ThalamusDB：多模态数据的近似查询处理

SAEHAN JO，美国康奈尔大学
美国康奈尔大学 IMMANUEL TRUMMER

我们介绍的 ThalamusDB 是一种新型近似查询处理系统，可处理多模态数据上的复杂 SQL 查询。ThalamusDB 支持在视觉、音频和文本数据上整合自然语言谓词的 SQL 查询。为回答此类查询，ThalamusDB 结合关系处理利用了一系列零点模型。ThalamusDB 采用确定性近似查询处理，利用关系处理的相对效率来减轻机器学习推理的计算需求。在评估自然语言谓词时，ThalamusDB会要求用户提供少量标签。用户可以指定自己对三个相关指标性能目标的偏好：近似误差、计算时间和标签开销。ThalamusDB 查询优化器会根据用户的偏好选择优化方案，优先处理数据和请求的标签，以最大限度地提高效果。使用 Craigslist、YouTube 和 Netflix 等多个真实数据集进行的实验表明，ThalamusDB 比精确处理基线 MindsDB 平均提速 35.0 倍，在 78.9% 的情况下优于基于采样的 ABAE 方法。

中央案例研究概念：- **信息系统 → 在线分析处理引擎**；**数据模型扩展**；
• **计算方法** → *机器学习*。

其他关键词和短语：查询处理、多模态数据、神经模型

ACM 参考格式：

Saehan Jo 和 Immanuel Trummer。2024.ThalamusDB：多模态数据的近似查询处理。
Proc.ACM Manag.Data 2, 3 (SIGMOD), Article 186 (June 2024), 26 pages. https://doi.org/10.1145/3654989

1 引言

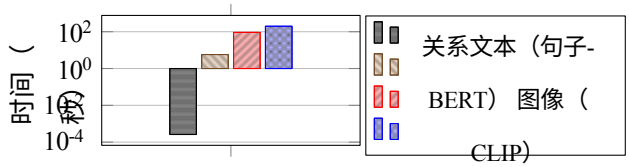
之前的工作已经引入了将机器学习与数据库相结合的系统，如 MindsDB [30] 和 EvaDB [17]。这些系统利用大型神经模型，使用户能够在统一的框架内处理多模态数据。在这种情况下，处理多模态数据查询的主要挑战是与模型推理相关的计算成本，如图 1 所示。处理大量多模态数据的成本非常高昂，成为查询处理的主要因素。这种处理开销凸显了对更高效解决方案的需求。因此，我们引入了 ThalamusDB，一个专为多模态数据复杂查询而设计的近似查询处理（AQP）系统。ThalamusDB 的代码见 https://github.com/saehanjo/thalamusdb。

ThalamusDB 支持将自然语言谓词整合到多模态数据中的检索和聚合查询（带连接）。它支持扩展的关系数据模型，引入了专门的列类型以整合图片和音频数据。查询使用支持

谓词的 SQL 方言，以自然语言对文本、图片或音频数据进行表述

作者联系方式：Saehan Jo，美国纽约伊萨卡康奈尔大学，sj683@cornell.edu；Immanuel Trummer，美国纽约伊萨卡康奈尔大学，itrummer@cornell.edu。

ACM 2836-6573/2024/6-ART186
https://doi.org/10.1145/3654989



竞赛 [29]

图 1.关系数据上的相等谓词与非结构化数据上的自然语言谓词在 10,000 个条目上进行谓词评估的运行时间比较。

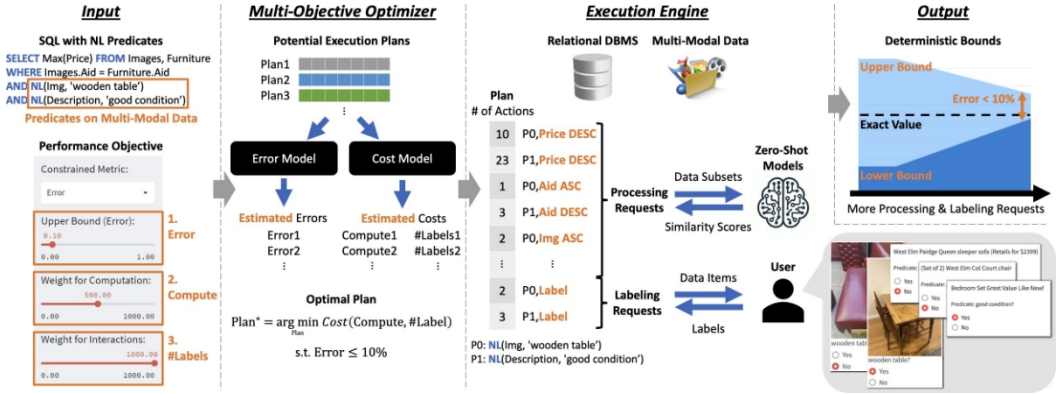


图 2.ThalamusDB 概述。

列。在谓词评估方面，ThalamusDB 采用了零镜头分类器，如 OpenAI CLIP [36] 和 Sentence-BERT [37]，该分类器可仅根据相关类别的自然语言描述对各种数据类型进行分类。

ThalamusDB 采用确定性 AQP 技术处理多模态数据。ThalamusDB 的近似方法是将模型应用于数据子集，并要求用户标注少量项目。ThalamusDB 不是随机抽样数据，而是精心选择要处理和标注的数据子集。ThalamusDB 依靠特定场景优化器来确定最佳数据子集，以获得最佳结果质量。用户在多模态数据上提交带有自然语言谓词的 SQL 查询。用户在三个相关指标（结果近似误差、标记开销和计算开销）之间进行权衡。查询处理开始后，用户会被要求标注少量多模态数据项，确定它们是否满足谓词要求。ThalamusDB 利用提供的标签和处理过的数据子集，为查询集合生成边界，并保证包含精确值。

基于采样的近似是以结果精度换取处理开销的最经典方法 [12]。然而，在我们的问题设置中，抽样有几个局限性。由于自然语言谓词的无限制性，通过分层抽样等方法生成非官方数据样本具有挑战性。此外，在确定相似性阈值时，采样并不能很好地减少所需的标签数量。相反，更有效的方法是系统性地选择具有最有助于完善阈值范围的相似性得分的项目。同时，抽样也会给最大值和最小值等集合带来缺陷[34]。相比之下，ThalamusDB 依靠确定性 AQP 进行逼近。所谓 "确定性"，是指结果边界始终保证包含真实值，而不

是置信度边界。之前关于确定性 AQP 的研究 [15, 34] 表明，处理数据子集可以产生确定性的查询结果界限。ThalamusDB 利用了这一观察结果，即处理不同的数据子集可能会对结果产生不同程度的改进。

精确ThalamusDB 的独特之处在于它专注于多模态数据。因此，ThalamusDB 利用相对便宜的关系处理（与机器学习推理相比）来仔细选择行子集，从而最大限度地减少近似误差。

ThalamusDB 依靠多目标优化器来探索计划空间中的备选处理方案（即选择处理哪些数据子集并为自然语言谓词贴标签）。因此，优化器配备了针对特定场景的误差和成本模型，可估算中间结果的大小、处理开销、所需标签数量以及执行一系列处理步骤后的近似误差。与不同自然语言谓词相关的分类器在每次调用成本上可能会有很大差异（例如，由于数据类型和模型大小的不同，如图 1 所示）。同时，谓词的选择性也各不相同，因此某些评估顺序比其他顺序更优。最后，确定谓词评估和项标签的优先级（两者都有助于提高近似精度）需要考虑数据和查询属性。针对特定场景的查询优化器能够对上述所有调整决策做出适当的选择。给定一个最优计划后，ThalamusDB 会在精心挑选的数据子集上应用分类器，处理输入查询的剩余部分（不涉及自然语言谓词），请求用户手动标记，最后将最终查询结果呈现给用户。

关键是要区分 ThalamusDB 中出现的不同类型的近似误差以及相关保证。首先，ThalamusDB 使用神经模型来处理多模态数据。这种模型本身并不完美，可能会犯错误。然而，提高中性模型的准确性是本研究的一个正交问题，不在我们的研究范围之内。相反，ThalamusDB 专注于最大限度地减少处理数据子集而非全部数据集所造成的错误。假设神经模型没有错误（当然，这一假设是简化的），ThalamusDB 将产生确定性边界，保证包含通过处理完整数据集获得的值。最后，我们需要区分*执行中的近似误差*和*优化中的近似误差*。ThalamusDB 在处理特定数据子集进行优化之前，会使用模型来估计误差（和成本）。与查询优化中的通常做法一样，这些模型*无法保证*确定性边界或最优处理计划的选择。不过，由于优化器误差模型*仅*用于*选择数据子集*，而不是生成查询结果，因此优化器模型缺乏保证并不影响对查询结果的保证。

在实验中，我们评估了 ThalamusDB 在 Craigslist、YouTube 和 Netflix 三个真实世界数据集上的各类查询。这些基准包括对不同数据类型的查询，包括图片、音频和文本，以及对不同数据类型组合的查询。ThalamusDB 的性能取决于数据分布，原则上可能出现病态情况（参见第 4.3 节中的示例 3）。不过，我们的实验证明，它在现实世界的数据集上表现良好。我们将 ThalamusDB 与多个基线、MindsDB 和 ABAE 进行了比较，结果显示 ThalamusDB 有了显著的改进。总之，本文的原创性科学贡献如下：

- 我们介绍的 ThalamusDB 是一个确定性近似查询处理系统，用于回答多模态数据中带有

自然语言谓词的复杂查询。

- 我们概述了一些处理方法，这些方法可以减少在大型数据集上运行涉及机器学习推理的查询时的处理开销。
- 我们引入了一个多目标优化器，使用户能够在不同的成本和误差指标之间进行权衡。
- 我们在多个真实世界数据集上对 ThalamusDB 进行了实验评估，并与 MindsDB 和 ABAE 等各种基线进行了比较。

Proc.ACM Manag.数据》，第 2 卷，第 3 期（SIGMOD），第 186 条。出版日期：2024 年 6 月。

```
CREATE TABLE Ads（AdID int PRIMARY KEY、
  创建日期，AdText 文本， Price int）；
创建表 Pics（文件路径 PICTURE PRIMARY KEY、
  AdID int, FOREIGN KEY (AdID) REFERENCES Ads(AdID)）；
```

图 3.运行示例数据库的模式。

```
SELECT Min(A.Price)
FROM 广告 A, 图片 P
WHERE Created > DATE'2022-01-01'
  AND P.AdID = A.AdID
  AND NL(P.FilePath、
    带蓝色坐垫的木椅"）。
```

图 4.查询示例数据库，计算 2022 年 1 月 1 日之前提供椅子的广告的最低价格。

2 系统概述

图 2 显示了 ThalamusDB 系统的概览。ThalamusDB 使用数据集上的自然语言谓词处理 SQL 查询，整合了结构化数据、图片、音频数据和文本。数据库模式是使用稍加扩展的 SQL DDL 命令定义的，并整合了专用数据类型以引用图片和音频文件。

