持线性复杂度的同时增强长上下文建模能力, 形成了与注意力机制并行的重要技术路线[56–58].

面对高度异质的工作负载, 查询在表数、连接形态与谓词选择率上差异显著, 使得单一模型难以在所有查询子空间内保持稳定精度[25,26]. 与此同时, 不同神经网络架构在长距离依赖建模、序列表达与少见模式记忆等方面存在差异化归纳偏置[51–57], 除了关系型查询, 序列模型近期也成功应用于图数据库的路径查询估计中[58]. 因此, 本文提出基于查询路由的基数估计方法 QRCE: 通过轻量的查询级路由器在推理时动态调度一组异构的序列建模候选模型, 使每条查询自适应匹配更合适的估计模型, 从而在可控开销下系统提升复杂与长尾查询的估计鲁棒性.

2 问题定义

设关系数据库实例为 $D=\{R_1,R_2,\ldots,R_N\}$. 针对常用 SPJ COUNT 类查询, 规范化定义为:

$$q:\ \text{SELECT COUNT}(*)\ \text{FROM}\ R_{i_1},R_{i_2},\ldots,R_{i_n}\ \text{WHERE}\ J\wedge F \tag{1}$$

其中, $\{R_{i_1},R_{i_2},\ldots,R_{i_n}\}\subseteq D$ 为参与查询的表集合, J 为等值连接谓词集合, F 为过滤谓词集合. 查询 q 在 D 上的真实基数记为 $card(q)=|q(D)|$. 基数估计任务旨在不执行查询的前提下, 给出估计值 $\widehat{card}(q)$ 以逼近 $card(q)$.

针对 q, 采用统一的查询语义空间表示. 记 $\phi(q)\in\mathbb{R}^d$ 为查询 q 的语义特征向量, 它编码表参与、连接结构、谓词类型及其归一化取值区间等信息. 为适配序列模型的输入要求, 我们引入共享序列化适配层 $S:\mathbb{R}^d\to\mathbb{R}^{L\times d_s}$, 将 $\phi(q)$ 转换为长度为 L、维度为 d_s 的定长序列 $Z(q)=S(\phi(q))$, 作为后续候选模型与路由器的统一输入.

在序列化特征空间上, 定义一组已训练好的、结构各异的候选模型集合 $M=\{m_1,m_2,\ldots,m_k\}$. 每个模型 $m_k:\mathbb{R}^{L\times d_s}\to\mathbb{R}$ 接收 $Z(q)$, 为处理基数可能为 0 的情况并改善数值稳定性, 我们对真实基数进行对数变换, 输出对基数对数 $y(q)=\log(1+card(q))$ 的估计 $\hat{y}_k(q)=m_k(Z(q))$, 对应的基数估计为 $\widehat{card}(q)=\exp(\hat{y}_k(q))-1$. 估计精度采用 Q-error[59]衡量:

$$Q_k(q)=\max\left(\frac{\widehat{card}_k(q)}{card(q)},\frac{card(q)}{\widehat{card}_k(q)}\right) \tag{2}$$

给定查询 q 及其序列化表示 $Z(q)=S(\phi(q))$, QRCE 的目标是学习一个路由函数, 从候选集合 M 中为 q 动态选择最合适的模型. 引入门控混合查询路由器 (GMQR) $g_\theta:\mathbb{R}^{L\times d_s}\to\Delta^{K-1}$, 其中 Δ^{K-1} 为 K 维概率单纯形. GMQR 以 $Z(q)$ 为输入, 输出路由权重向量, 满足 $\alpha_k(q)\in[0,1]$ 且 $\sum_{k=1}^{K}\alpha_k(q)=1$, 表示选择模型 m_k 的偏好概率. 在线推理采用 hard selection 路由策略: $k^*(q)=\arg\max_k\alpha_k(q)$, 最终输出被选模型的估计结果. 本文将 GMQR 基于查询表示 $Z(q)$ 输出模型偏好分布, 并通过 hard selection 完成候选模型选择的过程称为查询路由, 即一种面向基数估计任务的查询级模型选择机制. 其核心是在查询级别建立查询特征与模型偏好之间的对应关系, 从而使不同语义结构、连接模式和谓词特征的查询能够匹配到更适合的估计模型. 从机制上看, 上述过程可理解为依据当前查询表示在候选模型空间中完成一次动态选择, 从而使 QRCE 具备面向单条查询的自适应决策特征.

3 QRCE 框架

QRCE 由离线训练与在线推理两阶段组成 (详见图 1). 离线阶段对训练查询进行解析与特征工程, 将每条查询编码为统一的语义向量 $\phi(q)$, 并通过共享序列化适配层 S 将 $\phi(q)$ 映射为定长序列 $Z(q)$, 以统一输入接口并行训练多种序列基数估计模型, 形成候选集合 M. 随后在训练集上统计各候选模型的 Q-error, 并基于最小 Q-error 与 $(1+\varepsilon)$-近似策略构造多标签 $T^*(q)$, 用于监督训练查询感知路由器 GMQR, 使其以 $Z(q)$ 为输入预测对候选模型的偏好分布. 候选模型与路由器共享同一序列化输入 $Z(q)$, 从而保证训练与推理阶段的接口一致性. 在线阶段对新查询复用同一特征化流程得到 $\phi(q)$, 并进一步序列化为 $Z(q)$; GMQR 基于 $Z(q)$ 输出偏好分布并选择得分最高的模型完

成基数回归. 对外仍保持“输入查询、输出基数”的统一接口, 路由逻辑封装在框架内部. 因而, QRCE 在线阶段形成了“查询感知-模型选择-估计执行”的闭环流程.

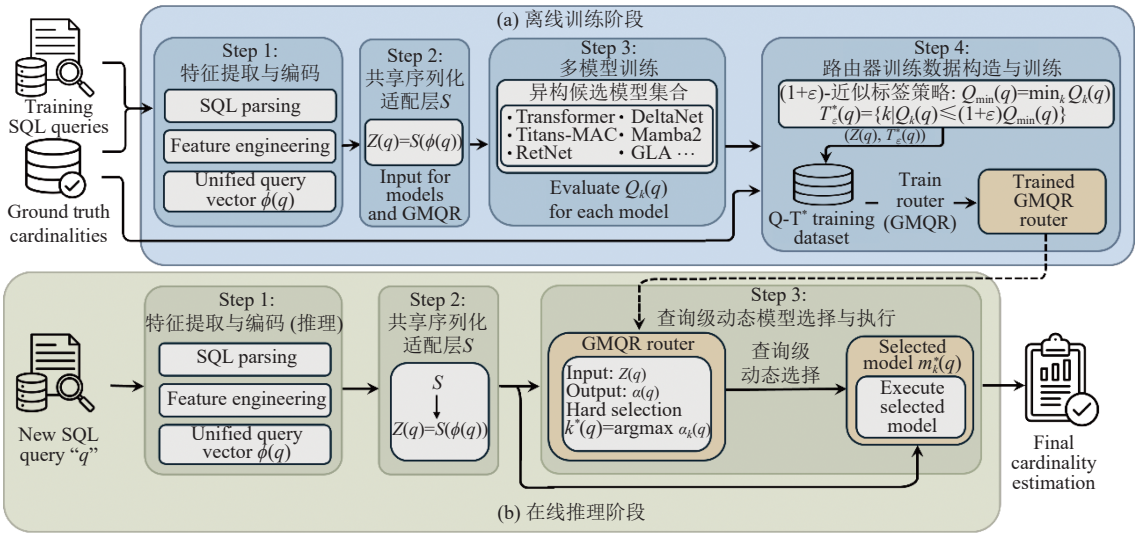

图 1 QRCE 框架

3.1 查询语义特征

针对数据库实例 D, 将每条 q 编码为定长实值向量 $\phi(q)$, 并进一步通过共享序列化适配层 S 转换为定长序列输入以适配不同序列模型. 整体特征化流程 (详见图 2): 左侧为示例 SPJ 查询, 中部为解析得到的 $T(q)$、$J(q)$、$F(q)$, 右侧 4 个模块分别对应 4 类子特征: 连接模式 $\gamma_{\text{join}}(J)$、过滤区间 $\gamma_{\text{range}}(F)$、连接顺序 $\gamma_{\text{order}}(J,\pi(q))$ 和谓词类型 $\gamma_{\text{ptype}}(F)$, 并拼接得到:

$$\phi(q)=[\gamma_{\text{join}}(J),\gamma_{\text{range}}(F),\gamma_{\text{order}}(J,\pi(q)),\gamma_{\text{ptype}}(F)] \tag{3}$$

