论决 策值映射与表征依赖 e

Gil Raitses

arXiv:2602.11295v1 [cs.AI] 11 Feb 2026

摘要

将计算引擎应用于同一数据的不同表示，可能产生不同的离散结果：有些表示会保持该结果，另一些则会彻底改变它。决策值映射记录哪些表示保持该结果、哪些会改变它，并将所声明的表征族中的每个成员与其产生的离散结果相关联。本文对决策值映射进行形式化，并描述了 DecisionDB，一种使用由内容计算得出的标识符、并以一次写入形式存储制品来记录、重放和审计这些关系的基础设施。确定性重放可从已存储的制品中精确恢复每个已记录的决策标识符，且全部三个标识字段均与其已持久化的值相匹配。这一贡献将表征空间划分为持久区域与边界，并将决策复用视为一种可机械检验的条件。

1 引言

分析流水线会产生离散结果，而这些结果取决于输入的表示方式。相同的数据通过相同的查询交由相同的计算引擎处理时，若内部表示发生变化，可能会得出不同的结果。有些表示变化不会影响结果，另一些则会彻底改变结果。

在当前实践中，流程管线以一种表示方式运行并报告结果，而该结果对表征替代方案的敏感性却无从显现。流水线输出无法用于区分哪些决策在表示方式变化后仍可复用，哪些则需要重新论证。

决策值映射 记录哪些表示方式能够保持结果不变、哪些会改变结果，并将表示族中的每个成员与引擎产生的离散结果关联起来。图 1 总结了用于构建此类映射的评估协议：在单一作用域内改变表示方式。

评估链的每个阶段都会记录为内容寻址制品，并由本文介绍的系统 DecisionDB 以只写形式存储。DecisionDB 支持通过系统地改变声明的表示参数来进行表征扫描；支持通过从已持久化制品中确定性地恢复决策标识符来进行重放验证；还支持对完整溯源链进行事后审计。

在一次图路由演示中，两个控制边代价构建的表示参数被扫描，而图快照与最短路径引擎保持不变。其中一个参数在其测试范围内保持决策标识不变；另一个参数在特定阈值处引发离散的标识变化。重放验证精确恢复了所有已持久化的标识符。

2 问题范围

决策值映射处理具有以下结构的系统。快照 s 是在指定时间窗口内外部输入的冻结切片；世界状态的任何变化都会产生新的

快照。表示 $r \in \mathcal{R}(s)$ 是对 s 的确定性编码，由核、阈值、加权规则和聚合策略等明确的结构选择定义；每种表示都由指定的参数集完整定义，并由带版本的工厂生成。引擎 E 是一种计算过程，它接收表示并生成原始输出，分析期间引擎配置和版本保持不变。等价策略 π 是一条声明的规则，用于将原始引擎输出简化为离散的决策标识 $d \in \mathcal{D}$，并定义何时两个原始输出对应相同的标识，而不受偶然数值差异的影响。

在保持快照、引擎和查询不变的情况下改变表示，可以揭示决策标识在哪些情况下能够持续存在，又在哪些情况下会发生变化。核心问题在于：当支撑某项决策的表示被改变时，先前生成的决策是否仍然可接受。持久性和边界的形成被视为计算环境的属性。只有在通过明确声明的策略将连续输出归约为离散决策标识时，才将其纳入考察范围。该映射不涉及训练过程、自适应更新、基于梯度的优化或在线学习。

这种框架适用于离散结果从复杂流程管线中产生，且表示选择可能影响这些结果的任何场景。采用不同成本编码评估的路由、采用不同特征构造评估的分类，以及采用不同聚合规则评估的资源分配，都呈现出相同的结构性问题。

3 决策值映射

研究的核心对象是一个映射

$$f : \mathcal{R} \to \mathcal{D},$$

其中，$\mathcal{R}$ 表示固定快照 s上的一族表示，$\mathcal{D}$ 表示一组离散决策标识。对于每个表示 $r \in \mathcal{R}$，引擎 E 都会生成原始输出 $E(r)$，而等价策略 π 则从中提取决策标识 $d = \pi(E(r))$。