ThalamusDB 依靠两种外部组件进行数据处理：关系数据库管理系统（DBMS）和用于多模态数据分析的神经模型库。关系数据库管理系统用于管理数据库模式和结构化数据处理。在内部，ThalamusDB 将图片和音频数据列表示为 DBMS 中的文本列，其中包含存储非关系型数据目录中文件的路径。神经模型用于评估输入查询中出现的自然语言谓词。目前，ThalamusDB 使用 CLIP [36] 处理图像，使用 Mei 等人的模型 [29] 和 CLAP [9] 处理音频数据，使用 Sentence-BERT [37] 和 BART [26] 处理文本上的自然语言谓词。每种数据类型的模型都可以轻松交换，不费吹灰之力。在内部，ThalamusDB 将传入的 DDL 指令和查询分解为引用多模态数据的部分（单独处理）和标准 SQL（由关系 DBMS 处理）。下面的示例说明了用户如何使用系统对多模态数据进行查询。

例 1.*爱丽丝正在为自己的公寓添置家具，她正在 Craigslist 上寻找一把带蓝色坐垫的木椅。为了了解价格范围，她想找到最近几个月的最低价格。然而，许多房源的文字描述中缺少颜色或材质细节，这就要求她分析相关图片。由于没有时间浏览大量图片，爱丽丝决定试用多模态数据分析新系统 ThalamusDB。*

Alice 从 Craigslist 搜刮数据，并以关系格式插入 ThalamusDB，其模式如图 3 所示。该模

式使用了非标准的 "PICTURE "数据类型，表明图像分析模型可用于相关文件。现在，爱丽丝可以使用图 4 中的查询来提出请求。该查询使用了 "NL "关键字，以便在非结构化（即图片、音频或文本）列上使用自然语言谓词。ThalamusDB 使用大型神经网络评估这些谓词，计算文本和数据之间的相似度得分。一旦相似度达到一定的阈值，谓词就会被认为满足要求。最佳阈值因谓词和数据集而异，因此 ThalamusDB 要求用户标注有限数量的样本进行校准。例如，图 2 显示了图片和文本的标注请求。此外，用户还可以对系统进行配置，以便在近似误差、标注开销和计算开销这三个指标之间做出不同的权衡。性能目标的用户界面（如图 2 所示）允许

指定一个度量的约束条件和其余两个[1] *的权重。为了节省时间，Alice 将标签请求的数量限制为 5 个。ThalamusDB 会选择一个最佳处理计划，分析图片子集，请求最多五张图片的标签，最后生成近似查询结果。*

对于每个输入查询，ThalamusDB 都会执行以下阶段。

初始化ThalamusDB 通过创建临时表为查询处理准备关系数据库，查询中出现的每个自然语言谓词都有一个临时表。每个表都包含两列，分别代表项目 ID 和相似度得分，用于捕捉谓词文本与项目（即图片、音频文件或文本）之间的相似度。初始化后，相似度得分是未知的，由 SQL NULL 值表示。

预处理接下来，ThalamusDB 会执行一个预处理阶段，针对每个谓词，它都会使用多模态模型计算谓词文本与项目之间的相似度值，并计算少量项目样本。同时，对于每个谓词，它都会要求用户为少量项目（三个）提供标签。这样，ThalamusDB 就能缩小相似性阈值，从这个阈值开始，谓词就被认为是满足要求的。预处理的目的是收集查询的特定统计数据，包括自然语言谓词选择性（无法根据直方图等进行估算），这些数据有助于优化查询。

查询优化。然后，ThalamusDB 会调用特定场景的查询优化器，生成优化的执行计划。用户可以对计算成本、请求的标签数量和近似误差这三个成本指标中的任何一个指定约束（上限）或权重。优化器会将这些性能目标考虑在内。优化后的执行计划将最大限度地降低遵守约束条件的加权成本。

查询执行。执行计划会转发给执行引擎，由其执行一系列操作。ThalamusDB 支持两种类型的执行操作：优先数据处理和标签请求。首先，优先数据处理包括应用多模态模型计算特定谓词和项目子集的相似性得分。这些项目不是随机选择的，而是根据同一张表中其他列的值确定优先级的（例如，参照例 1，我们可以按照价格升序选择 Craigslist 项目的图片）。由此得出的相似度得分存储在初始化阶段创建的临时表中。其次，标签请求要求用户确定特定自然语言谓词是否适用于特定物品（例如，要求爱丽丝验证特定图片是否显示了一把木椅）。ThalamusDB 不会在大型模型上进行（昂贵的）训练。相反，它使用标签来缩小相似性得分的阈值，并从得分较高的项目开始考虑满意度。在使用多模态模型计算出谓词和项目子集的相似性值，并通过用户标签缩小相似性值的阈值后，引擎会执行 SQL 查询，分别计算出查询结果聚合的下限和上限。

后处理和提前终止。优化器根据基于预处理的统计数据选择执行计划，使用（查询优化

领域的典型做法）简化假设来估计执行开销和近似误差，即查询集合的下限和上限之间的距离。因此，实际执行开销和由此产生的界限之间的距离可能会偏离估计值。ThalamusDB使用额外的机制来确保用户指定的约束条件得到遵守。更确切地说，只要聚合计算开销超过用户约束，执行引擎就会中断执行。如果近似误差不满足用户指定的约束条件，在计划执行后、

[1]允许用户对多个指标指定约束条件，可能会导致系统无法满足所有约束条件。如果只在一个指标上设置约束，就不会出现这种情况。例如，在处理足够多的数据和请求足够多的标签时，任何近似误差约束都可以满足。

ThalamusDB 会进行后处理，以进一步减少误差。在后处理过程中，ThalamusDB 会贪婪地选择能最大程度减少预期误差的操作。每个操作要么向用户请求特定谓词和项目的标签，要么计算特定谓词和项目子集的相似性得分。一旦边界之间的距离变得足够小，从而满足用户指定的误差约束条件，后处理就会停止。

3 正式模式

ThalamusDB 支持以下类型的查询和模式。

定义 1（多模式模式）。ThalamusDB 支持使用标准 SQL DDL（数据定义语言）命令和多模式数据扩展的星形模式：ThalamusDB 支持两种额外的数据类型：图片（PICTURE）和音频（AUDIO）。相应列包含图像或音频文件的路径。

定义 2（多模式查询）。ThalamusDB 支持多模式模式的 SPJA（选择-项目-连接-聚合）查询。与标准 SQL 相比，ThalamusDB 支持一个额外的关键字：NL。NL 可以在 WHERE 子句中使用，并在图像、音频或文本列上引入自然语言谓词。该函数需要两个参数，即应用谓词的列和描述谓词条件的字符串。

目前的实现仅限于在聚合中引用非负列的 SQL 查询，并假定在星形模式数据库中进行连接。ThalamusDB 借助接下来定义的模型处理查询中的自然语言谓词。

定义 3（零镜头模型）。给定一个谓词文本和一个数据项（图片、音频文件或文本），零点模型计算文本-数据相似度得分。如果相似度高于相似度阈值（下文将讨论），则认为相关谓词得到满足。

模型可能会产生不一致的相似性得分，从而导致满足谓词的数据项与其他数据项之间的不完全分离。不过，提高分类器的准确性是一个正交问题，不在本文讨论范围之内。ThalamusDB 的目标是尽可能高效地利用现有模型。对于每个谓词，都会通过标签请求来确定一个合适的相似性阈值（类似于最近其他分类工作中使用的阈值[13]）。

定义 4（标签请求）。标签请求由一对$\langle d, t, s\rangle$描述，其中d 是图片、音频文件或文本，t 是描述谓词的文本，s 是文本与谓词之间的相似度得分。$\langle d$、$t\rangle$ 会呈现给用户（使用合适的表示方法，如音频文件的播放控制台），要求用户输入d是否满足t。

ThalamusDB 能让用户以标签或处理开销换取查询结果的精确度。它产生的近似结果定义如下。

定义 5（近似结果）。ThalamusDB 计算了行子集的相似性得分，并通过标记请求缩小了相似性阈值，然后计算出查询集合的（确定性）下限和上限。给定上下限u 和l，根据 Krenovich [24] 的定义，近似误差为 $(u - l)/(u + l) \in [0, 1]$。如果查询包含多个聚合，ThalamusDB 会使用所有聚合的算术平均误差。对于不含聚合的查询，ThalamusDB 使用 $(u - l)/(u + l)$，其中u 和 l 是准确查询结果中行数的上下限。

ThalamusDB 使用户能够在所有相关成本和质量指标之间表达偏好。

输入 △带有 NL 谓词的查询；e：执行计划；R：关系数据库管理系统；M：多模态处理器
输出 近似结果（即确定性边界）

```
1: 函数 EXECUTE(△, e, R, M)
       // 根据数据处理的优先顺序计算相似性得分。
2: 对于所有⟨z, s, k⟩ ∈ e.C do
3:         ids ← R.Run("SELECT {z.col }"){z.col }}JOIN S{z.id }
                   使用 {z.col }WHERE Score IS NULL ORDER BY
                   {s }LIMIT {k }")
               // 将分数更新到相似性表中。
4:         对于所有 id∈ids do
5:             v ← M.COMPUTESIM(id, z)
6:             R.Run("UPDATE S{z.id } SET Score={v} WHERE id={id}")
       // 根据标签请求更新分数阈值。
7: 对于所有 ⟨z, k⟩ ∈ e.L do
8:         z.θ_l , z.θ_u ← REQESTLABELS(z, k)
       // 重写查询以使用相似性表。
9:     Q ← REWRITE(△)
       // 执行重写查询以获得确定性边界。
10:    return {R.Run(△) | △∈ Q}
```

算法 1.ThalamusDB 的执行过程。

定义 6（用户偏好）。ThalamusDB 将近似误差、计算量过大和标记请求数量作为成本指标。用户可以限制其中任何一个指标（通过设置上限），并选择两个非限制指标的权重。ThalamusDB 在查询处理过程中尊重约束条件，同时以最小化加权成本为目标。

ThalamusDB 解决了以下查询问题。

定义 7（多模式交互式查询）。ThalamusDB 会处理以下形式的请求
⟨△，o⟩，其中△是一个查询，包含多模态模式上的自然语言谓词，o是用户对性能目标的偏好说明。ThalamusDB 计算数据子集的相似性得分，并返回一组 L 标签请求。用户提交答案 $A = \{\langle r, b\rangle | r \in L, b \in \{Yes, No\}\}$，为每个标签请求分配一个布尔答案。最后，系统返回近似查询结果。

4 执行引擎

算法 1 在 ThalamusDB 中执行给定的查询计划。该算法可分解为三个部分。首先，在第 1 行至第 6 行，算法应用多模态模型计算量化谓词文本和数据项之间相似性的分数。该算法侧重于谓词和数据项的子集，作为输入计划（"优先数据处理"）的一部分进行指定。其次，在第 7 至第 8 行，算法 1 要求用户标注精心挑选的谓词-条目组合子集。这有助于确定特定谓词的相似性分数阈值，并从阈值开始考虑满足哪些谓词。第三，在第 9 至第 10 行，算法 1 使用前面步骤中收集的所有信息来计算查询总值的下限和上限。为此，算法会发出从