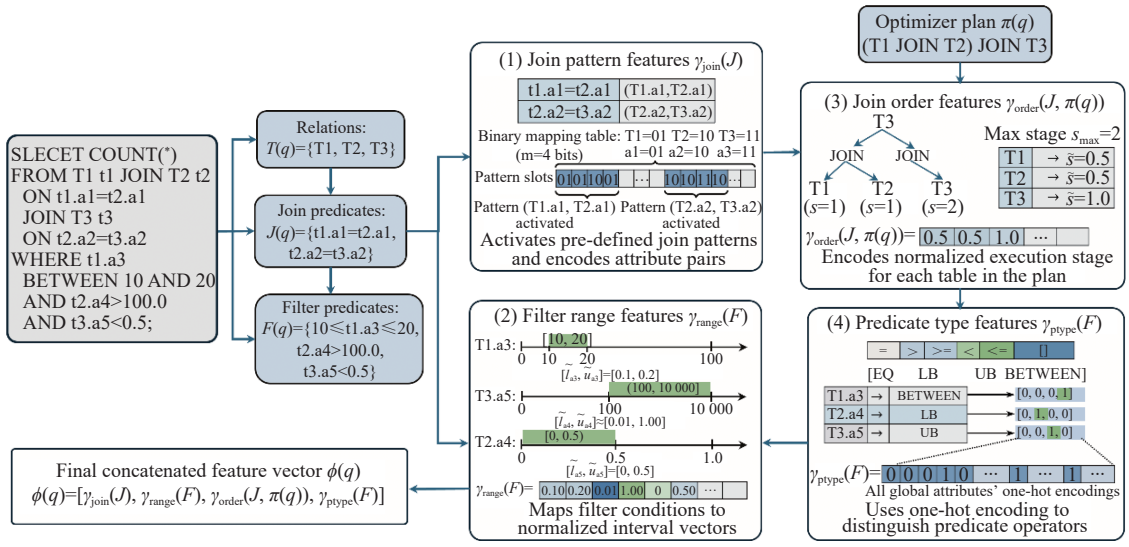

图 2 查询语义特征编码流程图

● 连接模式 $\gamma_{\text{join}}(J)$: 仅依赖等值连接谓词集合 J, 在属性粒度刻画“哪些表通过哪些属性相连”. 为消除等价写法差异 (如不同连接链式展开), 我们将等值谓词规约到属性等价类层面的连接集合, 使语义等价但表述不同的连

接条件映射到一致表示[19].

● 过滤区间 $\gamma_{\text{range}}(F)$: 将作用在数值/有序属性上的过滤谓词统一转写为区间约束, 并基于数据实例统计的属性域做归一化, 得到每个属性的 $(\tilde{l}_k, \tilde{u}_k)$ 端点对, 从而紧凑描述“过滤范围与强度”, 等价于在属性空间中刻画查询切出的搜索区域边界[19].

● 连接顺序 $\gamma_{\text{order}}(J, \pi(q))$: 连接模式无法区分“连接图相同但 join order 不同”的查询. 为捕获与执行计划强相关的结构差异[26], 我们从 PostgreSQL 内置优化器生成的连接计划 $\pi(q)$ 中抽取各基表首次参与连接的阶段编号, 并归一化后形成顺序特征, 用于表达“哪些关系更早进入连接、哪些关系在较后阶段并入”, 该特征仅保留与连接展开相关的轻量信息, 避免重建完整物理计划.

● 谓词类型 $\gamma_{\text{ptype}}(F)$: 在过滤强度相近时, 不同谓词形式仍可能对应不同访问路径与误差模式. 我们将过滤谓词规范化为原子三元组 (A, op, val), 其中 $\text{op} \in \{=, <, >, \leqslant, \geqslant\}$. 对于 BETWEEN…AND…等区间写法, 我们在 $\gamma_{\text{range}}(F)$ 数值特征构建中虽然将其规范化为上下界约束以统一计算区间覆盖率, 但在 $\gamma_{\text{ptype}}(F)$ 类型特征构建中, 我们保留其语义完整性. 随后按谓词类型集合 $\Omega_{\text{ptype}} = \{\text{EQ}, \text{LB}, \text{UB}, \text{BT}\}$ (分别表示等值、下界、上界与区间), 为每个属性 A_k 依据其在 F 中的主导形式确定 $\tau_k \in \Omega_{\text{ptype}}$ (例如同时存在上下界约束则判定为 BETWEEN), 并将 τ_k 编码为 $|\Omega_{\text{ptype}}|$-维 one-hot 向量, 最后按全局属性顺序串联所有向量, 得到 $\gamma_{\text{ptype}}(F)$. 本文工作负载仅包含上述比较谓词, 不涉及 LIKE 等字符串匹配条件.

3.2 共享序列化适配层

为将统一的查询语义向量 $\phi(q)$ 转换为可供多种序列模型共享的输入, 引入轻量的共享序列化适配层 S. 该层将 $\phi(q)$ 按语义块拆解为固定槽位的 token 序列, 把“扁平向量回归”转化为“token 级语义因素交互建模”, 便于序列架构显式学习连接形态、过滤范围、连接顺序与谓词类型之间的依赖. 对单条查询输出:

$$Z(q) = S(\phi(q)) \in \mathbb{R}^{L \times d},\ m(q) \in \{0,1\}^{L} \tag{4}$$

其中, L 为固定序列长度, d 为 token 维度, $m(q)$ 为注意力掩码: 当 $m(q)[i] = 1$ 表示第 i 个槽位为有效 token, 当 $m(q)[i] = 0$ 表示空槽/占位 token, 需在注意力与汇聚操作中屏蔽. 生成的 $Z(q)$ 作为统一输入接口, 可同时供候选序列基数估计模型与路由器使用.

这里, S 将 $\phi(q)$ 切分为语义块 $b \in \{\text{join, range, order, ptype}\}$, 并为每块预定义固定槽位集合 Ω_b ($|\Omega_b| = L_b$), 将块内特征整理为槽位矩阵 $U_b(q) \in \mathbb{R}^{L_b \times r_b}$ (无信息槽位以零向量填充并置掩码为 0). 为统一不同语义块的维度并注入块类别信息, 对每个槽位 $(i = 1, 2, \ldots, L_b)$ 生成 token:

$$\tilde{T}_b(q)[i] = W_b U_b(q)[i] + b_b + e_b,\ W_b \in \mathbb{R}^{d \times r_b},\ b_b, e_b \in \mathbb{R}^{d} \tag{5}$$

其中, e_b 为可学习的块类型嵌入, 用于区分不同语义块.

最后按预设顺序拼接得到:

$$Z(q) = concat(\tilde{T}_{\text{join}}(q), \tilde{T}_{\text{range}}(q), \tilde{T}_{\text{order}}(q), \tilde{T}_{\text{ptype}}(q)),\ L = \sum_{b} L_b \tag{6}$$

由于 L 由固定槽位决定, 缺失信息以零向量占位; 掩码 $m(q)$ 用于屏蔽空槽位置, 从而保证注意力与汇聚仅作用于有效 token. 批量训练时沿批次维度堆叠得到 $Z \in R^{B \times L \times d}$ 及对应 $m \in \{0,1\}^{B \times L}$, 从而保证多模型与路由器输入接口一致并避免重复序列化开销.

3.3 候选模型集合与架构可扩展性

QRCE 对候选模型集合 $M = \{m_1, m_2, \ldots, m_k\}$ 的约束是统一的输入-输出接口: 每个模型接收相同的序列化查询特征 $Z(q) = S(\phi(q))$, 并输出对 $\log(1 + card(q))$ 的标量估计. 基于统一接口, 框架在模型层面引入结构多样性, 以形成具有差异性的候选空间并支持查询级模型选择.

具体实现上, 本文选取 6 种近年来广泛使用的序列建模架构实例化为基数估计模型, 它们在输入输出形式和训练目标上保持一致, 但在序列建模机制和计算特性上存在明显结构差异, 如表 1 所示. Transformer[52]采用全局自注意力, 能够显式建模任意位置间的依赖关系, 但在序列长度和批量规模增大时计算与存储开销随之快速增长;

RetNet[53]和 DeltaNet[55]通过保留或增量更新隐式状态, 将注意力计算约化为近似线性复杂度, 设计初衷在于更高效地处理长序列; GLA[60]采用门控线性注意力结构, 在进一步压缩计算开销的同时保留对关键信息的聚合能力; Mamba2[56,57]引入选择性状态空间建模框架, 以改善对长序列和流式输入的表示能力; Titans-MAC[54]在自注意力框架中加入显式记忆模块, 用于增强对历史模式的存储与检索能力. 整体而言, 这些模型代表了当前若干具有不同归纳偏置和计算特性的序列建模范式. QRCE 的设计不依赖特定模型: 任何满足相同输入-输出规范的新模型均可直接纳入候选集合.

表 1　候选模型结构差异性对比

模型	序列建模机制	结构差异
Transformer	全局自注意力	强全局依赖建模、显式对齐与交互
RetNet	递推/隐状态 (线性化注意力)	“状态汇聚”式长程记忆, 偏流式/递推
Titans-MAC	注意力+显式记忆模块	显式存储/检索模块, 强化历史模式利用
DeltaNet	增量/差分式状态更新	以增量更新捕获序列累积信息
GLA	门控线性注意力	用门控控制信息流, 线性注意力聚合关键token
Mamba2	选择性状态空间模型 (SSM)	状态空间建模长程依赖, 偏顺序/流式表示

3.4　基于 Q-error 的标签构造

为将查询级模型选择转化为可监督学习任务, 我们利用候选模型在训练查询上的 Q-error 行为构造路由标签. 对训练查询 q, 在每个候选模型 m_k 上计算:

$$Q_k(q)=\max\left(\frac{\widehat{card}_k(q)}{card(q)},\frac{card(q)}{\widehat{card}_k(q)}\right),\ k=1,2,\ldots,K \tag{7}$$

并记 $Q_{\min}(q)=\min_k Q_k(q)$. 若采用“纯最优”标注, 可将 $\arg\min_k Q_k(q)$ 作为唯一正类, 但当多个模型误差接近时, one-hot 标签对微小排序扰动较敏感, 稳定性较差[16].