通过对 $\mathcal{R}$进行受控变化，可以观察到此映射的三个结构特征。持久区域是 $\mathcal{R}$ 的连通子集，在这些子集上 f 保持不变；在持久区域内，表征变化不会改变结果。边界是 $\mathcal{R}$ 中 f 值发生变化的位置，将具有不同决策标识的两个持久区域分隔开来。断裂是这样一种边界：表示参数的微小变化会引发离散的标识变化，这表明结果对表示方式高度敏感。

DecisionDB 的目的是实例化 f，在 $\mathcal{R}$中声明的点对其进行评估，将结果存储为不可变制品，然后使生成的映射可供查询、重放和审计。图 1 展示了此协议。

4 扫描协议

一次表示扫描会在一组已声明的表示上评估决策值映射 f。该协议分为五个阶段（见表 1），将冻结的快照转换为可查询的决策映射。

扫描首先冻结一个快照，并为其分配一个内容寻址标识符。随后，通过确定性工厂声明一个表示系列，扫描计划则指定参数网格、引擎和等价策略。对于每种表示，引擎都会独立执行，并生成以不可变制品形式存储的原始输出。等价策略会将每个输出归约为一个决策标识，而决策映射表会记录从快照和表示到

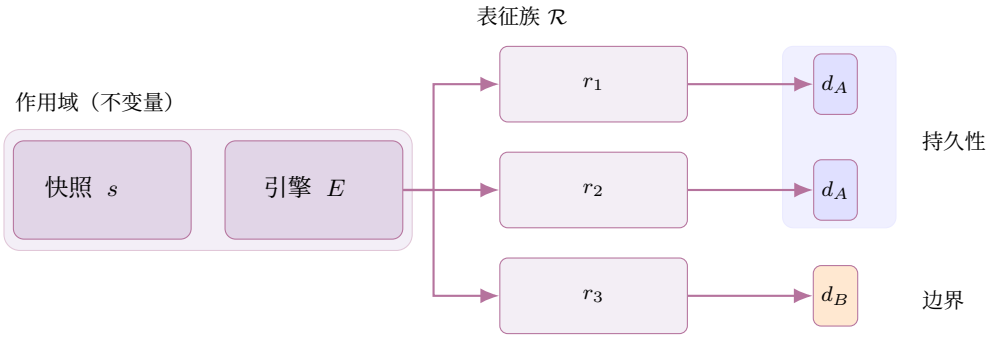


图 1：决策值映射的评估协议。评估过程中，重复使用同一个作用域，该作用域由数据快照和计算引擎组成。在此作用域内，对来自已声明系列的表示进行变化处理，并分别独立评估。不同表示所得决策标识符一致，表明决策标识具有持久性；标识符发生变化，则表明仅由表征变化引起了边界。

引擎运行再到决策标识的关联。所有制品都进行版本管理，并通过内容寻址标识符关联，因此生成的物化映射支持可复现重放和事后分析，无需重新执行引擎。

5 系统设计

DecisionDB 是一个由 SQLite 提供支持的 Python 软件包。它通过包含内容寻址主键和外键约束的关系模式管理五种实体类型。

5.1 内容寻址

所有标识符均根据内容确定性地计算得出。给定某个实体以 Python 字典表示的有效载荷，DecisionDB 会将其序列化为规范 JSON，其中键按字母顺序排列、不含空白字符、数组按声明顺序排列、浮点数表示为字符串，并包含显式版本字段。随后，它会计算该 JSON 的 UTF-8 编码的 SHA-256 摘要，截取前 16 个十六进制字符，并添加特定于类型的前缀。无论计算在何时何地进行，相同内容始终会生成相同的标识符。

