Csql：将文档映射到因果数据库 *

预印本

Sridhar Mahadevan Adobe Research 和马萨诸塞大学阿默斯特分校 smahadev@adobe.com, mahadeva@umass.edu

arXiv:2601.08109v1 [cs.DB] 13 Jan 2026

2026年1月14日

摘要

我们介绍了一种名为 Csql 的新系统，它能自动将非结构化文本文档集合转换为可通过 SQL 查询的因果数据库（CDB）。CDB 与传统数据库不同：它旨在通过因果干预和结构化因果查询回答“为什么”这类问题。Csql 基于我们早期开发的 Democritus 系统，该系统可将文档转换为数千个源自因果话语的局部因果模型 [Mahadevan, 2025b]。不同于基于 RAG的系统或以知识图谱为中心的方法，Csql 支持对文档集合进行因果分析，而不只是关联检索。例如，给定一篇关于人类双足行走起源的文章，Csql 可以支持如下查询：“对双足行走影响最强的因果因素有哪些？”或“哪些变量充当因果枢纽，具有最大的下游影响？”

除了单文档案例研究之外，我们还展示了 Csql 可以大规模摄取检索增强生成/由信息抽取编译的因果语料库：通过将经济学论文的因果主张检验（TCC）数据集编译为因果数据库，其中包含 265,656 个主张实例，涵盖 45,319 篇论文、44 年及 1,575 个报告的方法字符串，从而支持在 SQL 中进行语料库级因果查询和纵向分析。从概念上说，Csql 将文档集合转换为具有因果基础的关系数据库，从而能够通过标准 SQL 进行因果分析。与依赖人工设计本体、固定模式或特定领域信息抽取流程的既有工作不同，Csql 直接从语言（或由语言编译而成的因果产物）中归纳其模式。抽象地看，Csql 充当编译器，将非结构化文档转换为配备了规范查询代数的因果数据库，并可广泛应用于众多领域，包括商业、经济学、人文科学和科学。

关**键词** 因果性 · 自然语言 · 数据库 · SQL · 人工智能 · 机器学习

1 引言

我们介绍 Csql，这是一个因果知识发现系统，可自动将文档集合转换为可通过 SQL 查询的因果数据库（CDB）。与检索增强生成（RAG）系统或以知识图谱为中心的方法不同，后者生成的是传统数据库，而 Csql 支持对文档集合进行因果分析，而非仅进行关联检索。Csql 与传统因果推断方法也有根本区别 [Imbens and Rubin, 2015, Pearl, 2009]，；后者通常侧重于基于数值数据或实验数据开展范围狭窄、特定领域的研究。相比之下，Csql 直接处理大规模非结构化文本集合。

Csql 的一个核心特点是，它直接从语言中归纳出关系模式。与依赖人工设计的本体、固定模式或特定领域信息抽取流程的以往方法不同，Csql 会自动构建一个以因果关系为基础的数据库，其结构源自

*修订中的草稿。

属性	RAG系统	知识图谱	信息抽取流水线	Csql
主要目标	答案生成	结构化事实	实体/关系抽取	因果分析
输入	文本块	整理后的三元组	带注释的文本	非结构化文档
模式	无（隐式）	预定义本体	固定模式	**从话语中归纳**
因果关系支持	None	隐式／非正式	仅成对	**加权因果关系**
不确定性	None	罕见/临时	None	**聚合模型支持**
相互冲突的主张	不支持	人工解决	不支持	**已保留并量化**
组合性	基于提示	有限图遍历	有限	**SQL 连接和聚合**
反事实	No	No	No	**通过因果结构支持**
输出格式	生成文本	图数据库	三元组 / 表	**关系型因果数据库**
执行模型	神经推断	符号查询	批量提取	**确定性 SQL**

表 1：Csql 与检索增强生成系统、知识图谱及传统信息抽取（IE）流程的比较。Csql 的不同之处在于，它将因果话语编译为可通过 SQL 查询的因果数据库，并明确表示不确定性且支持因果分析。

话语本身。从抽象层面来看，Csql 就像一个编译器，能够将非结构化文档转换为配备了系统化查询代数的因果数据库。这种设计使其能够广泛应用于科学、公共政策、经济学和历史等领域。

表 1 对比了传统检索增强生成或知识图谱构建的数据库与 Csql 构建的因果数据库。Csql 直接基于我们此前的 Democritus 工作，该工作提出了一套端到端系统，用于从语言中发现和评估因果结构。给定一份文档，Democritus 会自动构建数千个局部因果模型，评估其合理性，并将其汇总为以自然语言表达的排名因果分析。Csql 更进一步，将 Democritus 的输出编译为可通过 SQL 查询的数据库，从而能够对整个文档集合进行系统性的因果查询。

Csql 依托范畴因果关系领域的最新进展 [Mahadevan, 2025a, Fritz, 2020] 以及几何深度学习领域的最新进展 [Fong et al., 2019, Mahadevan, 2024, Gavranovi´c et al., 2024], ，并将大语言模型（LLM）用作话语编译器。大语言模型能够生成丰富的因果叙事——列举子主题、提出因果问题，并阐述从宏观经济学到神经科学等不同领域中的机制。然而，仅靠基于大语言模型的文档摘要，无法生成具有因果结构且可查询的数据库。Csql 填补了这一空白，将因果话语转换为明确的关系表示。