原始查询衍生而来的重写查询。

下面三个小节将分别介绍算法 1 三个部分中的一个。表 1 描述了查询计划属性，用于算法 1 和后续算法。同样，算法 1 和下面的算法也提到了查询谓词的属性。表 2 总结了相关字段。

表 1.带有 NL 谓词及其属性的查询 △。

属性语义	
q.Z	括号中的自然语言谓词
蒇蒇蒇蒇蒇蒇蒇蒇蒇蒇蒇蒇蒇	伕伕*ag*
f.rom	在△中的 FROM 子句
翰蠢翰豃巍蟿危 翰蠢翰豃巍蟿危 翰蠢翰豃巍蟿危	括号中的 WHERE 子句
打击*.ilmit*	括号中 LIMIT 子句的记录数

表 2.NL 谓词*z* 及其属性。

属性语义	
z.id	NL 谓词中的唯一 ID
z.text	描述谓词的文本
z.type	非结构化数据类型，如图片、音频、文本
z.col	应用*z* 的列的名称
z.table	包含列*z.col*的表名
$z.\theta_l$	当前分数阈值的下限
$z.\theta_u$	当前分数阈值的上限

4.1 优先数据处理

首先，引擎会执行涉及评估多模态数据分类器的处理步骤（算法 1 中的第 2 行至第 6 行）。计算步骤以三元组$\langle z, s, k\rangle$描述，其中z是谓词，k是要评估的行和项的数量，s是行的优先级顺序。对于每个这样的三元组，ThalamusDB 会首先收集顺序为 s 的前 k 个项的 ID，对于这些项，谓词z 的相似性得分尚未可知（第 3 行）。ThalamusDB 通过查询用于处理的底层关系数据库来收集 ID（我们使用大括号在 SQL 查询字符串中包含变量，类似于 Python 语法）。接下来，ThalamusDB 会遍历选中的项目，应用合适的模型（取决于数据模式）计算相似度得分（第 5 行），并将得到的得分存储到与谓词相关联的临时表中（第 6 行）。正如下面的示例所示，处理顺序的选择会对产生的近似误差产生重大影响（我们将在第 5 节讨论如何选择处理顺序）。

示例 2.*运行中的示例查询（如图 4 所示）要求得到满足指定谓词的报价的最小价格。当 ThalamusDB 评估自然语言谓词以计算这个集合时（假设已经处理了日期不等式谓词），不同的行处理方法会产生不同的结果。如果我们以随机顺序处理行，那么即使我们处理了更多项目（除非我们处理了所有项目），最小价格仍然是未知的。这是因为价格较低的行可能还未被发现。相反，如果我们根据价格值以升序处理数据行，我们就能确保满足谓词的第一行确实是所有符合条件的数据行中的最小价格，从而无需处理后续数据行。类似的推理也可应用于最大值以及带连接的求和与计数，尤其是在偏斜分布的情况下。*

参数 k 指定了单个计算步骤中需要处理的项目数量。采用较小的 k 值可以实现更精细的

决策，但也会导致优化开销增加。因此，选择一个能平衡以下开销的k 值非常重要

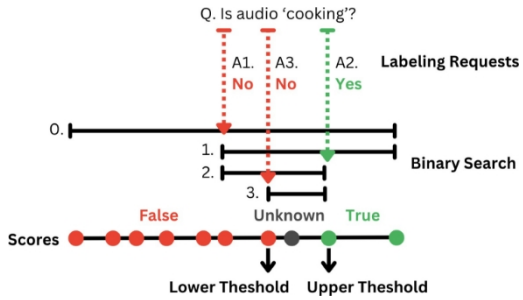


图 5.标签请求产生的下限和上限分数阈值。

与计划质量有关。ThalamusDB 为成本较高的模型配置了较小的k（在我们的实验中，音频使用 0.1% 的数据项，图像和文本使用 1%）。

4.2 标签要求

其次，在算法 1 的第 7-8 行中，ThalamusDB 向用户请求标签。标签请求以(z, k)的形式描述。这里，z 表示零次谓词，k 表示要检索的标签数。标签制作包括向用户展示一个相应的数据项（通过展示图片、文本或播放音频文件），同时要求用户判断谓词z 是否适用于该数据项。如图 5 所示，第 8 行中的 REQESTLABELS 执行的算法类似于二进制搜索。标记的目的是利用给定的标签数尽可能缩小相似性阈值的范围。因此，REQESTLABELS 会从相似性得分尽可能接近相似性中值的项目开始。根据标签（正标签或负标签）的不同，它要么继续选择相似度接近相似度中值上半部分（如果是正值）的项目，要么继续选择相似度中值下半部分的项目。为了采用更稳健的方法，我们可以呈现多个相似度得分连续的项目，而不是只有一个。这种方法降低了一个项目不能充分反映其当前得分位置的概率，并有助于找到准确的得分阈值。标注的结果是更新的相似性阈值下限和上限，即$z.\theta_l$ 和$z.\theta_u$ ，从这两个阈值开始，谓词被认为是满足的。最后，ThalamusDB 使用谓词和数据项子集的相似性值以及标记结果来计算近似查询结果。近似查询结果使用关系数据库管理系统计算。模型评估的结果--相似度得分已经存储在数据库内部的临时表中。相似度阈值由标签作为常量集成到 SQL 查询中。

4.3 查询重写

最后，在算法 1 的第 9 和第 10 行，ThalamusDB 会计算查询结果的上下限，并将其作为输出返回给用户。为了计算上下限，ThalamusDB 会对包含前几个执行阶段结果的数据库进行 SQL 查询。这些查询利用的是存储在数据库中的相似性分数（这些分数只存储在通过优先数据处理选出的谓词和项的子集中，而所有其他谓词-项组合的分数都被设置为 SQL 空值）。此外，它还利用了从用户标签中推断出的相似性阈值（$z.\theta_l$ 和 $z.\theta_u$ 代表谓词z的相应相似性阈值的下限和上限）。

算法 2 描述了如何重写原始查询，以获得一组生成近似结果的查询。这个结果由查询集合的下限和上限组成（或者，在没有集合的查询中，由一组结果行组成，其中一些行被标记为暂定结果，因为

输入△：带 NL 谓词的查询
输出（在相似性表上）重写查询，为每个集合计算确定性边界。

```
1: 函数 REWRITE(△)
       // 根据 FROM 子句的 NL 谓词连接相似性表。
2:     f ←△.from
3:     for all z ∈ △.Z do
4:         f ← f +" JOIN S{z.id }.ON S{z.id }.id = {z.col }"
       // 为下限和上限初始化 WHERE 子句。
5:     l w ← 默认.were
6:     u w ← 默认.were
       // 用相似性得分谓词替换 NL 谓词侧面。
7:     for all z ∈ △.Z do
           // 关于 "肯定为真"（-）和 "可能为真"（-），请参见表 3。
8:         l .REPLACE(¬z, DEFITRUE(¬z)) .REPLACE(z, DEFITRUE(z))
9:         u w .REPLACE(¬z, POSSTRUE(¬z)) .REPLACE(z, POSSTRUE(z))
       // 没有聚合的查询情况。
10:    若 △.ags = ∅ ：
11:        s贬l ← "SELECT * {f }{wl }LIMIT {2012.limit }"
12:        s贬u ← "SELECT * {f }{wu }LIMIT {2012.limit }"
13:        return {s checkedl , s checkedu}
       // 为每个集合体添加两个下限和上限查询。
14:    Q ← ∅
15:    for all a(col) ∈ △.a ggs do
16:        如果 a = Avg：
               // 对于平均值，用总和除以计数来代替。
17:            s贬l ← "SELECT (SELECT Sum(col) {f }){wl })
                                /(SELECT Count(col) {f }" {w  }){wu })"
18:            s贬u ← "SELECT (SELECT Sum(col) {f }){wu })
                                /(SELECT Count(col) {f }" {w  }){wl })"
19:        else if a = Min then
               // 为最小值，wl 和 wu 对调。
20:            s默认l ← "SELECT a(col) {f }" {w }{wu }"
21:            s默认u ← "SELECT a(col) {f }" {w }{wl }"
22:        其他
23:            l ← "SELECT a(col) {f }" {w }{wl }"
24:            u ← "SELECT a(col) {f }" {w }{wu }"
25:        Q ← Q∪ {sql l , sql }u
26:    返回 Q
```

算法 2.ThalamusDB 的查询重写。

不清楚它们是否满足所有相关谓词）。首先，重写程序通过添加所有相关的相似性表，存储自然语言谓词的文本-项目相似性值，以及适当的连接条件（第 2 至 4 行）来增强原始 FROM 子句。接下来，将生成 SQL WHERE 子句的两个不同变体（第 5 至 9 行）。第一个变体，w_l，选择可能满足也可能不满足所有自然语言谓词的记录。第二个w_u 选择肯定满足所有自然语言谓词的记录。由于计算和标注能力有限，我们无法验证所有行是否满足所有谓词（例如图 5 中的 "未知"，其分数阈值太宽，无法确定其状态）。两种不同的 WHERE 子句用于计算结果下限和上限，以表达这种不确定性。为了获得这两种变体，自然语言谓词的出现按照表 3 进行了替换。否定谓词

表 3.NL 谓词出现的重写规则。

原始 SQL	绝对真实(-)	可能为真(-)
NL(col, str)	得分≥ θ_u	得分> θ_l OR score IS NULL
NOT NL（col，str）	得分≤ θ_l	score< θ_u OR score IS NULL

```
SELECT Min(A.Price) FROM Ads A
  JOIN Pics P USING (AdID) JOIN
  S0 USING ( FilePath)
WHERE 创建日期 > DATE'2022-01-01
  AND (S0.Score > θl  OR S0.Score IS NULL)
```

图 6.计算 2022 年 1 月 1 日之前提供椅子的广告最低价格下限的重写查询。

会被单独处理，并先于其他值被替换。请注意表 3 中提到的 NULL 值。如果相似性表中的某一行出现了 NULL 值，则相应的谓词尚未对相关数据项进行评估。

无聚合的查询是一种特殊情况，在第 10 行至第 13 行中处理。这里的近似结果是两个行集，一个肯定满足所有谓词，另一个可能满足所有谓词。这些结果集可以根据上一步得到的两个 WHERE 子句变体直接生成。

最后，在第 14 行至第 29 行中，算法 2 对选择子句中的不同聚合进行迭代。对于每个聚合，算法会生成两个查询（收集在查询集 Q），分别计算该谓词的下界和上界。例如，对于最大值、总和和计数聚合，上界是通过考虑所有可能满足谓词的记录得到的(假设聚合列中的值为非负值）。如果只考虑肯定满足所有谓词的行，就会得出汇总结果值的下限。对于最小值，考虑所有可能满足谓词的行会得到一个下界，将范围限制在肯定满足谓词的行会得到一个上界。对于平均值，下界和上界可以通过还原为和与计数的下界和上界来获得。算法 2 最终返回重写查询的集合，这些查询在算法 1 中执行，产生近似结果。