5.2 模式

该关系模式包含五个核心表。图2展示了它们的外键关系，表2总结了每个表的作用。每个表都强制使用内容寻址主键和外键约束，连接从快照到表示、引擎执行再到决策标识的完整溯源链。所有写入均为仅追加，限定在实验标识符范围内，并在事务中执行。插入采用「插入或忽略」策略来保证幂等性，因此重复插入相同的内容寻址实体不会产生任何操作。

5.3 等价策略

等价策略定义如何将原始引擎输出归约为决策标识。每项策略都会指定一个哈希来源，用于标明原始输出中的哪个字段承载与决策相关的内容，例如路径节点序列。它还会指定一项规范化规则，用来决定如何

表 1：五阶段表征扫描协议。

	阶段	输入	输出
1	冻结快照	世界状态、时间窗口	快照标识符
2	声明表征	快照引用，参数化、工厂版本	表征标识符，每个参数设置一个
3	计划扫描	参数网格、引擎名称和版本，等价策略	扫描计划描述符，本身采用内容寻址
4	执行引擎	每种表示独立地	运行记录、原始输出制品存储为不可变文件
5	提取决策	原始输出、等价策略 π	决策标识符，决策映射条目连接表示与运行与决策

序列化提取出的内容，并配以一条匹配规则，用于判定标识是否相同，例如 SHA-256 相等。策略本身采用内容寻址，因此，对策略定义的任何更改都会生成新的策略标识符，以及新的下游决策标识符。

5.4 重放验证

重放验证会读取一条已持久化的决策记录，重新加载存储的原始输出和策略规范，重新计算策略 ID、负载哈希和决策标识符，然后将它们与已持久化的值进行核对。由于重放是只读的，不会写入任何行，因此验证成功表明，从原始输出到策略应用再到决策标识的内容寻址链具有确定性，且内部一致。

6 实证演示

图路由问题提供了一个具体场景，用于演示决策值映射如何使对表示的依赖变得可观察。

6.1 设置

本演示固定一个有向图，其节点集为 V，其中 $|V| = 564$；边集为 E，并包含不可变的边属性，其中包括根据地理距离得出的基准成本和归一化应力指标。该快照采用内容寻址，并在任何引擎执行之前封存。起点为 85、终点为 50 的单个起讫点查询在开始时即固定。

每种表示都将图的边成本编码为固定边属性的确定性函数。两个表示参数控制成本曲面。第一个参数是邻居权重，它为基于邻居的成本分量赋予权重，并分别取值 0.5 和 1.0 进行测试。第二个参数是二阶权重，它为二阶成本分量赋予权重，并分别取值 0.25 和 0.5 进行测试。每次扫描只改变一个参数，同时固定另一个参数，因此每次扫描会产生两种表示变体，总计进行四次引擎评估。

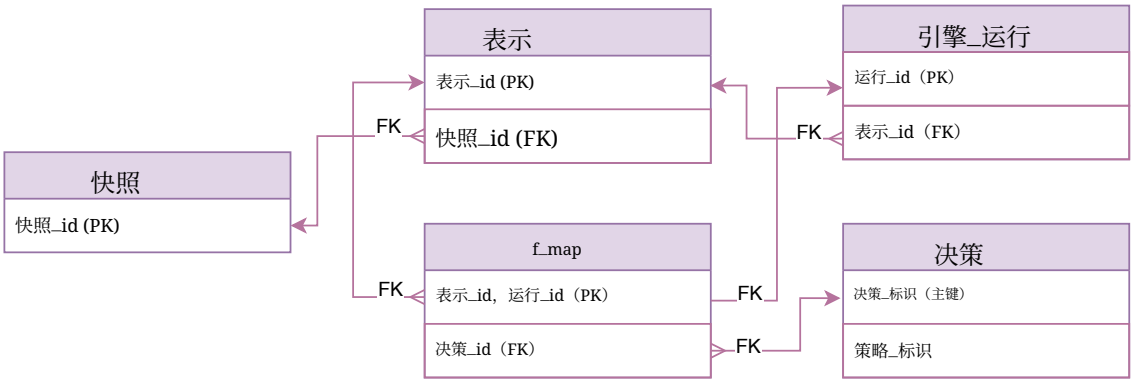


图2：DecisionDB 关系模式。五张表构成一条内容寻址溯源链。外键箭头表示引用依赖的方向，连接表示与其父快照、引擎运行与所消耗的表示、决策映射表与表示、运行及决策标识。