因此, 采用容忍因子 $\varepsilon\geqslant 0$ 的 $(1+\varepsilon)$-近似策略, 将所有满足 $Q_k(q)\leqslant(1+\varepsilon)Q_{\min}(q)$ 的模型共同纳入该查询的 $(1+\varepsilon)$-近似标签集合, 并记为 $T^*_\varepsilon(q)=\{k\mid Q_k(q)\leqslant(1+\varepsilon)Q_{\min}(q)\}$, 以此构造路由训练数据集 $\{(Z(q),T^*_\varepsilon(q))\}$, 其中 $(Z(q)=S(\phi(q))$ 为共享序列化适配层输出的序列化查询特征, 用于后续训练中的感知加权, 使路由器在优化整体误差的同时更关注长尾与复杂场景. 算法 1 给出构造流程.

算法 1. Q-T*标签数据集构造.

输入: $Q_{\text{train}}(card(q),\phi(q))$, $M=\{m_1,m_2,\ldots,m_k\}$, S, ε;
输出: Q-T*.

1. Initialize Q-T* = ∅
2. Foreach $q\in Q_{\text{train}}$ do
3. $Z\leftarrow S(\phi(q))$, $c\leftarrow card(q)$
4. For k=1, 2,…, K do
5. $\hat{y}_k\leftarrow m_k(Z)$, $\hat{c}_k=\exp(m_k(Z))-1$
6. $Q_k\leftarrow\max(\hat{c}_k/c,c/\hat{c}_k)$
7. End for
8. $Q_{\min}\leftarrow\min_k Q_k$
9. $T^*_\varepsilon(q)\leftarrow\{k|Q_k(q)\leqslant(1+\varepsilon)\cdot Q_{\min}\}$
10. Push $(Z,T^*_\varepsilon(q))$ into Q-T*
11. End foreach
12. Return Q-T*

在实践中, Q-T*标签构造在训练查询集合上离线执行. 算法首先将查询特征序列化 (第 2、3 行), 并在所有候选模型上计算对应的 Q-error (第 4–7 行); 随后以最小 Q-error 为基准, 采用 (1+ε)-近似准则构造查询级路由标签 (第 8、9 行), 并将生成的查询-标签对加入训练数据集 Q-T* (第 10 行).

3.5 门控混合查询路由器 GMQR

GMQR 的目标是在给定候选模型集合 M 的前提下, 基于查询的序列化语义表示 $Z(q)=S(\phi(q))\in\mathbb{R}^{L\times d}$ 输出对各候选模型的选择得分与概率分布, 使在该查询上历史评估中 Q-error 近似最优的模型获得更高偏好, 从而实现查询级自适应路由, 同时保持估计器调度接口不变. 如图 3 所示, GMQR 以 $Z(q)$ 为输入, 其中 $Z(q)$ 由 S 按连接结构、谓词范围与谓词类型等语义块序列化为 token 序列. 路由器首先对 token 进行线性投影与归一化得到稳定表征, 随后利用多头自注意力在 token 级显式建模不同语义块之间的交互关系, 并结合残差 MLP 聚合上下文信息, 通过 masked mean pooling 聚合所有有效 token, 得到查询级表示 $h(q)$. 在此基础上, 采用基于 Sigmoid 激活的乘性残差门控对候选模型得分进行特征感知的重标定, 生成最终选择得分 $z(q)$, 并通过温度 Softmax 得到偏好分布 $\alpha(q)$. 推理阶段依据 hard selection 选择候选模型并调度其完成基数估计. 需要强调的是, 推理阶段仅依赖查询特征, 不使用真实基数或 Q-error 信息. 训练阶段采用第 3.4 节提出的 (1+ε)-近似多标签作为监督信号, 根据近似最优模型集合 $T_{\varepsilon}^{*}(q)$, 构造对应的多热监督向量 $t(q)\in\{0,1\}^{K}$: 若 $m_k\in T_{\varepsilon}^{*}(q)$, 则 $t_k(q)=1$; 否则 $t_k(q)=0$. 进一步, 记 GMQR 输出的最终选择得分向量为 $z(q)=[z_1(q),z_2(q),\ldots,z_k(q)]$, 对应的偏好分布为 $\alpha(q)=[\alpha_1(q),\alpha_2(q),\ldots,\alpha_k(q)]$. 一方面, 将多标签归一化为软目标分布 $\tilde{t}(q)$, 并定义交叉熵损失为:

$$L_{\mathrm{CE}}(q)=-\sum_{k=1}^{K}\tilde{t}_k(q)\log\alpha_k(q) \tag{8}$$

用于约束路由器输出分布与软目标分布保持一致; 另一方面, 为扩大“近似最优”与“明显劣质”模型之间的得分间隔, 引入多标签二元交叉熵:

$$L_{\mathrm{BCE}}(q)=-\sum_{k=1}^{K}\left[t_k(q)\log\sigma(z_k(q))+(1-t_k(q))\log(1-\sigma(z_k(q)))\right] \tag{9}$$

其中, $\sigma(\cdot)$ 表示 Sigmoid 函数.

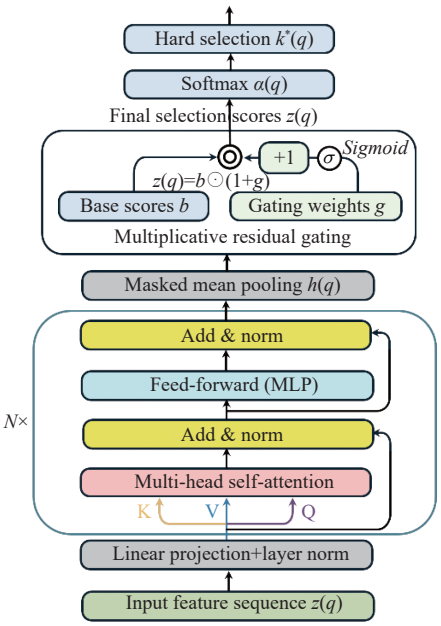

图 3 门控混合查询路由器 GMQR 的结构图

最终, GMQR 的训练目标定义为两类损失线性组合:

$$L(q) = \lambda L_{\mathrm{CE}}(q) + (1-\lambda) L_{\mathrm{BCE}}(q) \tag{10}$$

其中, $\lambda \in [0,1]$ 为平衡系数. 进一步叠加基于 Q-error 的样本权重, 得到加权训练目标:

$$L^{w}(q) = (1 + \log(1 + \min_k Q_k(q)))L(q) \tag{11}$$

其中, $Q_k(q)$ 表示候选模型 m_k 在查询 q 上的 Q-error, $\min_k Q_k(q)$ 表示该查询在所有候选模型中的最优 Q-error. 这种加权方式使复杂查询和长尾样本在训练中获得更高权重, 从而提升 GMQR 在复杂场景下的路由学习质量与稳定性.

4 实验研究

4.1 实验设置

数据集: 我们在 3 个真实关系数据集上评估各基数估计模型及 QRCE 框架, 分别对应社区问答场景的 STATS[61]、电影信息场景的 JOB-light[62], 以及决策支持基准 TPC-H (1 GB)[63]. 其中 STATS 与 JOB-light 基于已有静态工作负载, 直接采用其中标定的测试查询及其子查询作为测试集, 并在相同数据库实例上额外生成训练/验证查询; TPC-H 则在构建 1 GB 数据库实例后随机生成训练与测试查询及其子查询.

● STATS[61]包含 users、posts、postLinks、postHistory、comments、votes、badges、tags 共 8 张关系表, 测试集来自公开静态工作负载标定查询及其子查询.

● JOB-light[62]为 IMDB 数据集的子集, 包含 cast_info、movie_info、movie_companies、movie_keyword、movie_info_idx、title 共 6 张表, 查询仅涉及主外键 (PK-FK) 连接, 连接结构相对规整.

● TPC-H (1 GB)[63]包含 customer、lineitem、nation、orders、part、partsupp、region、supplier 共 8 张关系表. 为与现有学习式方法设置一致, 移除各表字符串类型 comment 列, 仅保留数值或可序属性.

在连接结构上, JOB-light 查询仅包含 PK-FK 连接, 而 STATS 与 TPC-H 还涉及多对多连接, join 结构更为复杂. 3 个数据集均被导入同一 PostgreSQL 实例, 通过实际执行查询获取真实基数, 所有候选模型和 QRCE 框架的训练与评估均在该环境下进行, 各数据集规模与工作负载统计如表 2 所示.