例 3.*为了计算带蓝色坐垫木椅的报价界限，Thala-musDB 将运行图 6 中重写的查询，计算最小价格的下限。在 FROM 子句中，它引用了原始表 Ads 和 Pics 以及表 S0。后一个表包含在优先处理过程中选择的项目的相似性得分（见第 4.1 节）。这些结果边界缩小的速度取决于具体的谓词和数据分布。对于这种最小查询，ThalamusDB 最好采用优先处理方式，按价格升序对广告进行排序。然而，考虑一个具有挑战性的场景，即有 1000 个销售铅笔的广告（价格低于最便宜的椅子），而用户对计算开销有限制，将可评估的最大条目数限制为 500 。在这种情况下，一些铅笔广告仍未被评估，从而导致下限的准确性降低。这是因为，如表 3 所示，下限值是根据可能满足谓词的行来计算的，即* score > θ_l OR score IS NULL*。要获得零误差的下界，必须对所有铅笔广告进行评估。*

输入 △△△△带有 NL 谓词的查询；o：性能目标
输出 最佳执行计划

函数 OPTIMIZE(△, o)
// 生成查询的所有可能操作。
$A \leftarrow \emptyset$
for all $z \in \triangle.Z$ do
// 添加计算操作。
$A \leftarrow A \cup \{\langle$'compute',$z$, "NULL"$\rangle\}$
for all $col \in \triangle.cols$ **do**
$A \leftarrow A \cup \{\langle$'compute',$z$,"{col}"$\rangle\}$。ASC"$\rangle\}$
$A \leftarrow \cup \{\langle$'计算',$z$, "{col}DESC"$\rangle\}$
// 添加标签请求操作。
$A \leftarrow A \cup \{\langle$'labeling', $z\rangle\}$
// 找到满足用户约束条件的计划。
$e \leftarrow 0^A$
$\delta \leftarrow 1$
while ¬ Cost(e, △)≤ o .b ：
// δ：指数递增增量值（绊≥ 1）。
$\delta \leftarrow$ 繫δ
// 添加一个能最大程度改善受限指标的操作。
// u_a ：单位向量，其中 a 的维度值为 1。
$a^* \leftarrow \arg\min_{a \in A} \max_m (\text{Cost}(e + \lfloor\delta\rfloor u_a, \triangle)_m - o.b)_m$
$e_{a*} \leftarrow e_{a*} + \lfloor\delta\rfloor$
// 本地搜索，直至无成本改进。
$e^* \leftarrow$ Null
$\delta \leftarrow 1$
while $e^* \neq e$ ：
$e^* \leftarrow e$
$c \leftarrow$ Cost（e，△)
$\delta \leftarrow$ 螯δ
// 遍历距离为⌊δ ⌋的相邻计划。
for all $e' \in \{e + s\lfloor u_a \mid s \in \{-1, +1\}; a \in A\}$ **do**
// 如果e' 满足约束条件且成本更低，则进行更新。
$c' \leftarrow \text{Cost}(e', \triangle)$
if $c' \leq o.b$ **and** $c' - o.w < c - o.w$ ：
$e \leftarrow e'$
打破
return e^*

算法 3.ThalamusDB 的优化过程。

在 ThalamusDB 识别出价格最低的第一张椅子之前。尽管如此，我们在现实世界数据集上的实验表明，与基线相比，最小查询实际上是可以显著提高性能的场景。例如，在我们的 Craigslist 数据集中，有人免费提供椅子，这就避免了上述情况。

5 查询优化

算法 3 由查询优化组件执行。作为输入，优化器会考虑要优化的查询以及用户指定的性能权衡偏好和限制。优化器执行多目标优化，同时考虑近似误差、执行时间和用户请求的标

签数量这三个指标。正如

在上一节的详细介绍中，近似误差是由于自然语言谓词的标签缺失或相似性得分缺失造成的。同时，处理开销主要是大型神经网络的推理时间，因此 SQL 查询处理时间可以忽略不计。因此，综上所述，根据这三个指标计算的计划处理成本主要取决于自然语言谓词的处理。因此，优化的重点在于仔细选择用于标记和谓词评估的数据子集。

计划空间。执行计划用向量表示。向量分量表示执行引擎可能采取的行动。操作分为两类：向用户索取标签和计算谓词的相似度得分。每类行动都由目标谓词以及可选的行排序顺序作为参数。对于每个操作，执行计划矢量都包含一个整数，代表操作所涉及的行数（按排序顺序）。

应用。算法 3 收集第 2 行至第 8 行中的相关操作。它考虑了查询 $(\triangle.Z)$ 中的所有自然语言谓词。对于每个谓词，它都会在第 4 至第 7 行中添加代表模型调用的操作。我们考虑了默认的行排序（第 4 行）以及按查询中出现的任何列排序（$\triangle.cols$）。正如第 6 节所讨论的，近似误差可能会因应用操作的行子集而不同。

满足用户约束。给定相关行动集 A，优化器初始化一个空计划（在第 9 行将每个向量分量设为零）。接下来，优化器会贪婪地添加行动，以获得一个满足用户提出的所有成本约束条件的计划（第 11 至 14 行）。给定一个执行计划和一个向量后，函数成本会返回一个向量，其中包含所有三个成本指标的成本估计值。唯一能通过增加操作来改善成本的指标是近似误差。因此，如果无法满足用户定义的误差约束，优化器就会贪婪地增加操作，以最小化近似误差。这个过程保证会终止：如果计算出所有相似性得分并获得所有标签，近似误差必须减小为零。

局部搜索最优计划。接下来，优化器通过局部搜索改进计划，直到找到局部最优（第 15 至 25 行）。一旦上一次迭代中的最佳方案等于当前迭代后的最佳方案，循环就会终止。每次迭代都会考虑当前所选计划在搜索空间中的相邻计划，这些相邻计划是通过改变一种行动类型（即一个向量分量）的行动数量获得的。计划根据其成本向量进行评估。首先，每个

计划必须满足用户定义的成本约束条件，这些约束条件通过成本向量 $o.b$ 表示。表达式为 $c' \ o.b$ 表示多维比较，只有当当前计划的成本向量 c' 低于或达到每个成本指标的界限 $o.b$ 时，才会评估为真。其次，根据加权成本 $c'.o.w$（即计划成本向量与成本权重的点积）对满足成本约束的计划进行评估。一旦发现一个新计划能改善

加权成本。

在构建初始计划和局部搜索时，我们会改变 "步长"，即在一次迭代中增加或减少的操作次数。这是由于我们最初使用固定步长的经验所导致的，因为固定步长会导致优化速度变慢，从而无法满足用户的限制条件。然而，使用较大的步长可能会导致需要较少操作的计划出现次优结果。因此，我们通常从小步长开始，并随着时间的推移逐渐增大，从而在比大步长更精细的搜索空间内优化小计划。这里，"绊 "是一个调整参数，决定了步长增加的速度。需要注意的是，由于最终的行动数必须是整数，因此我们将代表行动数的原始浮点数值去掉。我们在第 7.3 节中总结的实验表明，可变步长可以提高性能。

6 误差估计模型

第 5 节中介绍的优化器依靠模型来估算处理成本、标签数量以及计划执行后的近似误差。ThalamusDB 通过标准方法（使用标记请求数和模型调用数，再乘以 ThalamusDB 在预处理期间测量的每次调用的平均时间）根据前两个指标预测成本。本节将介绍 ThalamusDB 如何在执行特定计划后估算近似误差。

6.1 估计选择性

要估计近似误差，我们首先必须估计自然语言谓词的选择性。计划执行后，给定一个特定的谓词，表中的行可以分为肯定满足该谓词的行、肯定不满足该谓词的行以及不清楚是否满足该谓词的行。ThalamusDB 会估算各类行的比例。首先，ThalamusDB 会根据相似度值（谓词文本与分析条目之间的相似度）估算阈值θ_u，从阈值开始，谓词就会被视为肯定满足。此外，它还会估算阈值 θ_l，低于该阈值时，谓词将被视为肯定未满足（对于相似度值在 $[\theta_l, \theta_u]$范围内的谓词，谓词状态是未确定的）。这两个阈值都是由用户对标签请求的回答决定的。由于用户答案难以预测，ThalamusDB 假设每个（二元）标签请求的两种可能答案分布均匀，并根据这一假设计算估计值。其次，ThalamusDB 会估算相似性值已知且高于 θ_u、低于 θ_l 或在 $[\theta_l, \theta_u]$ 范围内的行比例。计划直接给出已知相似性值的行数（为每个谓词指定应用相关多模态模型的行数）。ThalamusDB 会根据对同一谓词和数据样本进行预处理时获得的相似性值，估算相似性值高于和低于两个阈值的行数比例。

估算复合谓词的选择性更具挑战性。特别是，我们需要考虑到不同的谓词可能在不同的行子集上进行过处理。考虑一下复合谓词（即连接词或析取词）中涉及的两个自然语言谓词。按照选择性估计领域的常见做法 [39]，我们假设统计独立性为

两个谓词之间的排序。用 S 表示优化器考虑的排序顺序集（基于查询聚合中出现的列）。此外，对于$i \in \{0, 1\}$ 和$s \in S$，用$p_{i,s}$ 表示根据排序顺序s 选择的行的比率，其中谓词i 已被评估。然后，我们可以写出

$p = \bar{}_{s \in S} \min_i (p_{i,s})$，以捕捉对两个谓词进行评估的行的总比率。
仅考虑这些行，已知满足谓词连接的行的比率为 $pt\hat{}_1 t\hat{}_2$ ，其中 $i \in \{0, 1\}$ 的 $t\hat{}_i$ *仅指*-1谓词被评估的行。它表示在该行子集内，执行后已知满足该谓词的行的比率（并且是.....）。

根据预处理样本估计）。我们用 $r_i = \bar{}_{s \in S} p_{i,s} - p$ 表示剩余数据的比率$^{\circ}$
个谓词处理过的行。对于这些行，不能保证其他谓词
也进行了评估。但是，不同顺序的行有可能重叠。更确切地说，我们希望同时对两个谓词进行评估的行的比率为 $r_1 - r_2$，即使相应的评估操作使用了不同的行顺序。已知满足所有谓词的行的预期比率为 $r_1 r_2 t\hat{}_1 t\hat{}_2$。因此，总的来说，已知满足连词的行的预期比率是 $t\hat{}_1 t\hat{}_2 (p + r_1 r_2)$。

用$\sigma_T(x)$、$\sigma_F(x)$和$\sigma_U(x)$表示谓词x执行后已知为真、为假或两者皆非的行的预期比率，我们通常有$\sigma_F(x) = \sigma_T(\neg x)$、
$_U\sigma(x) = 1 - \sigma_T(x) - \sigma_F(x)$，并且 $\sigma_T(x_1 \vee x_2) = \sigma_T(\neg(\neg x_1 \wedge \neg x_2))$.因此，我们可以应用同样的