表2：DecisionDB 实体角色。

表	键前缀	Role
快照	snap	不可变输入状态及其制品清单
表示	repr	在以下条件下对快照进行确定性编码所声明的参数化
引擎运行	run	固定引擎在特定表示
决策	dec	由一种等价策略
f 映射	复合体	关联的物化决策值映射表示、运行和决策的物化决策值映射

该引擎是一个使用 Dijkstra 算法 [4], 的最短路径求解器，所有运行的配置和版本均保持不变。执行时间介于 0.5 至 1.4 毫秒之间。

当且仅当两条路线遍历相同的节点序列时，它们才会被赋予相同的决策标识。等价策略通过根据路线节点的 JSON 序列化表示（键已排序并以 UTF-8 编码）计算标识符来确保这一点。

6.2 结果

在此实例中，决策标识对应路线等价类，但该协议适用于固定决策查询所产生的任何离散结果。

邻居权重扫描将邻居权重从 0.5 调整至 1.0，同时将二阶权重固定为 0.25。两种表示均得到相同的 16 节点路径（决策 A），表明在测试范围内具有持久性。

二阶权重扫描将二阶权重从 0.25 变动至 0.5，同时将邻居权重固定为 0.5。当二阶权重为 0.25 时，引擎生成决策 A。当二阶权重为 0.5 时，引擎生成决策 B，即一条不同的、穿过某一

表3：表征扫描结果。

扫描参数	值	决策	节点	边界
邻居权重	0.5	A	16	
邻居权重	1.0	A	16	No
二阶权重	0.25	A	16	
二阶权重	0.5	B	14	Yes

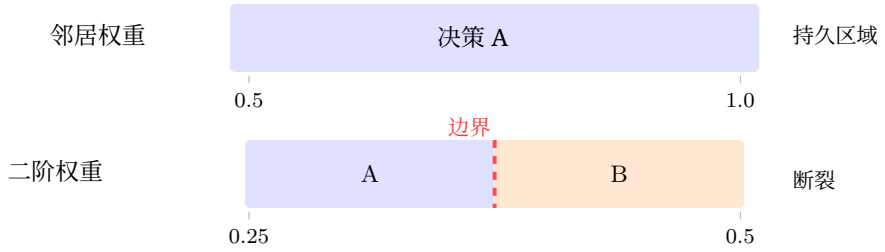


图 3：两个表示参数的决策标识持久性与断裂结构。邻居权重参数涵盖单个持久区域，在该区域内，两个测试值均产生决策 A，表明在此范围内决策标识保持稳定。二阶权重参数涵盖两个由断裂分隔的区域，在 0.25 到 0.5 之间，决策标识从决策 A 变为决策 B。这个两点样本无法确定确切的阈值。

图中不同区域的 14 节点路径。这两个决策标识之间的边界位于二阶权重值 0.25 和 0.5 之间，在此处，微小的参数变化会诱发离散结果的定性变化。每个决策的完整节点序列均作为原始输出制品存储，并可通过重放验证恢复。表3总结了扫描结果；图3展示了持久性和断裂结构。

6.3 重放验证

重放验证针对扫描生成的一项决策执行。该过程重新加载已存储的原始输出和策略规范，重新计算全部三个标识字段，以便与已持久化的值进行核对。正如表4所示，所有重新计算的值均完全匹配。

重放为只读操作，不会改变数据库状态。重放前后的表计数完全相同，均为 1 次引擎运行、1 项决策和 1 个 f 映射条目。这证实了从原始输出、经由策略应用到决策标识的内容寻址链具有确定性，并且整个过程均可审计。