表 2　数据集与工作负载统计

数据集	表数	属性数	记录数	测试查询	测试子查询	训练查询 (额外生成)	连接特性
STATS	8	43	1 029 842	146	2 603	1 000	含多对多连接, 结构较复杂
JOB-light	6	14	62 121 470	208	3 248	2 000	仅PK-FK连接, 结构规整
TPC-H	8	46	8 661 245	123	1 554	2 000	移除comment列, 含多对多连接

对比方法: 为更系统地比较本文方法与现有工作的差异, 我们参考已有基准、评测工作以及相关综述中常见的分类方式, 选取具有代表性的实现作为对比基线. 所选方法覆盖传统统计、查询驱动、数据驱动、混合以及多模型等不同技术路线, 能够从不同角度反映当前基数估计方法的发展特点与性能差异. 为保证比较公平性, 本文纳入实验的方法均在统一的 PostgreSQL 环境、统一数据集划分、统一真实基数获取方式以及统一评价指标 (Q50、Q90、Q95、Q99、Building Time、Latency 与 E2E Time) 下进行评测. 各基线方法具体如下.

● PG[64]. PostgreSQL 内置的一维直方图估计器, 基于单列直方图、频率表与采样, 并在独立性与均匀性假设下推导选择率与中间结果规模, 用作传统优化器基线.

● MSCN[41]. 典型查询驱动神经方法, 将表、连接与谓词编码为多集合特征, 并通过多集合卷积聚合后回归基数.

● NNGP[36]. 基于 neural network Gaussian process[37]的学习式方法, 在特征空间中同时建模预测均值与不确定性, 通过高斯过程后验给出基数估计.

● FactorJoin[26]. 因子分解式估计方法, 将多表联合分布分解为局部因子并结合连接统计, 在因子图上推理多表连接查询基数.

● NeuroCard[47]. 数据驱动联合分布方法, 基于深度自回归模型[35]学习高维分布, 并在评估阶段通过概率计算得到查询基数.

● DeepDB[12]. 典型的数据驱动概率建模方法, 利用和积网络 (sum product network, SPN) 学习数据库属性间的层次化概率分布, 并通过概率推理实现查询选择率与基数估计.

● ALECE[19]. 注意力机制驱动的学习式基数估计方法, 将查询与相关数据分布信息进行注意力检索与聚合以输出基数估计, 在分布变化下具有较好的稳健性.

● QRCE (ours). 本文提出的基于查询路由的基数估计: 在统一查询语义表示上训练 6 个候选模型 (Transformer、RetNet、Titans-MAC、DeltaNet、GLA、Mamba2), 并以基于 Q-error 的 $(1+\varepsilon)$-近似标签训练 GMQR 路由器, 实现查询级模型选择与基数输出.

此外, 我们还引入基于真实基数的理想基线 Optimal 进行对比: 在该设置下每条查询的估计值等于真实基数, 用于刻画基数估计在当前工作负载下所能达到的最佳性能上界. 表 3 总结了不同方法的分类与关键特性. 需要说明的是, 本文纳入的 MSCN、NNGP、FactorJoin、NeuroCard、DeepDB 与 ALECE 分别代表了查询驱动、数据驱动与混合类学习式基数估计中的典型技术路线. 对于近年来提出的 Fauce、AutoCE 等多模型方法, 考虑到公开实现, 本文未将其纳入统一评测协议下的定量比较, 而是在相关工作中讨论其方法特点与本文的差异.

表 3　不同方法的分类与关键特性

方法	类别	输入信息	建模方式	模型形态
PG[64]	传统统计	统计摘要	直方图/采样	单模型
MSCN[41]	查询驱动	查询	多集合卷积回归	单模型
NNGP[36]	查询驱动	查询	高斯过程/贝叶斯建模	单模型
FactorJoin[26]	数据驱动	数据	因子分解/连接统计	单模型
NeuroCard[47]	数据驱动	数据	深度自回归分布建模	单模型
DeepDB[12]	数据驱动	数据	SPN/概率分布建模	单模型
ALECE[19]	混合	查询+数据	注意力联合建模	单模型
QRCE (ours)	多模型	查询	路由式多模型估计	多模型

评价指标: 我们通过以下指标来评估每个 CardEst 方法的有效性.

● E2E Time: 端到端执行时间. 在评测阶段, 将某一方法对所有测试子查询给出的基数估计注入 PostgreSQL 优化器, 由其生成执行计划并完成查询执行, 统计全部测试查询的总执行时间. 该指标直接反映基数估计对计划质量与实际执行性能的综合影响, 是核心的系统级指标.

● Q-error[59]: 刻画估计基数与真实基数之间的相对乘性误差. 对任意 (子) 查询 q, 记真实基数为 $c(q)$, 预测基数为 $\hat{c}(q)$, 则:

$$Q(q)=\max\left(\frac{\hat{c}(q)}{c(q)},\frac{c(q)}{\hat{c}(q)}\right) \tag{12}$$

我们报告 Q-error 的 50% (中位数)、90%、95%、99% 分位数 (分别记为 Q50、Q90、Q95、Q99), 用于分别反映典型查询的精度以及尾部误差 (长尾与极端查询) 的稳定性.

● Building Time: 构建时间, 反映离线准备开销. 对于查询驱动方法, 该指标对应模型训练与 Q-T*标签生成的耗时; 对于数据驱动方法, 对应数据摘要/概率模型的构建耗时. 对于我们的 QRCE, 该指标统计 6 个候选单模型的离线训练、基于 Q-error 生成 Q-T 标签以及路由器 (GMQR) 的离线训练所需的总耗时.

● Size: 模型/摘要在推理阶段的存储开销, 包括参数与必要统计信息的总大小, 用于衡量部署成本.

● Latency: 估计时延, 定义为每条测试 (子) 查询在模型/路由器已加载的前提下, 从输入特征到输出基数预测的平均推理耗时, 用于衡量在线开销; 不包含模型加载与离线特征/标签构建时间, 且对多模型方法包含路由决策与被选中模型的一次前向推理.

本文当前主要通过 Building Time、Size 与 Latency 分别从离线构建、部署存储与单次在线推理这 3 个角度

对方法成本进行基础量化, 并未将并发推理内存占用与批量在线吞吐作为独立指标单独报告.

实验设计: 本实验旨在系统评估 QRCE 框架. 为清晰地呈现结果, 我们首先在第 4.3 节展示其整体性能对比. 随后, 为深入理解性能背后的原因, 我们在第 4.4 节聚焦于核心的查询路由机制, 分析 GMQR 的决策质量; 并在第 4.5 节通过消融实验, 分别验证路由器结构和多模型策略的贡献. 这一设计完整呈现了方法有效性的同时, 也系统解释了其性能提升的来源.

4.2 训练设置

所有模型在第 3 节定义的统一查询特征空间上训练. 对每条查询 q, 先提取语义向量 $\phi(q)$, 并经共享序列化适配层得到 $Z(q) = S(\phi(q))$. 回归目标为 $y(q) = \log(1 + card(q))$. 所有实验均在同一服务器平台完成 (Intel Xeon CPU、NVIDIA RTX 4090 GPU).

候选模型训练: 候选集合包含 Transformer、RetNet、Titans-MAC、DeltaNet、GLA 与 Mamba2 这 6 个模型共享相同的序列化输入 $Z(q)$ 与训练目标, 仅内部结构机制不同. 对每个数据集 (STATS、JOB-light、TPC-H), 我们在训练集上并行训练 6 个模型, 统一采用 AdamW, 初始学习率为 3×10^{-4}, batch size 为 64, 最多 100 个 epoch, 并结合 warmup、学习率调度与梯度裁剪. 训练过程中在验证集上监控平均 Q-error, 采用早停并选择验证集平均 Q-error 最小的 checkpoint 作为最终模型.

GMQR 训练: 在候选模型收敛后, 按照第 3.4 节的 (1+ε)-近似策略, 我们在训练/验证查询上离线运行全部候选模型, 得到每条查询的 Q-error 向量并构造路由数据集 Q-T*: 样本由 $Z(q)$ 与多标签 $T^*(q)$ 组成. 路由器采用第 3.5 节的 GMQR 结构 (门控 MLP 与多头自注意力机制, 其 3 个隐藏层维度依次为 1 024、512 和 256, 注意力头数为 8), 输出端使用温度 $\tau = 0.5$ 的 Softmax 得到偏好分布. GMQR 训练使用 AdamW, 初始学习率 2×10^{-4}, batch size 128, 最多 100 个 epoch, 并采用短 warmup 与学习率调度. 损失为软标签交叉熵与多标签 BCE 的加权和, 并基于候选集合上的 Q-error 对样本加权, 以提升长尾与复杂查询的路由质量; 默认 ε=0.20.

4.3 整体性能分析

表 4 给出了各方法在 STATS、JOB-light 与 TPC-H 这 3 个基准上的端到端执行时间 (E2E Time)、Q-error 分布以及构建/推理开销. 其中, Optimal 表示基于真实基数信息获得的理论最优参考结果; 表 4 及后续表格中加粗数据表示各对比方法中对应评价指标下的最优结果. FactorJoin 的开放实现是针对 STATS 和 JOB-light 数据集硬编码的, 不支持 TPC-H 数据集. 因此, 不在此数据集上测试 FactorJoin. 另外, 在我们实验比较设置中, DeepDB 的比较范围主要限于 PK-FK join 场景, 因此未在 STATS 和 TPC-H 数据集上进行比较. 总体来看, 各类单模型方法在不同基准上各有优势: 传统统计 PG 具有最低的部署与计算开销, 但在尾部误差上显著偏高; 而学习式方法虽在部分指标上表现更优, 但往往在长尾精度、构建成本或推理效率之间存在权衡.