如前推理，获得上述所有选择性值和谓词类型的估计值。

6.2 估计误差

我们利用选择性估计来估计计划执行后不同类型聚合的误差范围。我们用 n_T 表示已知满足所有谓词的连接结果行的预期数量。我们用 n_U x 表示状态未知的行数。对于计数（*）类型的聚合，下界和上界可以立即从选择性估计中获得，即 $[n_T, n_T + n_U]$。对于 sum(a) 类型的集合，我们只需用集合列 a 中的平均值 v_a 乘以前面的界限即可：$[v_a n_T, v_a (n_T + n_U)]$. 我们将平均聚类简化为总和聚类和计数聚类的边界。对于一个集合 avg(a)，我们估算出 $[sum_L(a)/count_U(*), sum_U(a)/count_L(*)]$，其中$sum_L(a)$，$sum_U(a)$，$count_L(*)$，$count_U(*)$--注意计数集合和总和集合的上下限。

我们使用一个辅助函数来估计最小值 min(a) 的下限和上限。考虑值 $V = \{v_i\}$，使得 $v_i \leq v_{i+1}$（即值按升序排序）。函数 $F(V, d, b)$计算以下实验的预期结果：考虑到数值按升序排列，我们在每个数值后掷一枚偏差为b 的硬币，如果掷硬币成功，则返回该数值。如果没有掷硬币成功，则返回默认值 d。在我们的方案中，我们使用这个函数如下。用 V_a 表示出现在聚合列 a 中的值，用 $d = m_a$ 表示数据库中该列的最大值。我们设置 $b = \hat{t}$，其中 $\hat{t}$ 是所有谓词在评估后评估为（肯定）"真 "的概率（计算方法如上一节所述）。考虑到 a 列中的值按升序排列，满足所有谓词的第一行为我们提供了最小值的上限（最大值作为默认值）。函数$F(V_a, m_a, \hat{t})$ 计算的就是这个上界。同样，按照行在列中的值由大到小的顺序排列

从整个数据库中的最小值开始，第一行的谓词评估结果为真或未知状态，为我们提供了最小值的下限。设 $b = \hat{t} + \hat{u}$（即状态为真或未知的概率），V_a 为 a 列中的最小值，$d = \max(V_a)$，表达式 $F(V_a, d, \hat{t} + \hat{u})$也会产生一个下限。

我们计算$F(V, d, b)$ 的方法如下。掷硬币成功的概率为 b。因此，返回V 中第一个值的概率是 b。返回第二个值的概率是 $b - (1-b)$（第二次尝试成功而第一次没有成功的概率）。根据这一推理，我们

得到 $\sum_i b - (1-b)^{i-1} - v_i + (1-b)^{|V|} - d$ 为预期值。对该表达式进行求值

需要检索和遍历数据库内容。对于大型数据库来说，这可能代价高昂（因为在优化过程中会反复调用代价函数）。取而代之的是，我们通过假设V 中最小值和最大值之间的值均匀分布来近似F。这简化了

$\sum_i b - (1-b)^{i-1} - (\min(V) + \delta \cdot i) + (1-b)^{|V|} \cdot d$ where $\delta = (\max(V) - \min(V))/|V|$

是V 中元素之间的平均距离。为了估算边界，相关的值范围来自执行计划中出现的行顺序。我们使用列值统计来

估计特定排序和指定评估行数的最小值和最大值。我们省略了对最大值以及特定排序顺序下的求和与计数的讨论，这些问题的处理方式与此类似。

7 实验评估

第 7.1 节介绍了实验设置。第 7.2 节介绍 ThalamusDB 与 MindsDB [30] 和 ABAE [20] 等基线的端到端性能比较。第 7.3 节详细分析了 ThalamusDB，第 7.4 节提供了误差分析。

表 4.基准数据集概览。

数据集	表格	#行	#Columns	多种模式专栏
YouTube	视频	46,774	11	音频剪辑 音频 说明 文字
Craigslist	图片	14,674	2	文件路径 PICTURE
	广告	3,000	7	标题文本
Netflix	电影	17,770	4	精选评论正文
	评级	100M	4	-

7.1 实验装置

基准测试我们基于来自 YouTube、Craigslist、Netflix 和 IMDb 的真实数据，创建了三个多模态数据处理基准。表 4 显示了数据集的属性概览。YouTube 数据集包含 YouTube 上的视频信息。视频表由包括音频和文本在内的多模态数据列组成。AudioClip 列包含 AudioCaps 数据集 [10, 23] 中音频文件的路径。AudioCaps 数据集由从 YouTube 上提取的简短音频片段组成。我们以关系格式从相应的视频帖子中提取信息（如标题、描述、浏览量和点赞数）来增强 AudioCaps 数据集。Craigslist 数据集包含 Craigslist 网站上的广告信息（如图片、价格、标题）。我们抓取了 3,000 条出售家具的广告帖子，每条广告至少包含一张图片。数据集包括广告表和图片表，因为一个广告可能有不止一张图片。Netflix 数据集 [16] 包含电影及其评分。我们在 "电影 "表中引入了一列新内容，即在每部电影中添加来自 IMDb 评论数据集[28]的评论。YouTube、Craigslist 和 Netflix 基准分别由 24、25 和 6 个查询组成（Netflix 基准由于有一个多模式列，因此查询次数较少）。基准包括最小值、最大值、总和、平均值和计数聚合查询，以及带有限制条款的选择查询。我们在基准查询中对表 4 中描述的音频、图像和文本列引入了 NL 谓词。每个查询至少包含一个 NL 谓词，也可能包含两个通过连接或析取连接起来的谓词组合。对于 Craigslist 和 Netflix 基准，我们引入了将图片表与广告表以及将电影表与评分表连接起来的查询。为了高效地试验大量配置，我们根据手动确定的阈值模拟用户对标签请求的回答。阈值范围从 -28 到 27，即使是相同的模型和列，也会因谓词的不同而有很大差异（例如，Pic.FilePath 列的谓词 "木制 "和 "皮沙发 "的阈值范围在 16 到 27 之间）。阈值范围如此之大，证明了使用高效标签请求校准模型是合理的。

基准。我们将我们的系统与支持基准查询子集的两个最新基准进行了比较。MindsDB

[30] 是一个将机器学习 (ML) 集成到数据库中的数据处理系统。目前（v23.1.3.2），它还不支持图像和音频分类。ABAE [20] 是一个查询处理系统，用于加速涉及昂贵的 ML 方法的昂贵预设值的聚合查询。该系统利用分层抽样和试点抽样技术，因此支持求和、计数和平均值聚合。请注意，EvaDB [17] 的当前版本（v0.3.7）不支持零点分类，因此难以进行比较。

实现ThalamusDB 使用 Python 3 实现。它使用 DuckDB [35] 作为关系数据库管理系统。我们在图像处理中使用 CLIP [36]，在音频处理中使用 Mei 等人的模型 [29]，在文本处理中使用 Sentence-BERT [37] 和 BART [26]。考虑到

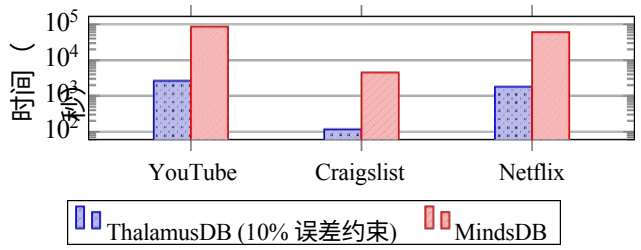


图 7.在所有基准上与 MindsDB 的运行时间比较。

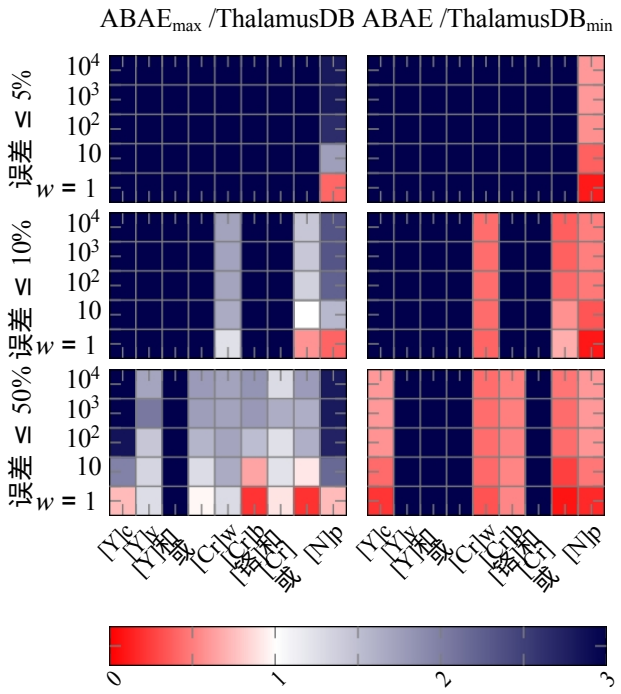


图 8.与 ThalamusDB 相比，ABAE 在所有基准（Y、Cr、N）上使用不同谓词（"和 "表示连词，"或 "表示析取，其余为单个谓词）时的缩放成本。蓝色阴影的程度越高，表示 ThalamusDB 与 ABAE 相比成本越低（红色则表示相反）。

考虑到每个模型的推理速度，ThalamusDB 每个计算步骤处理 1%的图像和文本数据项，0.1% 的音频数据项。在预处理过程中，我们请求三个标签，并对查询中的每个 NL 谓词处理一个计算步骤。所有实验都是在配备了两个英特尔至强 Gold 5218 CPU（2.3GHz，32 个物理内核）、384 GB 内存和两个 GeForce RTX 2080Ti GPU 的服务器上进行的。

7.2 与基准线的比较

MindsDB。在图 7 中，我们展示了 ThalamusDB 和 MindsDB 在 MindsDB 支持的所有查询

上的计算开销（即运行时间）。MindsDB 仅限于涉及文本数据类型自然语言谓词的查询。因此，它只能执行 49 个基准查询中的 12 个。此外，为了解决 MindsDB 不支持收集标签的问题，我们提供了

在不计算额外成本的情况下，我们使用 NL 谓词的基本真实得分阈值来计算系统的误差。对于 ThalamusDB，我们采用了 10% 的误差约束。由于采用了近似查询处理和优先数据评估，ThalamusDB 比 MindsDB 平均提速 35.0 倍。对于单次查询，ThalamusDB 的速度提高了 1,648.4 倍。在本实验中，由于在 MindsDB 上运行 BERT 存在技术问题，我们在两个系统中都使用了 BART。我们在其余实验中都使用了 BERT。