7 相关工作

分析结论对表示方式和分析选择的敏感性，已在多个学科中得到证实。本节将决策值映射框架置于有关分析灵活性、可复现性基础设施和数据溯源系统的现有研究中加以讨论。表5总结了主要区别。

表4：重放验证结果。

字段	已持久化	重新计算
策略 ID	pol_d8da3e00e9584eb1	pol_d8da3e00e9584eb1
负载哈希	3a9d63ac28378116	3a9d63ac28378116
决策 ID	决策_e28092c4dc33b8f1	决策_e28092c4dc33b8f1

7.1 分析灵活性与多元宇宙方法

统计结论对分析选择的依赖性，已从多个互补视角得到考察。规格曲线分析 [14]评估报告的效应如何随合理的分析设定而变化。多重宇宙分析 [15] 则进一步同时改变数据处理和模型设定决策，以刻画与某一数据集相符的全部结果空间。即使研究人员遵循预注册方案，这种不稳定性也依然存在：隐含的自由度会影响研究发现，而无需有意进行P值操纵 [6]。在观察性研究中，这种影响规模可观，效应估计值会随模型设定而波动 [11]。

决策值映射与其他方法有着共同前提，即分析结论取决于一些往往未明示的选择。规格曲线分析和多元宇宙分析基于回归系数和p值等连续效应估计值，并将其分布可视化。决策值映射则针对离散结果身份，刻画表征空间的拓扑，识别哪些区域能够保持身份，以及边界在哪里形成。由此得到的是对表征空间的划分，与规格曲线分析和多元宇宙分析所产生的分布形成对照。

7.2 敏感性分析

敏感性分析量化模型输入的变化对模型输出变化的贡献，通常通过连续输出空间中的方差分解或基于导数的指标来实现 [13]。敏感性分析对输出方差进行分解并计算敏感性指数，而决策值映射则直接观察离散结果在表示变化下是改变还是保持不变。决策值映射与敏感性分析兼容，因为可以基于二值化的决策映射计算敏感性指数，但它并不要求或假定存在连续输出度量。

7.3 机器学习中的可复现性

满足相同训练标准的流程管线，在分布偏移下可能产生行为迥异的预测器 [3]。深度强化学习的结果对实现细节和超参数选择十分敏感，而标准报告方式往往掩盖了这一点 [8], ；机器学习基准中的方差核算已实现形式化，以量化这种影响 [2]。可复现性清单 [12] 鼓励报告实验细节，却未涵盖内容寻址溯源，也未包含确定性重放。

决策值映射解决的是一个相关但有所不同的问题。可复现性研究记录训练运行和超参数设置之间的差异，而决策值映射则固定快照和引擎，同时改变表示，并追踪其对离散结果标识的影响。决策值映射通过提供一种

Ta表 5：决策值映射与相关方法的比较总结 .

维度	典型方法	决策值映射
分析对象	连续效应 估计值 ，方差 ， p值	离散决策恒等式
变异来源	模型规格， 预处理， 超参数	表示参数 在固定快照和 引擎
输出 表征	分布、敏感性 指数、设定 曲线	持久区域、 边界、断裂
基础设施	日志记录、数据溯源 元数据、工作流 图	内容寻址， 可重放决策映射

更底层的基础设施，用于测试特定结果在特定表示变化下是否稳定。

7.4 溯源与工作流系统

通用溯源词汇表记录计算产物的派生历史 [10]。科学计算的工作流溯源 [5], 、机器学习流水线的实验跟踪 [17], 、用于可复现执行的标准化工作流定义 [1] 以及面向容错分布式计算的沿袭跟踪 [16]，都涉及这一问题的不同方面。

此处描述的诊断层在范围上比这些系统更窄，其不变量也更具体。通用溯源系统管理任意工作流并跟踪通用溯源图，而这一层则强制采用特定结构，其中包含不可变快照、声明式表征族、固定引擎和等价策略，可将原始输出缩减为离散标识。内容寻址方案确保相同输入始终产生相同标识符，而重放验证则检查溯源链的端到端一致性。这种结构的特异性产生了一个决策值映射中的可查询诊断对象，这是通用溯源系统无法直接支持的能力。