在同样的评测设置下, QRCE 在 3 个数据集上整体优于所有单模型基线. 从 Q-error 分布看, QRCE 在 Q50 和 Q90 上普遍不弱于表现最好的单模型, 在 Q95 和 Q99 上对误差尾部有更明显的压缩效果, 尤其在 STATS 上, 相对于最强单模型 ALECE, 长尾分位的 Q-error 有 52.9% 的下降; 在 JOB-light 上, 其 Q99 与最佳单模型 NeuroCard 处于同一量级, 同时在 Q50 与 Q90 上保持优势; 在 TPC-H 上, QRCE 的 Q99 较 ALECE 进一步降低约 15.8%. 这表明查询级路由机制能有效识别并抑制复杂查询上的极端估计误差, 提升长尾稳定性, 在系统执行效率方面, QRCE 的端到端执行时间 (E2E Time) 与各数据集上的最佳单模型表现基本相当, 整体处于同一数量级. 其中, 在 JOB-light 与 TPC-H 上, QRCE 的 E2E Time 仅比最快的 ALECE 增加了约 2%–3%, 差异几乎可以忽略不计; 在高度异质的 STATS 上, 虽然开销增加了约 7.4%, 但换取了尾部误差的显著降低, 这表明 QRCE 能以可控的时间代价, 有效提升对复杂特别是异质查询的估计鲁棒性.

在开销方面, QRCE 需要维护 6 个候选模型和一个路由器, 模型规模与离线构建时间自然高于单一模型, 但推理延迟仍保持在几十毫秒量级, 与轻量级查询驱动方法 (如 ALECE、MSCN) 处于同一数量级, 远低于 NeuroCard 等重型方法. 需要指出的是, QRCE 在线阶段并非对所有候选模型同时执行推理, 而是先由 GMQR 完成一次路由判断, 再调用被选中的单个候选模型进行估计, 因此其在线推理开销并不是所有候选模型推理成本的简单叠加, 而

主要由“一次路由+一次单模型估计”构成. 综合端到端执行时间、误差分布和开销这 3 项指标, 可以认为: 在离线资源允许的前提下, QRCE 通过适度增加模型规模与训练时间, 换取了在整体 Q-error 尤其是长尾场景上的稳定优势, 同时保持与最优单模型相当的系统执行性能, 体现出查询级多模型选择相对于单模型范式的实际价值.

表 4　整体性能分析

数据集	模型	E2E Time (s)	Q50	Q90	Q95	Q99	Size (MB)	Building Time (min)	Latency (ms)
STATS	PG	12 777	1.8	21.84	106	7 950	—	—	—
	MSCN	15 162	3.85	39.56	99.81	1 273	**1.61**	12.41	0.79
	NNGP	20 181	8.1	694	3 294	2.3×10^5	9.56	**0.68**	21.28
	FactorJoin	50 179	5.41	150.52	666.59	2.1×10^5	1.86	23.7	**0.55**
	NeuroCard	19 847	2.91	192	1 511	1.5×10^5	121.88	27.77	40.16
	ALECE	**7 704**	1.47	4.86	9.11	56.98	22.31	6.44	8.64
	QRCE	8 273	**1.40**	**2.93**	**7.76**	**26.84**	134.44	31.39	12.47
	Optimal	7 622	1	1	1	1	—	—	—
JOB-light	PG	19 820	1.7	9.41	16.25	50.19	—	—	—
	MSCN	24 829	10.94	200	396	2, 741	**1.56**	31.68	**1.04**
	NNGP	36 290	6.66	121	414	1.5×10^5	32.44	**0.88**	30.35
	FactorJoin	35 886	13.86	348.96	1, 027.19	4 794.8	5.59	78.45	3.2
	NeuroCard	18 153	1.37	4.16	6.35	15.22	49.96	9.83	16.76
	DeepDB	26 636	11.31	663	6 328	5.4×10^5	11.52	11.93	11.92
	ALECE	**18 015**	1.44	3.23	5.75	47.28	22.25	5.88	7.32
	QRCE	18 426	**1.25**	**2.74**	**4.80**	**17.52**	149.75	70.92	14.38
	Optimal	17 939	1	1	1	1	—	—	—
TPC-H	PG	12 217	1.23	3.95	6.22	11.33	—	—	—
	MSCN	11 893	4.29	36.76	83.76	321	**1.58**	24.17	**0.8**
	NNGP	13 183	31.89	1 052	2 104	11 811	32.44	**0.91**	39.66
	NeuroCard	12 020	1.09	2.97	395	450	850.53	44.39	35.98
	ALECE	**8 717**	**1.24**	**2.36**	6.43	12.75	22.34	7.15	10.31
	QRCE	9 070	1.28	2.96	**6.26**	**10.73**	128.98	57.51	20.63
	Optimal	8 706	1	1	1	1	—	—	—

4.4 GMQR 路由行为分析

为理解 QRCE 在第 4.3 节所展现的性能表现, 本节将重点分析其核心组件——GMQR 路由器的决策行为. 我们将系统评估 GMQR 的决策质量, 包括容忍因子 ε 的选择如何影响路由监督信号的稳定性、在不同设定下的路由命中率, 以及面对不同连接复杂度查询时的模型选择偏好, 从而验证该路由器能否实现精准、自适应的查询-模型匹配, 并说明其查询感知决策机制对整体性能提升的支撑作用.

4.4.1 (1+ε)-近似标签中的容忍因子选择

为确认 (1+ε) 近似标签中容忍因子 ε 对路由器落入近似最优模型集合的命中表现的影响. 我们在 STATS、JOB-light 与 TPC-H 上对 ε 进行扫描, 并统计路由器输出是否落入“近似最优模型集合”的命中率. 根据第 3.4 节的标签构造, 查询 q 在容忍因子 ε 下对应的 (1+ε)-近似标签集合记为 $T_\varepsilon^*(q)$. 设 $\hat{k}(q)$ 为路由器对查询 q 的最终选择, 则定义 ε 下的集合命中率为:

$$Hit@\varepsilon = \frac{1}{|Q|}\sum_{q\in Q} A[\hat{k}(q) \in T_\varepsilon^*(q)] \tag{13}$$

其中, $A[\cdot]$ 为指示函数, 表示路由选择是否命中 (1+ε)-近似标签集合. 当 ε=0 时, $T_0(q)$ 退化为仅包含严格最优模型 $k^*(q) = \arg\min_k Q_k(q)$ 的集合, 因此 $Hit@0$ 等价于“选中严格最优模型的查询占比”. 如图 4 所示, 随着 ε 从 0 增大, 3 个数据集的 $Hit@\varepsilon$ 在小 ε 区间均显著上升, 表明适度放宽严格最优判定能够降低多模型误差接近时标签对细微

排序扰动的敏感性, 从而提升监督信号的稳定性与可学习性. 当 ε 接近 0.2 时, 各数据集的命中表现达到较高水平并开始趋于饱和. 其中在 ε=0.20 设置下, STATS、JOB-light 和 TPC-H 数据集的 *Hit*@ε 分别达到 0.94、0.90 和 0.89. 继续增大 ε 带来的增益有限, 同时近似最优集合的扩张会削弱监督区分度. 综合增益与判别性, 本文在 3 个基准上统一采用 ε=0.20 作为默认设置用于后续实验.

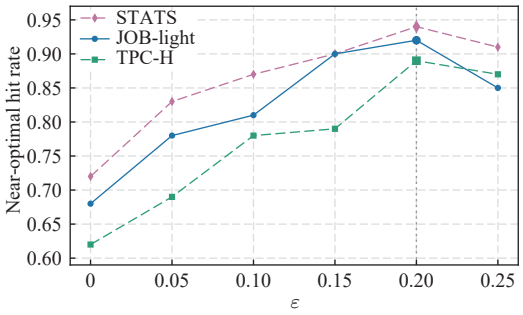

图 4　GMQR 命中率随 ε 的变化

4.4.2　路由命中率

为评估 GMQR 路由决策与监督标签的一致性, 我们在默认 ε=0.2 下对比了严格最优命中率 *Hit*@0 与近似最优命中率 *Hit*@0.2. 如表 5 所示, 在 3 个数据集上, 采用 (1+ε)-近似策略后路由命中率均有显著提升: STATS 提升约 30.6%, JOB-light 提升约 35.3%, TPC-H 提升约 43.5%. 结果表明, 路由器能够将大多数查询分配到误差接近最优的候选模型上, 验证了 (1+ε)-近似策略可有效增强监督信号的稳定性与可学习性.