ABAE。在图 8 中，我们展示了 ABAE 与 ThalamusDB 相比的缩放成本。与 ABAE 相比，蓝色越浓表示 ThalamusDB 的成本越低（红色表示相反）。在 [18] 上提供的 ABAE 能够运行 49 个基准查询中的 8 个。与 MindsDB 类似，我们为 ABAE 提供了无计算开销的 NL 谓词标签。我们评估了两个版本的 ABAE：一个版本为每个 NL 谓词收集三个标签（名为 "$ABAE_{min}$ "），这是 ThalamusDB 收集的最少标签数；另一个版本收集最多标签数，以确定精确阈值（名为 "$ABAE_{max}$ "）。ABAE 要求每个谓词都有一个高效的代理模型，以计算所有数据项的代理分数。我们向 ABAE 免费提供根据甲骨文模型评估的基本真实相似性分数作为代理分数（即不计算任何计算开销）。我们为 ABAE 选择的置信度值为 $1 - 10^9$ ，以确保高置信度，尽可能接近 ThalamusDB 的 100% 置信度保证。MindsDB 采用精确查询处理，而 ThalamusDB 和 ABAE 则采用近似查询处理。因此，对于这两个系统，我们在 5%、10% 和 50% 之间改变误差约束，在 1、10、10^2 、10^3 和 10^4 之间改变权重值。最深的蓝色阴影表示成本差异最大，或者 ABAE 在处理与整个数据集相同数量的行时仍无法满足误差约束的情况。在大多数情况下，ThalamusDB 的成本低于 $ABAE_{max}$ 和 $ABAE_{min}$ （分别为 91.1% 和 67.7%）。平均而言，$ABAE_{max}$ 产生了

与 ThalamusDB 相比，$ABAE_{min}$ 产生 2.4 倍的成本开销，而 ThalamusDB 产生 2.1 倍的成本开销。一般来说，当误差约束更严格（如 5%）时，ThalamusDB 的性能比 ABAE 高出更多。对于高选择性谓词（如连词），ABAE 难以满足误差约束，因为它是基于抽样的。在下一小节中，我们将进一步介绍谓词选择性如何影响我们系统的性能。

7.3 进一步分析

我们证明 ThalamusDB 有能力适应用户偏好并实现最佳性能。我们用六种不同的用户偏好配置来评估 ThalamusDB，在下面的图中分别表示为 $Cl, cL, Ce, cE, eL,$ 和 lE。在这里，我们用字母c、l和e来表示计算开销（以秒为单位）、标签数量和误差。与配置相关的两个字母代表成本公式中出现的两个指标，小写字母代表小权重（此处：1），大写字母代表大权重（10^4）。在每种情况下，第三个指标（不以任何字母表示）都会受到阈值的限制：10 秒、每个谓词 5 个标签，以及计算开销、标签数量和误差分别为 10%。

ThalamusDB 的简化版本。针对消融研究，我们引入了 ThalamusDB 的三个简化版本。

首先，我们考虑精确处理基线（命名为 "精确"），该基线会计算受 NL 谓词约束的所有数据项的相似性得分。然后，它通过使用二进制搜索的标签请求为每个谓词找到精确的相似性阈值。第二条基线（名为 "+Ordered"）使用 NL 谓词的最佳评估顺序（考虑选择性和评估成本），以减少计算开销。第三条基线（名为 "+Approx."）利用确定性近似，从而减少了回答查询所需的数据量。ThalamusDB 还使用了基于成本的查询优化器，可根据用户偏好和约束条件优化执行计划。

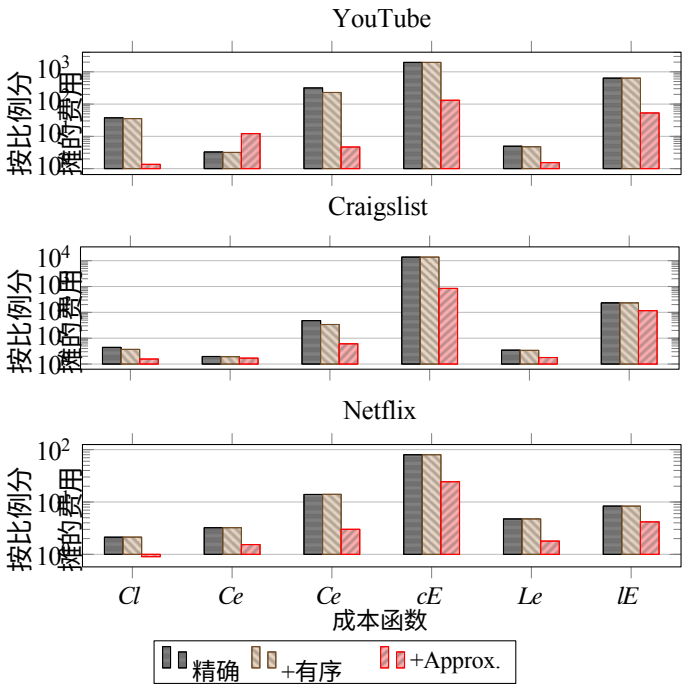


图 9.ThalamusDB 简化版本与最终版本的缩放成本比较（即最终版本的缩放成本为 1，未进行绘图）。

图 9 显示了与 ThalamusDB 相比，不同消融的缩放成本（即 ThalamusDB 的缩放成本为 1）。我们给出了所有基准和不同性能目标规格的平均值。与 ThalamusDB 相比，在所有基准中，Exact 基线的平均成本开销为 951.5 倍。与 Exact 相比，+Ordered 基线的性能更好，但与 ThalamusDB 相比，其平均成本开销为 945.5 倍。而
与其他两个基线相比，+Approx.基线能够减少计算开销，但由于缺乏优化器来选择最有效的操作而受到影响。因此，与 ThalamusDB 相比，它在 YouTube、Craigslist 和 Netflix 上的平均成本开销分别为 34.0 倍、162.1 倍和 6.0 倍。

作为另一项消融研究的一部分，我们停用了集成在优化过程中的粗化方案。在 YouTube 基准中，在优化时间特别耗时的 12 个案例中，使用可变步长（1.01 步）与固定步长（1 步）相比，优化速度平均提高了 2.4 倍，结果质量相当。**运行时间分解。**图 10 显示了 ThalamusDB 不同处理部分的运行时间百分比。在所有情况下，通过模型推理计算相似性得分占运行时间的大部分。平均而言，YouTube 的模型推理占总运行时间的 92.4%，Craigslist 的模型推理占 89.6%，Netflix 的模型推理占 50.7%。相比之下，YouTube、Craigslist 和 Netflix 的关系处理（包括排序数据以进行优先数据处理）平均分别占 4.1%、2.5% 和 45.5%。与其他基准相比，Netflix 的关系处理比例更高。这是因为 Netflix 基准中的自然语言谓词仅适用于文本数据，而不适用于图像或音频数据。还需注意的是，这一细分是基于我们优化后的运行时间，以尽量减少

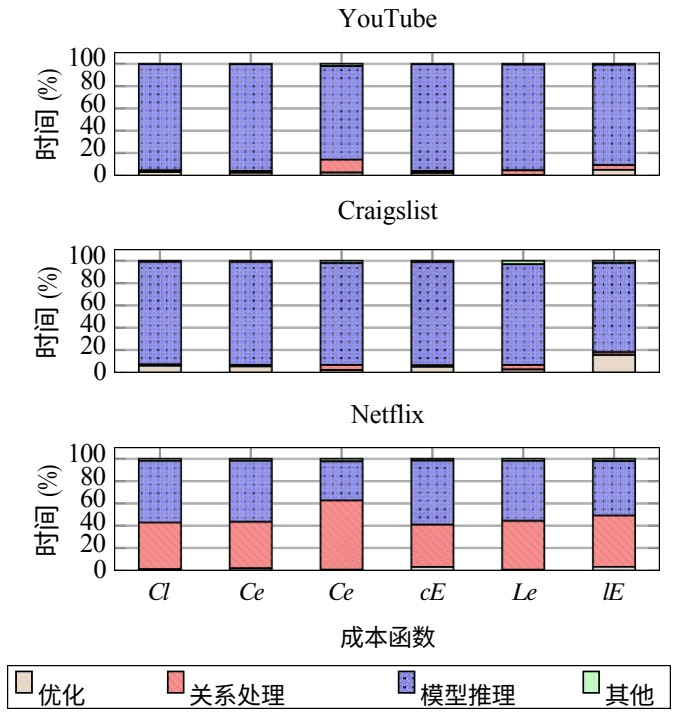


图 10.ThalamusDB 的运行时间分解。

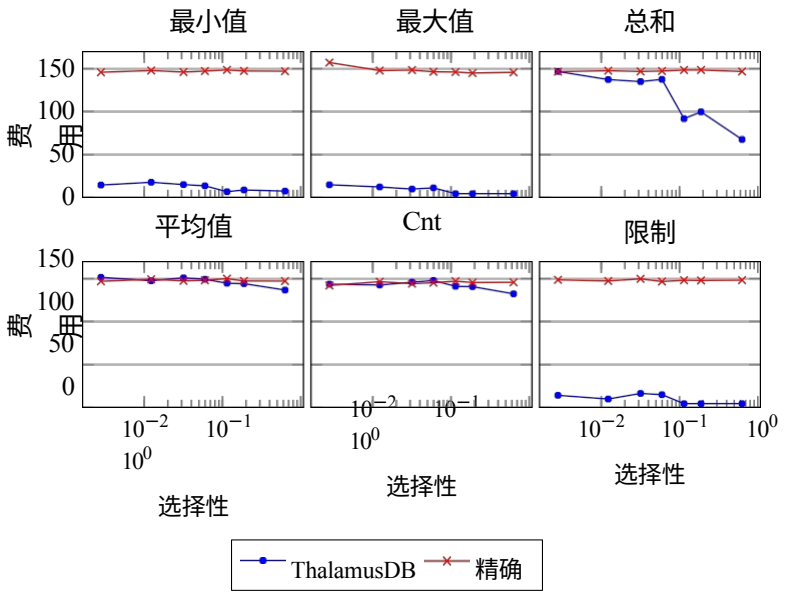


图 11.ThalamusDB 与精确基线在具有不同选择性的多个 NL 谓词上的比较。

推理开销。模型推理耗资巨大，因此有必要投入精力优化数据处理，以减少处理开销

。

谓词选择性图 11 展示了谓词选择性对 ThalamusDB 性能的影响。我们使用 Pic.FilePath 列上的单谓词基准查询，在 Craigslist 数据集上与精确基准进行了比较。具体来说，我们使用了七个 NL