7.5 基础设施先例

这种方法遵循了基础抽象发展过程中观察到的一种模式。抽象数据类型 [9]将表示与可观察行为分离，使系统能够在内部不断演进，同时保持对外保证。预写日志 [7]将持久性从一种临时属性转变为可审计、可重放的状态转换协议。在这两种情况下，主要贡献都在于引入了一个诊断层，使结构依赖变得明确且可测试。

每项决策都发生在一个作用域内，该作用域由快照、引擎和查询定义。协议记录结果标识如何在该作用域内跨不同表示变体演变。在评估期间调整模型参数的方法，着力于在该作用域内生成更优的解决方案。本协议则转而关注复用，探讨记录的上下文是否足以证明将离散结果跨表示及下游

使用者传递是合理的。决策值映射将这一脉络扩展到分析流水线，其输出是离散且影响重大的，并取决于表示方式的选择。

8 局限性

决策值映射适用于输出可通过已声明的等价策略归约为离散身份的系统。除非明确定义了这种归约方式，否则概率分布和回归曲面等连续输出不在讨论范围内。等价策略的选择直接决定了何为“相同结果”；对于同一原始输出，采用不同策略通常会得到不同的决策映射。

DecisionDB 不会跟踪决策映射如何随快照或引擎版本演变。本文报告的结果假定单一快照和引擎保持不变，因此每种快照与引擎的组合都需要单独分析。输入数据、模型参数或执行逻辑的任何变化，都会构成新的分析情境。

报告中的扫描涵盖两个表示参数：邻居权重取值为 0.5 和 1.0，二阶权重取值为 0.25 和 0.5；这些参数应用于单个图快照和单个起终点对。表征空间总共在四个点上进行了采样，未观测区域则不受约束。

此处识别出的持久区域和边界，未必适用于更精细的参数网格、不同的起终点对或不同的图拓扑。

实证演示采用图路由。该方法旨在不依赖特定领域，但尚未在分类、资源分配或其他类型的流程中得到验证。将决策值映射应用于新领域，需要定义恰当的等价策略，而这需要结合领域知识判断何为“相同结果”。

观察到两个参数值之间存在边界，并不能解释边界为何存在。该框架能够识别决策标识在表征变化下发生改变的位置，但无法将这种改变归因于引擎或表示构造中的任何特定机制。

当前实现使用 SQLite，并已在个位数的表征族上进行测试。要扩展至包含数百或数千种表示的大型参数网格，需要评估存储、查询性能和扫描编排，但这些工作均尚未开展。

9 结论

在一次图路由演示中，一种表示参数在测试范围内保持决策标识不变，而另一种则导致标识发生离散变化。DecisionDB 通过五阶段扫描协议、五表关系模式和回放验证流程，将这一结构具体化；该流程能够准确恢复所有已持久化的标识符。

这种方法受到多方面限制：仅适用于离散结果，依赖保持不变的快照和引擎，且本文报告的实证覆盖范围有限。一个自然的扩展方向是加密参数扫描，在更多领域应用相同的协议，并引入跨快照比较程序，将“相同查询，不同快照”作为复用可接受性的一个明确维度。