表 5　GMQR 路由命中率: 严格最优与近似最优对比 (ε=0.2)

数据集	样本数量	*Hit*@0	*Hit*@0.2	Δ*Hit*	Δ*Hit*/*Hit*@0 (%)
STATS	2 749	0.72	0.94	0.22	30.6
JOB-light	3 456	0.68	0.92	0.24	35.3
TPC-H	1 677	0.62	0.89	0.27	43.5

注: Δ*Hit*=*Hit*@0.2–*Hit*@0, Δ*Hit*/*Hit*@0为相对提升比例

4.4.3　不同连接规模下的路由选择偏好

为进一步了解 GMQR 的路由行为, 我们按查询参与关系表的数量将子查询分组 (1–2、3–4、≥5 张表), 并统计 6 个候选模型被选中的相对频率. 为使路由偏好与结构复杂度的关联更清楚, 我们选择在连接模式异质性最强、长尾误差最显著的 STATS 数据集上进行分析; 而在 JOB-light (PK-FK 规整) 与 TPC-H (模板化更强) 上做该分析更容易引入结构差异带来的混淆. 如表 6 所示, 对应的分组柱状图见图 5. 结果显示: 在仅包含 1–2 张表的简单查询上, Transformer 与 RetNet 的被选比例最高, 其余 4 个模型份额相对较小; 当参与表数量增加到 3–4 张时, 6 个模型的使用趋于均衡, 其中 Titans-MAC、Mamba2 以及部分场景下的 DeltaNet 选中比例明显上升, 而 GLA 和 DeltaNet 整体上更集中于中等连接规模的查询; 对于包含≥5 张表的长链查询, Titans-MAC 与 Mamba2 的选中频率进一步提高, Transformer 的占比则明显下降.图 5 中不同连接规模下各模型柱体高度的变化, 直观体现出 GMQR 能够感知查询的连接复杂度, 并动态调整其路由策略: 对于更易出现严重估计偏差的长链、多表查询, 路由器更倾向于调用具备显式记忆 (Titans-MAC) 或状态空间建模 (Mamba2) 能力的模型. 这直接解释了第 4.3 节中 QRCE 在 Q90/Q95/Q99 等尾部指标上的显著优势, 其核心在于通过查询感知的路由, 系统性地为复杂查询匹配了更稳健的估计器.

表 6　STATS 测试集上按参与关系数量分组的模型选择分布

连接关系	Transformer	RetNet	Titans-MAC	DeltaNet	GLA	Mamba2
1–2 joins	0.36	0.24	0.10	0.11	0.11	0.08
3–4 joins	0.27	0.21	0.16	0.12	0.13	0.11
≥5 joins	0.15	0.17	0.24	0.10	0.14	0.20

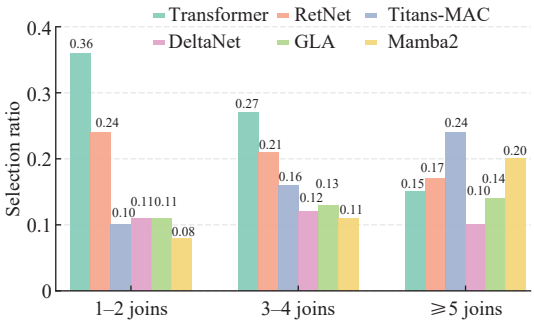

图 5 STATS 测试集上按参与关系数量分组的模型选择分布

4.5 消融实验

为验证 QRCE 框架中各关键设计的有效性, 我们在 3 个数据集上进行了系统的消融研究. 所有实验均遵循统一设置: 固定候选基数估计模型集合及其参数、查询语义特征表示方式, 并在推理阶段采用相同的 hard selection 决策规则; 路由器的训练保持相同的数据划分与优化策略, 仅对目标消融因素进行变更. 本节主要从两个核心方面展开分析: 路由器结构 (第 4.5.1 节, GMQR vs. MLP) 与集成策略有效性 (第 4.5.2 节, 单模型 vs. QRCE) 的验证.

4.5.1 路由器结构消融: GMQR vs. MLP

在上述消融实验总体设置保持一致的前提下, 我们进一步评估路由器结构对 QRCE 查询级模型选择的影响: 在 3 个数据集上对路由模块进行结构消融, 仅将默认的 GMQR 路由器替换为标准 MLP 路由器进行对比. 结果如表 7 所示, 采用 GMQR 的 QRCE 在 *Hit*@0 与 *Hit*@0.2 上相较 MLP 提升约 15%–20%, 并在 Q95/Q99 等尾部 Q-error 上体现出更明显的优势, 表明结构化路由器能够更稳定地将查询分配到严格最优或近似最优的候选模型集合; 同时, 这一优势进一步体现在端到端误差上, GMQR 的 Q-error 分位整体优于 MLP, 且差距在 Q95/Q99 等尾部指标上更为明显. 总体而言, 在不改变候选模型与训练设定的条件下, 引入 GMQR 能提升 QRCE 的路由决策质量, 并将其转化为更稳健的基数估计误差表现, 尤其增强了长尾查询场景下的鲁棒性, 从而验证了 GMQR 结构对 QRCE 查询级模型选择的有效性与必要性.

表 7 GMQR 与 MLP 路由器对 QRCE 性能影响对比

数据集	路由器	*Hit*@0	*Hit*@0.2	Q50	Q90	Q95	Q99
STATS	MLP	0.58	0.76	3.14	6.56	17.38	60.12
	GMQR	**0.72**	**0.94**	**1.40**	**2.93**	**7.76**	**26.84**
JOB-light	MLP	0.54	0.73	2.72	6.88	20.49	65.67
	GMQR	**0.68**	**0.92**	**1.25**	**2.74**	**4.80**	**17.52**
TPC-H	MLP	0.51	0.70	2.59	7.45	18.76	39.94
	GMQR	**0.62**	**0.89**	**1.28**	**2.96**	**6.26**	**10.73**

4.5.2 集成策略有效性验证: 单模型 vs. QRCE

本节通过消融对比评估了 QRCE 与各单一候选模型在基数估计任务上的表现, 表 8 显示, 在 STATS、JOB-light 与 TPC-H 这 3 个基准上, QRCE 的优势主要体现在尾部误差的显著降低: 在 Q90/Q95/Q99% 上, QRCE 均稳定优于任意单模型, 并显著压缩极端误差 (例如 STATS 的 Q99, QRCE 为 26.84, 而最佳单模型为 84.98). 与此同时, QRCE 在中位数 (Q50) 上与最优单模型基本持平, 说明其增益主要来自对复杂查询与长尾场景的误差抑制, 而非对多数常规查询的额外改动. 该结果充分验证了查询级路由策略的有效性: 通过为每条查询动态选择其最匹配的模型, QRCE 系统性地规避了单一模型在其不擅长的查询子空间上可能产生的灾难性偏差, 从而实现了尾部估计鲁棒性的根本性提升.

表 8　候选单模型与 QRCE 的 Q-error 分位数对比

数据集	模型	Q50	Q90	Q95	Q99
STATS	Transformer	1.53	11.38	15.39	100.87
	Titans-MAC	1.42	9.01	19.18	93.47
	RetNet	1.49	12.03	18.26	84.98
	DeltaNet	1.58	11.40	21.31	99.94
	GLA	1.67	9.24	20.57	86.59
	Mamba2	1.54	10.13	16.96	91.91
	QRCE	**1.40**	**2.93**	**7.76**	**26.84**
JOB-light	Transformer	1.33	6.04	14.53	69.47
	Titans-MAC	1.32	6.65	14.17	59.69
	RetNet	1.39	6.85	15.87	63.42
	DeltaNet	1.28	6.23	16.43	66.38
	GLA	1.41	6.04	14.71	58.40
	Mamba2	1.24	5.79	12.48	55.25
	QRCE	**1.25**	**2.74**	**4.80**	**17.52**
TPC-H	Transformer	1.34	5.95	10.67	36.46
	Titans-MAC	1.27	5.10	9.28	29.73
	RetNet	1.36	6.49	11.77	34.79
	DeltaNet	1.38	6.12	11.83	35.84
	GLA	1.30	4.96	8.98	26.91
	Mamba2	1.25	5.36	9.35	28.58
	QRCE	**1.28**	**2.96**	**6.26**	**10.73**

4.6 动态环境下的适用边界与扩展讨论

需要说明的是, GMQR 的监督信号来自候选模型在训练查询上的 Q-error 行为, 并基于最小 Q-error 与 $(1+\varepsilon)$-近似策略构造“查询-最优/近似最优模型”标签集合. 该设计能够在当前训练分布下为查询级模型选择提供直接监督, 从而支持 GMQR 学习查询子空间与候选模型性能之间的匹配关系.

本文当前实验主要在统一离线训练与测试协议下验证查询级路由机制的有效性, 尚未专门考察数据分布或查询模式变化条件下 QRCE 的表现. 需要指出的是, 当底层数据分布、连接结构或谓词模式相较训练阶段发生明显变化时, 原有离线标签所刻画的“查询-模型”对应关系可能不再完全适用, 从而带来一定的路由偏差风险. 对此, 后续可考虑结合新到达查询的真实反馈, 周期性更新候选模型的 Q-error, 并相应刷新近优标签与路由器参数; 也可结合工作负载变化情况, 仅对发生明显变化的查询子空间进行局部重标注与局部重训练, 以降低整体维护成本; 同时, 还可引入基于路由置信度或候选模型分歧的回退机制, 在低置信度情形下回退到更稳健的候选模型或默认估计器. 上述方向将为 QRCE 在动态环境中的持续稳定运行提供支持.