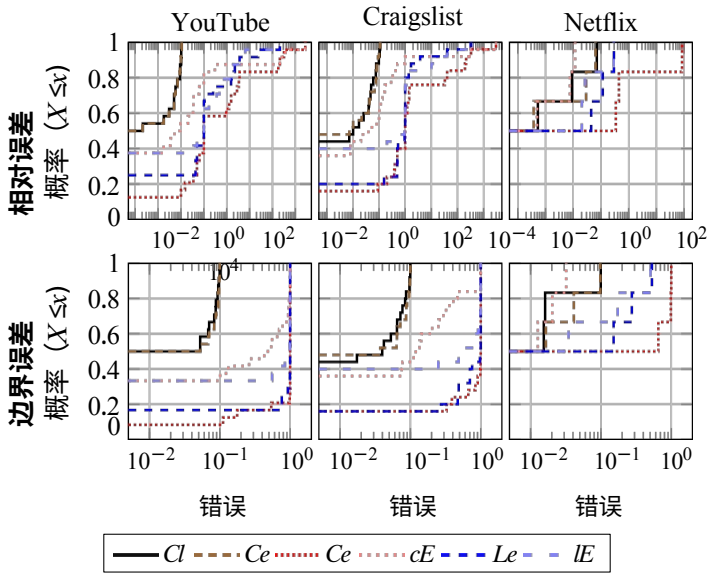


图 12.所有基准查询的相对误差（第一行）和约束误差（第二行）的累积分布函数。

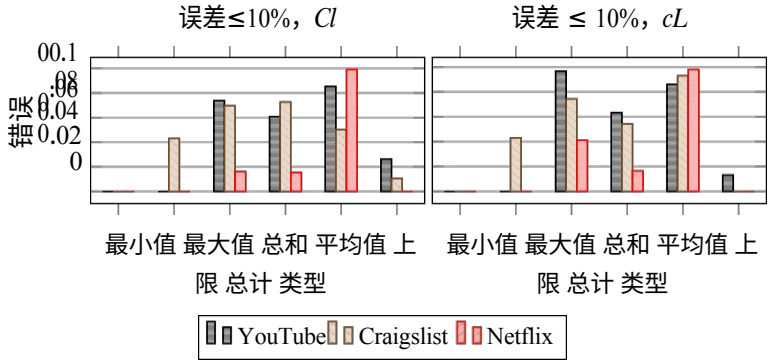


图 13.各类总量的界限误差。

谓词具有不同的选择性：皮沙发、蓝椅、木椅、餐桌、沙发、桌子和木质。由于精确基线在大多数查询中都无法满足对其他两个指标的约束，因此我们使用了对误差指标有约束的性能目标。我们使用w = 1 的权重值，显示每个谓词和聚合函数的成本值。对于总和聚合，ThalamusDB 在选择性较低的谓词方面比 Exact 基线有更大的成本改进。通过优先处理聚合列中数值较大的数据项，ThalamusDB 可以处理较少的数据项来满足误差约束。但是，当谓词的选择性很高时，ThalamusDB 需要处理几乎所有数据项才能满足误差约束。ThalamusDB在最小值、最大值和极限值查询方面实现了较大的成本改进。对于 Count 和 Avg 聚合，ThalamusDB 的改进很小（注意，Netflix 基准在这两种聚合类型上的改进更大）。不过，在所有情况下，选择性较低的谓词的成本都有提高的趋势。

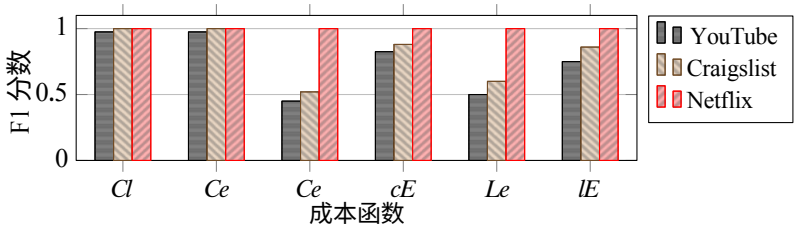


图 14.带 limit 子句的选择查询的 F1 分数。

7.4 误差分析

我们详细介绍了 ThalamusDB 生成的确定性边界的近似误差。我们考虑了两种误差：1) 边界误差和 2) 相对误差。首先，边界误差是定义 5 中定义为 $(u-l)/(u+l)$ 的误差，它测量的是下边界和上边界 l 和 u 之间的相对距离[24]。这个误差的优势在于它不需要我们知道真实值 v_t。因此，它是 ThalamusDB 查询优化过程中使用的误差指标。其次，相对误差的计算公式为 $|(v_t - v_e)/v_t|$，其中 v_t 为真实值，v_e 为 v_t 的估计值。我们通过精确处理获得 v_t，在精确处理过程中，所有项目都要进行评估，并提供足够数量的标签以确定精确的分数阈值。如果模型表现出完美的精确度，v_t 的这个值就是真实值。对于估计值 v_e，我们使用公式 $(u - l)/2$ 将其计算为确定性界限之间的中点。

相对误差和边界误差。图 12 显示了每个基准中所有查询的相对误差和边界误差的累积分布函数。它表明 ThalamusDB 能根据指定的性能有效生成适当的边界误差。目标。当约束误差限制在 10%（*Cl* 和 *cL*）时，随着误差接近 10%，累积概率确实接近 100%。这一观察结果不仅对约束
误差，但也适用于相对误差。正如文献[24]中提出的，这表明当真实值未知时，约束误差是相对误差的良好替代方案。对误差指标赋予较大权重的性能配置（*cE* 和 *lE*）与对应配置（*Ce* 和 *Le*）相比，概率积累更快，表明误差更低。例如，在第 50 百分位数时、
在 YouTube 和 Craigslist 中，*cE* 的相对误差分别为 5.2% 和 3.9%，而 *Ce* 的误差为 100%。同样，在 Netflix 基准中，*cE* 在第 80 百分位数时的相对误差为 0.8%，而 *Ce* 的误差为 44.4%。

每种总量类型的误差。图 13 显示了每种总量的近似误差
类型。我们采用了受误差指标限制的性能目标。与总和、平均值和计数聚合相比，ThalamusDB 在最小值、最大值和极限值聚合方面产生的误差更低。由于最小值和最大值是由单个项目决定的，因此优先处理的可能性会增加。在所有情况下，边界误差都小于或等于 10%，符合误差约束条件。图 14 显示了极限查询在所有基准中的 F1 分数。它显示了当误差受限时最高的 F1 分数（*Cl* and *cL*），其次是当误差权重较大时最高的分数（*cE* and *lE*），而当误差不受限且权重不高时分数较低（*Ce* and *L e*）。

预处理的影响。图 15 显示了在预处理阶段处理不同数量项目时估计选择性的相对误差。随着处理的项目越多，收集的统计数据越多，估算结果也就越准确。然而，随着预处理项目数量的增加，收益也在减少。此外，大量的预处理也限制了 ThalamusDB 减少计算开销的能力。

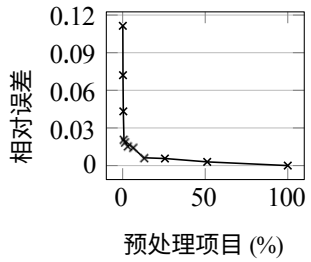


图 15.在不同比例的预处理项目中，估计选择性与真实选择性之间的相对误差。

表 5.误差小于 10%的性能目标的后处理操作比率和误差改进。

基准	成本函数	# Post-处理 / # 所有行动	误差改进
YouTube	*Cl*	0.183	0.153
YouTube	*cL*	0.248	0.246
Craigslist	*Cl*	0.145	0.254
Craigslist	*cL*	0.170	0.247
Netflix	*Cl*	0.007	0.001
Netflix	*cL*	0.000	0.000

后处理的影响。表 5 列出了所有基准中后处理操作的比例和误差改进情况。后处理操作占操作总数的 比例相对较小，在所有基准和成本函数中都低于 25%。同样，后处理操作带来的误差改善也不大（在所有情况下都低于 26%），这表明初始误差已经相当接近目标误差。

模型准确性的影响。图 16 展示了 ThalamusDB 在 Netflix 基准测试中使用两种不同准确度的模型：Sentence- BERT（all-MiniLM-L6-v2）和 BART 所得出的下限和上限。该图将这些界限与采用 Netflix 基准的基本真实标签的精确查询结果进行了比较。BART 是一个比 Sentence-BERT 更大的模型（pytorch_model.bin 的大小为 1.02 GB，而 Sentence-BERT 为 90.9 MB），它允许 ThalamusDB 建立更接近地面实况值的界（对于查询 q3 和 q5）。需要注意的是，由于模型的准确性不尽如人意，这些界限可能并不包含准确的查询结果。不过，ThalamusDB 在所有情况下都能使用较大的模型提供准确的结果边界，只有查询 q3 除外，其结果仍然非常接近地面真实值。

8 相关工作

零镜头模型。传统的分类器（如使用 ImageNet [14] 训练的分类器）随着零镜头和少镜头模型的出现而发展，这些模型利用网络规模文本数据的大量预训练，仅从自然语言描述中就能理解和执行分类任务 [5]。这些模型已经渗透到计算机视觉[36]和音频处理[29]等多个领

域，但在处理大型输入时往往面临限制，需要 ThalamusDB 这样的框架将复杂的查询解构为更易于管理的处理步骤，以实现高效处理。

使用深度神经模型的数据处理框架。SeeSaw[31]、Symphony[7]、VIVA[22]和TASTI[21]等相关作品利用深度神经模型完成了各种数据处理和查询解答任务，而ThalamusDB则利用零点模型支持涉及多模态数据自然语言谓词的查询。其显著特点是

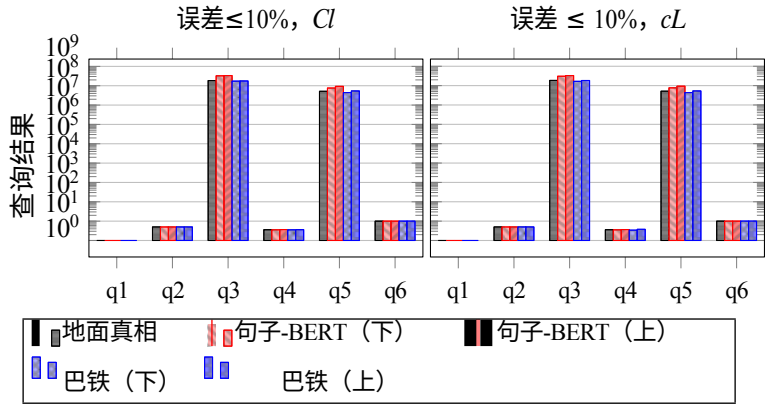


图 16.模型准确性对 ThalamusDB 的影响。使用不同准确度模型的 ThalamusDB 边界与来自地面实况标签的查询结果的比较。

与 MindsDB [30]、EvaDB [17] 和 SQLFlow [40]等精确处理系统相比，ThalamusDB 通过使用近似查询处理，在实验中提高了速度，也显示出明显的效率。最近提出的其他基于神经模型的框架通常侧重于视频数据处理[8, 11, 19, 22, 25]。

分类阈值。受多标签分类研究的启发[13, 38]，Tha-lamusDB 使用阈值为实例分配类别。在多标签分类中，通常会计算标签-实例对的相关性得分，然后为得分超过阈值的实例分配标签[13]。ThalamusDB 是为满足单个用户的需求而设计的，因此假定用户对其希望分类为 "真 "的内容最了解。因此，ThalamusDB 通过向用户征集少量标签请求来确定每个自然语言谓词的分数阈值。

近似查询处理ThalamusDB 与之前大量的近似查询处理相关[6]。与采用数据采样的方法相反 [1-3, 12, 32, 33] 。

包括 ABAE [20]在内，ThalamusDB 利用确定性近似方法 [4、15、27、34]，确保其结果边界具有 100% 的置信度。ABAE 要求每个谓词都有一个代理模型（以便比随机抽样做得更好），它可以计算谓词的近似分数，与甲骨文模型相比，其评估成本要低得多。相比之下，ThalamusDB 并不依赖代理模型。

9 结 论

ThalamusDB 是一个确定性近似查询处理系统，用于在多模态数据上运行复杂查询。它扩展了关系型数据库系统，支持图像、音频和文本数据的自然语言谓词。在对来自 YouTube、Craigslist 和 Netflix 的真实数据集进行的实验中，我们证明 ThalamusDB 的效率明显高于

MindsDB 和 ABAE 等几种基线系统。

10 致谢

本资料基于美国国家科学基金会（National Science Foundation）第 2239326 号奖资助的研究成果。

参考文献

[1] Swarup Acharya, Phillip B Gibbons, and Viswanath Poosala.1999.Aqua：使用近似查询答案的快速决策支持系统。In *VLDB*.754-757.