这项工作的贡献在于基础设施层面：引入一个诊断层，用于记录离散决策如何从表示族中产生。决策值映射直接揭示稳定性与断裂，并将明确的表示上下文置于下游复用的核心。

决策，而置信度或性能摘要则起到补充作用。决策复用成为一个兼容性问题，可在跨任务和不同时间尺度部署之前通过机械方式进行检查。

参考文献

[1] Peter Amstutz、Michael R. Crusoe、Nebojˇsa Tijani´c、Brad Chapman、John Chilton、Michael Heuer、Andrey Kartashov、John Kern、Dan Leehr、Herv´e M´enager 等。通用工作流语言 v1.0。https://www.commonwl.org/v1.0/，2016 年。[2] Xavier Bouthillier 、 C´esar Laurent 和 Pascal Vincent。机器学习基准中的方差考量。机器学习与系统会议论文集，3：747–769，2021 年。[3] Alexander D'Amour、Katherine Heller、Dan Moldovan、Ben Adlam、Babak Alipanahi、Alex Beutel、Christina Chen、Jonathan Deaton、Jacob Eisenstein、Matthew D. Hoffman 等。规格不足给现代机器学习中的可信度带来了挑战。机器学习研究期刊，23：1–61，2022 年。[4] Edsger W. Dijkstra。关于与图相关的两个问题的一则说明。数值数学，1：269–271，1959 年。[5] Juliana Freire、David Koop、Emanuele Santos 和 Cl´audio T. Silva。计算任务的数据溯源：综述。计算科学与工程，10(3)：11–21，2008 年。[6] Andrew Gelman 和 Eric Loken。分叉路径花园：为何多重比较可能成为问题，即使不存在"探索性搜寻"或"P值操纵"，且研究假设早已提出。哥伦比亚大学统计系，2013 年。[7] Jim Gray 和 Andreas Reuter。事务处理：概念与技术。Morgan Kaufmann，1993 年。[8] Peter Henderson、Riashat Islam、Philip Bachman、Joelle Pineau、Doina Precup 和 David Meger。具有重要意义的深度强化学习。载于美国人工智能协会人工智能会议论文集，第 32 卷，2018 年。[9] Barbara Liskov 和 Stephen Zilles。使用抽象数据类型编程。载于*ACM SIGPLAN* 超高级语言研讨会论文集，第 50–59 页。ACM，1974 年。[10] Luc Moreau 和 Paolo Missier。PROV-DM：PROV 数据模型。W3C 建议标准，2013 年。https://www.w3.org/TR/prov-dm/。[11] Chirag J. Patel、Jay Bhatt、Atul J. Patel 和 John P. A. Ioannidis。针对 2 型糖尿病的全环境关联研究（EWAS）。*PLoS ONE*，10(7)：e0132002，2015 年。[12] Joelle Pineau、Philippe Vincent-Lamarre、Koustuv Sinha、Vincent Larivi`ere、Alina Beygelzimer、Flo- rence d'Alch´e Buc、Emily Fox 和 Hugo Larochelle。提升机器学习研究的可复现性（NeurIPS 2019 可复现性计划报告）。机器学习研究期刊，22(164)：1–20，2021 年。[13] Andrea Saltelli、Marco Ratto、Terry Andres、Francesca Campolongo、Jessica Cariboni、Debora Gatelli、Michaela Saisana 和 Stefano Tarantola。全局敏感性分析：入门指南。John Wiley & Sons，2008 年。[14] Uri Simonsohn、Joseph P. Simmons 和 Leif D. Nelson。规格曲线分析。自然·人类行为，4(11)：1208–1214，2020 年。[15] Sara Steegen、Francis Tuerlinckx、Andrew Gelman 和 Wolf Vanpaemel。通过多重宇宙分析提高透明度。心理科学展望，11(5)：702–712，2016 年。[16] Matei Zaharia、Mosharaf Chowdhury、Tathagata Das、Ankur Dave、Justin Ma、Murphy McCauley、Michael J. Franklin、Scott Shenker 和 Ion Stoica。弹性分布式数据集：一种用于内存集群计算的容错抽象。载于第 *9* 届 *USENIX* 网络系统设计与实现研讨会论文集（*NSDI*），第 15–28 页，2012 年。[17] Matei Zaharia、Andrew Chen、Aaron Davidson、Ali Ghodsi、Sue Ann Hong、Andy Konwinski、Siddharth Murching、Tomas Nykodym、Paul Ogilvie、Mani Parkhe 等。借助 MLflow 加速机器学习生命周期。*IEEE* 数据工程简报，41：39–45，2018 年。