4.7 部署成本与工程可行性讨论

综合前文评价指标与整体性能分析可以看出, QRCE 的额外开销主要体现在候选模型集合与路由器带来的离线训练和模型维护成本, 而其在线阶段仍保持了较低的单次推理开销. 因此, 从工程实现角度看, QRCE 更适合作为一种“以适度增加离线构建与模型维护代价, 换取复杂查询和长尾场景下更强鲁棒性”的查询级基数估计框架. 在实际部署中, QRCE 对外仍保持与单模型方法一致的基数估计接口, 即输入查询表示、输出基数估计结果, 因此无需改变优化器原有的调用方式. 候选模型训练、Q-T 标签生成以及路由器训练均可在离线阶段完成, 不影响在线优化流程. 对于资源受限场景, 还可进一步采用候选模型裁剪、较小规模候选池部署以及低置信度回退等方式控制部署成本: 一方面, 可基于目标工作负载特点仅保留性能互补性较强的少量专家, 以压缩模型存储与维护开销; 另一方面, 在延迟或资源约束更强的场景下, 可结合路由置信度在低置信度情形下回退到默认估计器或更稳健的轻量模型, 从而降低复杂部署带来的额外代价. 需要说明的是, 候选池裁剪主要面向工程化部署需求, 本文当前未进一步单独评估不同裁剪比例下的性能-成本折中关系, 而将其作为后续部署优化方向.

5 总 结

本文面向高度异质查询负载, 提出一种基于查询路由的基数估计方法 QRCE. 区别于传统“单模型覆盖全部查询”的范式, QRCE 在统一输入输出接口下组织多种序列建模架构的候选基数估计模型, 并通过查询级路由机制对每条查询进行动态模型选择, 从而提升估计的稳定性. 为支撑查询级选择, 我们构建了统一的查询语义表示与共享序列化适配层, 将连接结构与谓词信息映射到可复用的特征空间; 基于候选模型在训练查询上的 Q-error 行为, 提出 $(1+\varepsilon)$-近似标签构造策略, 形成查询到最优/近似最优模型集合的监督信号; 进一步设计门控混合查询路由器 GMQR, 用以学习查询模式与候选模型误差分布之间的对应关系, 实现可训练的查询级决策. 在 STATS、JOB-light 与 TPC-H 这 3 个基准上的系统评估表明, QRCE 在不显著影响中位误差的前提下, 能够稳定降低高分位误差 (Q90/Q95/Q99), 并在复杂连接与长尾查询场景下显著压缩极端估计偏差, 从而为代价模型提供更可靠的输入. 结果验证在真实异质工作负载下, 引入查询级多模型选择能够有效提升基数估计的鲁棒性与可用性.

References

[1] Yue WJ, Qu WW, Lin K, Wang XL. Machine learning-based cardinality estimation: A survey. Journal of Computer Research and Development, 2024, 61(2): 413–427 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.202220649]

[2] Lan H, Bao ZF, Peng YW. A survey on advancing the DBMS query optimizer: Cardinality estimation, cost model, and plan enumeration. Data Science and Engineering, 2021, 6(1): 86–101. [doi: 10.1007/s41019-020-00149-7]

[3] Sun J, Zhang JT, Sun ZY, Li GL, Tang N. Learned cardinality estimation: A design space exploration and a comparative evaluation. Proc. of the VLDB Endowment, 2021, 15(1): 85–97. [doi: 10.14778/3485450.3485459]

[4] Poosala V, Ioannidis YE. Selectivity estimation without the attribute value independence assumption. In: Proc. of the 23rd Int'l Conf. on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 1997. 486–495.

[5] Bruno N, Chaudhuri S, Gravano L. STHoles: A multidimensional workload-aware histogram. In: Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data. Santa Barbara: ACM, 2001. 211–222. [doi: 10.1145/375663.375686]

[6] Wang H, Sevcik KC. A multi-dimensional histogram for selectivity estimation and fast approximate query answering. In: Proc. of the 2003 Conf. of the Centre for Advanced Studies on Collaborative Research. Toronto: IBM Press, 2003. 328–342.

[7] Chen Y, Yi K. Two-level sampling for join size estimation. In: Proc. of the 2017 ACM Int'l Conf. on Management of Data. Chicago: ACM, 2017. 759–774. [doi: 10.1145/3035918.3035921]

[8] Kim K, Jung J, Seo I, Han WS, Choi K, Chong J. Learned cardinality estimation: An in-depth study. In: Proc. of the 2022 Int'l Conf. on Management of Data. Philadelphia: ACM, 2022. 1214–1227. [doi: 10.1145/3514221.3526154]

[9] Kiefer M, Heimel M, Breß S, Markl V. Estimating join selectivities using bandwidth-optimized kernel density models. Proc. of the VLDB Endowment, 2017, 10(13): 2085–2096. [doi: 10.14778/3151106.3151112]

[10] Wang XY, Qu CB, Wu WY, Wang JN, Zhou QQ. Are we ready for learned cardinality estimation? Proc. of the VLDB Endowment, 2021, 14(9): 1640–1654. [doi: 10.14778/3461535.3461552]

[11] Stillger M, Lohman GM, Markl V, Kandil M. LEO-DB's LEarning optimizer. In: Proc. of the 27th Int'l Conf. on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 2001. 19–28.

[12] Hilprecht B, Schmidt A, Kulessa M, Molina A, Kersting K, Binnig C. DeepDB: Learn from data, not from queries! Proc. of the VLDB Endowment, 2020, 13(7): 992–1005. [doi: 10.14778/3384345.3384349]

[13] Wu ZN, Shaikhha A, Zhu R, Zeng K, Han Y, Zhou J. BayesCard: Revitilizing Bayesian frameworks for cardinality estimation. arXiv:2012.14743, 2021.

[14] Wang JY, Chai CL, Liu JB, Li GL. FACE: A normalizing flow based cardinality estimator. Proc. of the VLDB Endowment, 2021, 15(1): 72–84. [doi: 10.14778/3485450.3485458]

[15] Park Y, Zhong SC, Mozafari B. QuickSel: Quick selectivity learning with mixture models. In: Proc. of the 2020 ACM SIGMOD Int'l Conf. on Management of Data. Portland: ACM, 2020. 1017–1033. [doi: 10.1145/3318464.3389727]

[16] Negi P, Marcus R, Kipf A, Mao HZ, Tatbul N, Kraska T, Alizadeh M. Flow-loss: Learning cardinality estimates that matter. Proc. of the VLDB Endowment, 2021, 14(11): 2019–2032. [doi: 10.14778/3476249.3476259]

[17] Negi P, Wu ZN, Kipf A, Tatbul N, Marcus R, Madden S, Kraska T, Alizadeh M. Robust query driven cardinality estimation under changing workloads. Proc. of the VLDB Endowment, 2023, 16(6): 1520–1533. [doi: 10.14778/3583140.3583164]

[18] Wu PZ, Cong G. A unified deep model of learning from both data and queries for cardinality estimation. In: Proc. of the 2021 Int'l Conf. on Management of Data. ACM, 2021. 2009–2022. [doi: 10.1145/3448016.3452830]

[19] Li PF, Wei WQ, Zhu R, Ding BL, Zhou JR, Lu H. ALECE: An attention-based learned cardinality estimator for SPJ queries on dynamic workloads. Proc. of the VLDB Endowment, 2023, 17(2): 197–210. [doi: 10.14778/3626292.3626302]

[20] Zeng QX, Wang ZL, Zhang ZC, Xi PF, Wang N. A robust attention-based cardinality estimator with domain adaptation. Expert Systems with Applications, 2025, 276: 127065. [doi: 10.1016/j.eswa.2025.127065]

[21] Liu J, Dong WQ, Zhou QQ, Li D. Fauce: Fast and accurate deep ensembles with uncertainty for cardinality estimation. Proc. of the VLDB Endowment, 2021, 14(11): 1950–1963. [doi: 10.14778/3476249.3476254]

[22] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6405–6416.