[2] Swarup Acharya、Phillip B. Gibbons 和 Viswanath Poosala。2000.用于近似回答分组查询的国会样本。*SIGMOD 2000 - Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (2000), 487-498. https://doi.org/10.1145/342009.335450

[3] Sameer Agarwal、Barzan Mozafari 和 Aurojit Panda。2013.BlinkDB：超大数据上的有界错误和有界响应时间查询。In *European Conf*.29-42. arXiv:arXiv:1203.5485v2 http：//dl.acm.org/citation.cfm?id=2465355

[4] Etienne Boursier, E N S Paris-saclay, and Chunbin Lin.2018.柏拉图：具有严格确定性误差保证的压缩时间序列近似分析，第 1 卷。https://doi.org/10.1145/nnnnnnn.nnnnnnn arXiv:arXiv:1709.01073v2

[5] Tom B. Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel M. Ziegler、Jeff Wu、Clemens Winter、Christopher Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray。Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei.2020.语言模型是少数学习者。*神经信息处理系统的进展*。1877-1901. ArXiv:2005.14165

[6] Surajit Chaudhuri、Bolin Ding 和 Srikanth Kandula.2017.近似查询处理：没有银弹。在 *SIGMOD*.511–519. https://doi.org/10.1145/3035918.3056097

[7] Zui Chen, Zihui Gu, Lei Cao, Ju Fan, Sam Madden, and Nan Tang.2023.Symphony：在多模态数据湖上实现自然语言查询 Answering。*CIDR*.

[8] Maureen Daum、Magdalena Balazinska、Brandon Haynes、Ranjay Krishna、Apryle Craig 和 Aaron Wirsing。2022.VOCAL : Video Organization and Interactive Compositional AnaLytics.In *CIDR*.

[9] Benjamin Elizalde、Soham Deshmukh、Mahmoud Al Ismail 和 Huaming Wang。2022.CLAP：从自然语言监督中学习音频概念。*CoRR* abs/2206.04769 (2022). https://doi.org/10.48550/arXiv.2206.04769

[10] Jort F Gemmeke、Daniel P W Ellis、Dylan Freedman、Aren Jansen、Wade Lawrence、R Channing Moore、Manoj Plakal 和 Marvin Ritter。2017.音频集：音频事件的本体和人类标签数据集。In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*.IEEE，776-780。

[11] Brandon Haynes、Maureen Daum、Amrita Mazumdar、Magdalena Balazinska、Alvin Cheung 和 Luis Ceze。2020.VisualWorldDB : A DBMS for the Visual World.In *CIDR*.

[12] Joseph M. JM Hellerstein, PJ Peter J. Haas, and HJ Helen J. Wang.1997.在线聚合。*SIGMOD Record* 26, 2 (1997), 171-182. https://doi.org/10.1145/253262.253291

[13] Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu.2021.多标签意图检测的少量学习。In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*.AAAI Press, 13036-13044. https://ojs.aaai.org/index.php/AAAI/article/ view/17541

[14] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei.2009.ImageNet：大规模分层图像 数据库。(https://doi.org/10.1109/cvprw.2009.5206848

[15] Saehan Jo 和 Immanuel Trummer。2020.BitGourmet：通过优化位选择实现确定性近似。在 *CIDR*。1-7.

[16] Kaggle.2023.Netflix Prize 数据。https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data。

[17] Gaurav Tarlok Kakkar, Jiashen Cao, Pramod Chunduri, Zhuangdi Xu, Suryatej Reddy Vyalla, Prashanth Dintyala,

Anirudh Prabakaran, Jaeho Bang, Aubhro Sengupta, Kaushik Ravichandran, Ishwarya Sivakumar、Aryan Rajoria、Ashmita Raju、Tushar Aggarwal、Abdullah Shah、Sanjana Garg、Shashank Suman、Myna Prasanna Kalluraya、Subrata Mitra、Ali Payani、Yao Lu、Umakishore Ramachandran 和 Joy Arulraj。2023.EVA：端到端探索性视频分析系统。In *Proceedings of the Seventh Workshop on Data Management for End-to-End Machine Learning, DEEM 2023, Seattle, WA, USA, 18 June 2023*.ACM, 8:1-8:5. https://doi.org/10.1145/3595360.3595858

[18] 丹尼尔-康2023.用昂贵的谓词加速近似聚合查询》。https://github.com/ stanford-futuredata/abae。

[19] Daniel Kang, Peter Bailis, and Matei Zaharia.2019.Blazelt：为基于神经网络的视频分析优化声明式聚合和限制查询。*VLDB* 13, 4 (2019), 533-546. https://doi.org/10.14778/3372716.3372725 arXiv:1805.01046

[20] Daniel Kang, John Guibas, Peter Bailis, Tatsunori Hashimoto, Yi Sun, and Matei Zaharia.2021.加速昂贵谓词的近似聚合查询。*Proc.VLDB Endow.*14, 11 (2021), 2341-2354. https://doi.org/10.14778/3476249.3476285

[21] Daniel Kang, John Guibas, Peter D. Bailis, Tatsunori Hashimoto, and Matei Zaharia.2022.TASTI：基于机器学习的非结构化数据查询语义索引。In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022,* Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.).ACM, 1934-1947. https://doi.org/10.1145/3514221.3517897

[22] Daniel Kang、Francisco Romero、Peter Bailis、Christos Kozyrakis 和 Matei Zaharia。2022.VIVA：用于交互式视频分析的端到端 系统。In *CIDR*.计算机械协会。

[23] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim.[无日期]。AudioCaps：为 野外音频生成字幕。

[24] Vladik Kreinovich.2013.如何定义区间估计的相对近似误差？一个建议.*应用 数学科学* 7, 5-8 (2013), 211-216. https://doi.org/10.12988/ams.2013.13019

[25] Sanjay Krishnan, Adam Dziedzic, and Aaron J. Elmore.2019.DeepLens: Towards a visual data management system. *CIDR 2019 - 第 9 届创新数据系统研究双年会* (2019). arXiv:1812.07607

[26] Mike Lewis、Yinhan Liu、Naman Goyal、Marjan Ghazvininejad、Abdelrahman Mohamed、Omer Levy、Veselin Stoyanov 和 Luke Zettlemoyer。2020.BART：用于自然语言生成、翻译和理解的序列间去噪预训练。In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, *ACL 2020, Online, July 5-10, 2020,* Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.).
计算语言学协会，7871-7880。https://doi.org/10.18653/v1/2020.acl-main.703

[27] Xi Liang, Stavros Sintos, Zechao Shang, and Sanjay Krishnan.2021.近似查询处理的聚合与采样（近似）优化组合。In *SIGMOD*.1129–1141. https://doi.org/10.1145/3448016.3457277 arXiv:2103.15994

[28] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.2011.为情感分析学习单词向量》。*第 49 届计算语言学协会年会：人类语言技术》，会议论文集，2011 年 6 月 19-24 日，美国俄勒冈州波特兰市，*Dekang Lin、Yuji Matsumoto 和 Rada Mihalcea（编辑）。计算机语言学协会，142-150。https://aclanthology.org/ P11-1015/

[29] Xinhao Mei, Xubo Liu, Jianyuan Sun, Mark D. Plumbley, and Wenwu Wang.2022.论音频-文本 跨模态检索的度量学习。(2022). arXiv:2203.15537 http://arxiv.org/abs/2203.15537

[30] MindsDB.2023.MindsDB。https://mindsdb.com。

[31] Oscar Moll、Manuel Favela、Samuel Madden 和 Vijay Gadepally。2022.SeeSaw：图像 数据库上的交互式临时搜索。In *ArXiV*.ArXiv:2208.06497 http://arxiv.org/abs/2208.06497

[32] Frank Olken and Doron Rotem.1995.从数据库中随机抽样：一项调查》。*Statistics and Computing* 5 (1995), 25-42. https://doi.org/10.1007/BF00140664

[33] Gregory Piatetsky-Shapiro and Charles Connell.1984.满足 条件的图元数量的精确估算。In *SIGMOD*.256–276. https://doi.org/10.1145/602259.602294

[34] Navneet Potti 和 Jignesh M. Patel。2015.DAQ：近似查询处理的新范式。*VLDB* 8, 9 (2015), 898-909. https://doi.org/10.14778/2777598.2777599

[35] Mark Raasveldt 和 Hannes Mühleisen.2019.DuckDB：可嵌入的分析数据库。*2019 数据管理国际会议论文集》*。1981-1984.

[36] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger 和 Ilya Sutskever。2021.从自然语言监督中学习可转移的视觉模型 。(2021). arXiv:2103.00020 http://arxiv.org/abs/2103.00020

[37] Nils Reimers 和 Iryna Gurevych.2020.Sentence-BERT: Sentence embeddings using siamese BERT-networks.*EMNLP-IJCNLP 2019 - 2019 自然语言处理实证方法会议暨第九届自然语言处理国际联合会议，会议论文集*（2020 年），3982-3992。https://doi.org/10.18653/v1/d19-1410 arXiv:1908.10084

[38] Tal Ridnik、Emanuel Ben Baruch、Nadav Zamir、Asaf Noy、Itamar Friedman、Matan Protter 和 Lihi Zelnik-Manor。2021.多标签分类的非对称损失。*2021 年 IEEE/CVF 计算机视觉国际会议、ICCV 2021，加拿大蒙特利尔，2021 年 10 月 10-17 日*。IEEE, 82-91. https://doi.org/10.1109/ICCV48922.2021.00015

[39] PG G Selinger、MM M Astrahan、D D Chamberlin、R A Lorie 和 T G Price。1979.关系 数据库管理系统中的访问路径选择。在 *SIGMOD*。23–34. http://dl.acm.org/citation.cfm?id=582095.582099

[40] Yi Wang, Yang Yang, Weiguo Zhu, Yi Wu, Xu Yan, Yongfeng Liu, Yu Wang, Liang Xie, Ziyao Gao, Wenjing Zhu, et al.SQLflow: a bridge between SQL and machine learning. *arXiv preprint arXiv:2001.06846* (2020).

2023 年 10 月收到；2024 年 1 月修订；2024 年 3 月接受