[23] Woltmann L, Hartmann C, Thiele M, Habich D, Lehner W. Cardinality estimation with local deep learning models. In: Proc. of the 2nd Int'l Workshop on Exploiting Artificial Intelligence Techniques for Data Management. Amsterdam: ACM, 2019. 5. [doi: 10.1145/3329859.3329875]

[24] Zhang JT, Zhang C, Li GL, Chai CL. AutoCE: An accurate and efficient model advisor for learned cardinality estimation. In: Proc. of the 39th IEEE Int'l Conf. on Data Engineering. Anaheim: IEEE, 2023. 2621–2633. [doi: 10.1109/ICDE55515.2023.00201]

[25] Rashedi N, Moerkotte G. The accuracy of cardinality estimators: Unraveling the evaluation result conundrum. Proc. of the VLDB Endowment, 2025, 18(11): 3744–3756. [doi: 10.14778/3749646.3749651]

[26] Wu ZN, Negi P, Alizadeh M, Kraska T, Madden S. FactorJoin: A new cardinality estimation framework for join queries. Proc. of the ACM on Management of Data, 2023, 1(1): 41. [doi: 10.1145/3588721]

[27] Thirumuruganathan S, Shetiya S, Koudas N, Das G. Prediction intervals for learned cardinality estimation: An experimental evaluation. In: Proc. of the 38th IEEE Int'l Conf. on Data Engineering. Kuala Lumpur: IEEE, 2022. 3051–3064. [doi: 10.1109/ICDE53745.2022.00274]

[28] Qiu Y, Wang YL, Yi K, Li FF, Wu B, Zhan CQ. Weighted distinct sampling: Cardinality estimation for SPJ queries. In: Proc. of the 2021 Int'l Conf. on Management of Data. ACM, 2021. 1465–1477. [doi: 10.1145/3448016.3452821]

[29] Fuchs D, He Z, Lee BS. Compressed histograms with arbitrary bucket layouts for selectivity estimation. Information Sciences, 2007, 177(3): 680–702. [doi: 10.1016/j.ins.2006.07.013]

[30] Khachatryan A, Müller E, Stier C, Bohm K. Improving accuracy and robustness of self-tuning histograms by subspace clustering. IEEE Trans. on Knowledge and Data Engineering, 2015, 27(9): 2377–2389. [doi: 10.1109/TKDE.2015.2416725]

[31] Srivastava U, Haas PJ, Markl V, Kutsch M, Tran TM. ISOMER: Consistent histogram construction using query feedback. In: Proc. of the 22nd Int'l Conf. on Data Engineering (ICDE 2006). Atlanta: IEEE, 2006. 39. [doi: 10.1109/ICDE.2006.84]

[32] Wu CG, Jindal A, Amizadeh S, Patel H, Le WC, Qiao S, Rao S. Towards a learning optimizer for shared clouds. Proc. of the VLDB Endowment, 2018, 12(3): 210–222. [doi: 10.14778/3291264.3291267]

[33] Dutt A, Wang C, Nazi A, Kandula S, Narasayya V, Chaudhuri S. Selectivity estimation for range predicates using lightweight models. Proc. of the VLDB Endowment, 2019, 12(9): 1044–1057. [doi: 10.14778/3329772.3329780]

[34] Wu ZN, Yu P, Yang PL, Zhu R, Han YX, Li YL, Lian DF, Zeng K, Zhou JR. A unified transferable model for ML-enhanced DBMS. In: Proc. of the 12th Biennial Conf. on Innovative Data Systems Research. Chaminade: CIDRDB, 2022.

[35] Gregor K, Danihelka I, Mnih A, Blundell C, Wierstra D. Deep autoregressive networks. In: Proc. of the 31st Int'l Conf. on Machine Learning. Beijing: JMLR.org, 2014. 1242–1250.

[36] Zhao KF, Yu JX, He ZY, Li R, Zhang H. Lightweight and accurate cardinality estimation by neural network Gaussian process. In: Proc. of the 2022 Int'l Conf. on Management of Data. Philadelphia: ACM, 2022. 973–987. [doi: 10.1145/3514221.3526156]

[37] Lee J, Bahri Y, Novak R, Schoenholz SS, Pennington J, Sohl-Dickstein J. Deep neural networks as Gaussian processes. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[38] Selinger PG, Astrahan MM, Chamberlin DD, Lorie RA, Price TG. Access path selection in a relational database management system. In: Proc. of the 1979 ACM SIGMOD Int'l Conf. on Management of Data. Boston: ACM, 1979. 23–34. [doi: 10.1145/582095.582099]

[39] Deshpande A, Garofalakis M, Rastogi R. Independence is good: Dependency-based histogram synopses for high-dimensional data. ACM SIGMOD Record, 2001, 30(2): 199–210. [doi: 10.1145/376284.375685]

[40] Gunopulos D, Kollios G, Tsotras VJ, Domeniconi C. Approximating multi-dimensional aggregate range queries over real attributes. In: Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM, 2000. 463–474. [doi: 10.1145/342009.335448]

[41] Kipf A, Kipf T, Radke B, Leis V, Boncz P, Kemper A. Learned cardinalities: Estimating correlated joins with deep learning. In: Proc. of the 9th Biennial Conf. on Innovative Data Systems Research. Asilomar: CIDRDB, 2019.

[42] Li FF, Wu B, Yi K, Zhao ZY. Wander join: Online aggregation via random walks. In: Proc. of the 2016 Int'l Conf. on Management of Data. San Francisco: ACM, 2016. 615–629. [doi: 10.1145/2882903.2915235]

[43] Chow C, Liu C. Approximating discrete probability distributions with dependence trees. IEEE Trans. on Information Theory, 1968, 14(3): 462–467. [doi: 10.1109/TIT.1968.1054142]

[44] Getoor L, Taskar B, Koller D. Selectivity estimation using probabilistic models. In: Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data. Santa Barbara: ACM, 2001. 461–472. [doi: 10.1145/375663.375727]

[45] Tzoumas K, Deshpande A, Jensen CS. Lightweight graphical models for selectivity estimation without independence assumptions. Proc. of the VLDB Endowment, 2011, 4(11): 852–863. [doi: 10.14778/3402707.3402724]

[46] Yang ZH, Liang E, Kamsetty A, Wu CG, Duan Y, Chen X, Abbeel P, Hellerstein JM, Krishnan S, Stoica I. Deep unsupervised cardinality estimation. Proc. of the VLDB Endowment, 2019, 13(3): 279–292. [doi: 10.14778/3368289.3368294]

[47] Yang ZH, Kamsetty A, Luan SF, Liang E, Duan Y, Chen X, Stoica I. NeuroCard: One cardinality estimator for all tables. Proc. of the VLDB Endowment, 2020, 14(1): 61–73. [doi: 10.14778/3421424.3421432]

[48] Zhu R, Wu ZN, Han YX, Zeng K, Pfadler A, Qian ZP, Zhou JR, Cui B. FLAT: Fast, lightweight and accurate method for cardinality estimation. Proc. of the VLDB Endowment, 2021, 14(9): 1489–1502. [doi: 10.14778/3461535.3461539]

[49] Zeng TJ, Lan JW, Ma JH, Wei WQ, Zhu R, Zhou YL, Li PF, Ding BL, Lian DF, Wei ZW, Zhou JR. PRICE: A pretrained model for cross-database cardinality estimation. Proc. of the VLDB Endowment, 2024, 18(3): 637–650. [doi: 10.14778/3712221.3712231]

[50] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

[51] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. on Neural Networks, 1994, 5(2): 157–166. [doi: 10.1109/72.279181]

[52] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

[53] Sun YT, Dong L, Huang SH, Ma SM, Xia YQ, Xue JL, Wang JY, Wei F. Retentive network: A successor to Transformer for large language models. In: Proc. of the 2024 Int'l Conf. on Learning Representations. Vienna: OpenReview.net, 2024.

[54] Behrouz A, Zhong PL, Mirrokni V. Titans: Learning to memorize at test time. In: Proc. of the 39th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2025.

[55] Yang SL, Wang BL, Zhang Y, Shen YK, Kim Y. Parallelizing linear Transformers with the delta rule over sequence length. In: Proc. of the 38th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2024. 3668.

[56] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces. In: Proc. of the 1st Conf. on Language Modeling. Philadelphia: OpenReview.net, 2024.

[57] Dao T, Gu A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In: Proc. of the 41st Int'l Conf. on Machine Learning. Vienna: JMLR.org, 2024. 399.

[58] Aytimur M, Chondrogiannis T, Grossniklaus M. SPACE: Cardinality estimation for path queries using cardinality-aware sequence-based learning. Proc. of the ACM on Management of Data, 2025, 3(3): 218. [doi: 10.1145/3725355]

[59] Moerkotte G, Neumann T, Steidl G. Preventing bad plans by bounding the impact of cardinality estimation errors. Proc. of the VLDB Endowment, 2009, 2(1): 982–993. [doi: 10.14778/1687627.1687738]

[60] Yang SL, Wang BL, Shen YK, Panda R, Kim Y. Gated linear attention Transformers with hardware-efficient training. In: Proc. of the 41st Int'l Conf. on Machine Learning. Vienna: PMLR, 2024. 56501–56523.

[61] FIT ČVUT. Projekt relational. 2026. https://relational.fit.cvut.cz/dataset/Stats

[62] Leis V, Gubichev A, Mirchev A, Boncz P A, Kemper A, Neumann T. How good are query optimizers, really? Proc. of the VLDB Endowment, 2015, 9(3): 204–215.

[63] TPC. TPC-H version 2 and version 3. 2026. https://www.tpc.org/tpch/default5.asp

[64] PostgreSQL Global Development Group. PostgreSQL: The world's most advanced open source relational database. https://www.postgresql.org

附中文参考文献

[1] 岳文静, 屈稳稳, 林宽, 王晓玲. 基于机器学习的基数估计技术综述. 计算机研究与发展, 2024, 61(2): 413–427. [doi: 10.7544/issn1000-1239.202220649]

作者简介

余晓婷, 硕士生, CCF 学生会员, 主要研究领域为数据库查询优化, 基数估计.

高锦涛, 博士, 副教授, CCF 高级会员, 主要研究领域为数据库查询优化, AI4DB, 推荐系统, LLM.