例如，给定一篇报纸文章——比如《华盛顿邮报》最近的一篇报道，探讨食用黑巧克力是否可能延长寿命，[2]——Csql 可以构建一个因果数据库，支持如下查询：“对黑巧克力消费影响最大的因果影响是什么？”或“在这一话语中，哪些变量充当下游因果枢纽？”更广泛地说，Csql 旨在支持对因果主张进行探索性比较分析，分析范围不仅限于单篇文档，还涵盖从整个文档集合中提取的因果话语框架。

Csql 的一项关键设计选择是构建这一话语框架。系统并不将分析局限于输入文档本身，而是将每篇文档视为更广泛因果邻域的种子。Democritus 首先对输入文本进行主题发现，然后利用语言模型扩展这些主题，从而构建局部话语流形。因果主张会依据这一扩展后的语境生成并评估。因此，Csql 推理的不仅是文档明确陈述的内容，还包括其中的主张如何置身于更广阔的因果话语图景之中。

2 双足行走的起源：一个贯穿全文的示例

我们将通过一个贯穿全文的示例来说明 Csql 的构建过程。该示例基于《华盛顿邮报》的一篇文章，探讨人类祖先双足行走的起源，可追溯至700万年前（见图1）。[3]。构建 CSQL 数据库的过程，首先要使用我们此前开发的 Democritus [Mahadevan, 2025b] 系统，对包含这篇文章的 PDF文档进行预处理。Democritus 是一种系统，可通过向大语言模型（LLM）提出精心设计的查询，构建大型因果有向无环图模型 [OpenAI, 2025,

[2]https://www.washingtonpost.com/wellness/2025年12月26日/dark-chocolate-health-benefits[3] urlhttps://www.washingtonpost.com/科学/2026年1月2日/human-ancestor-biped/

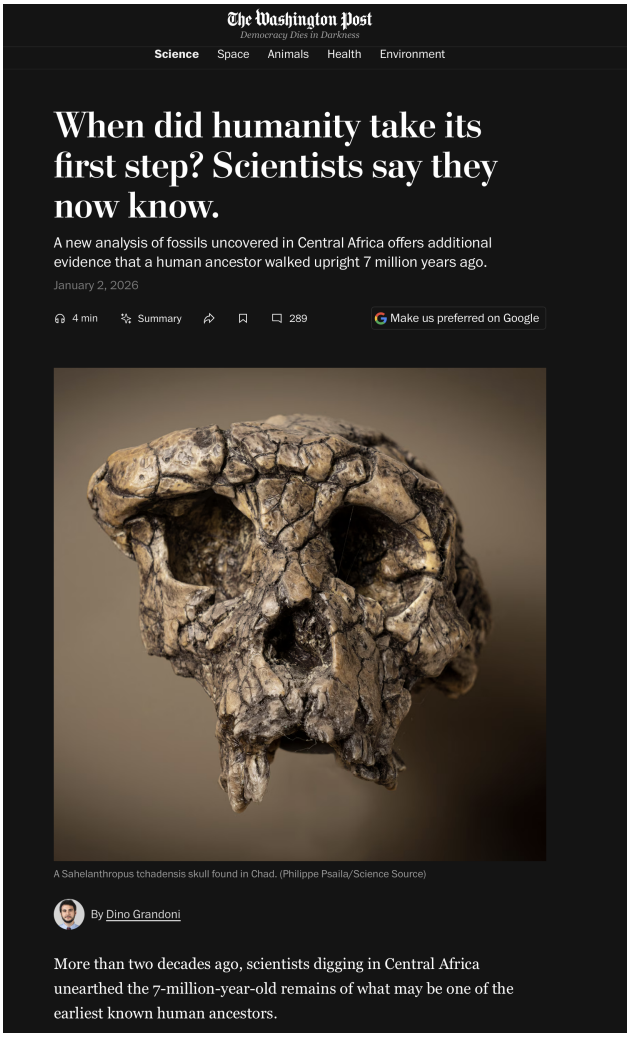


图1：《华盛顿邮报》上关于人类双足行走起源的文章 g.

2024, Anthropic, 2024, Dubey 等人, 2024, Mistral AI, 2024]。Democritus 能够从 PDF 格式的文本文档中构建丰富的有向无环图模型库，对这些主张进行定量评估和排名，然后生成自然语言执行摘要。图2展示了根据这篇文章自动构建的局部因果模型示例。

Democritus 会针对文档中的因果主张，以及运行期间自动生成的周边背景信息，撰写一份详细的书面摘要。以下是这类报告的简短摘录。

> # 因果深度分析 —
> 0001_when_did_humanity_take_its_first_step_scientists_say_they_no_70ccf7f26a62

本说明是对 Democritus 可信度分析的通俗解读。它**比人工智能概述更详细**，但**不如完整模型报告那样专业**。

最可信的内容

- 人类祖先的直立行走——导致→ 骨盆发生结构重组，以支持双足运动 - 人类祖先的直立行走——增加 → 骨盆的宽度并使其变短，以支持双足运动

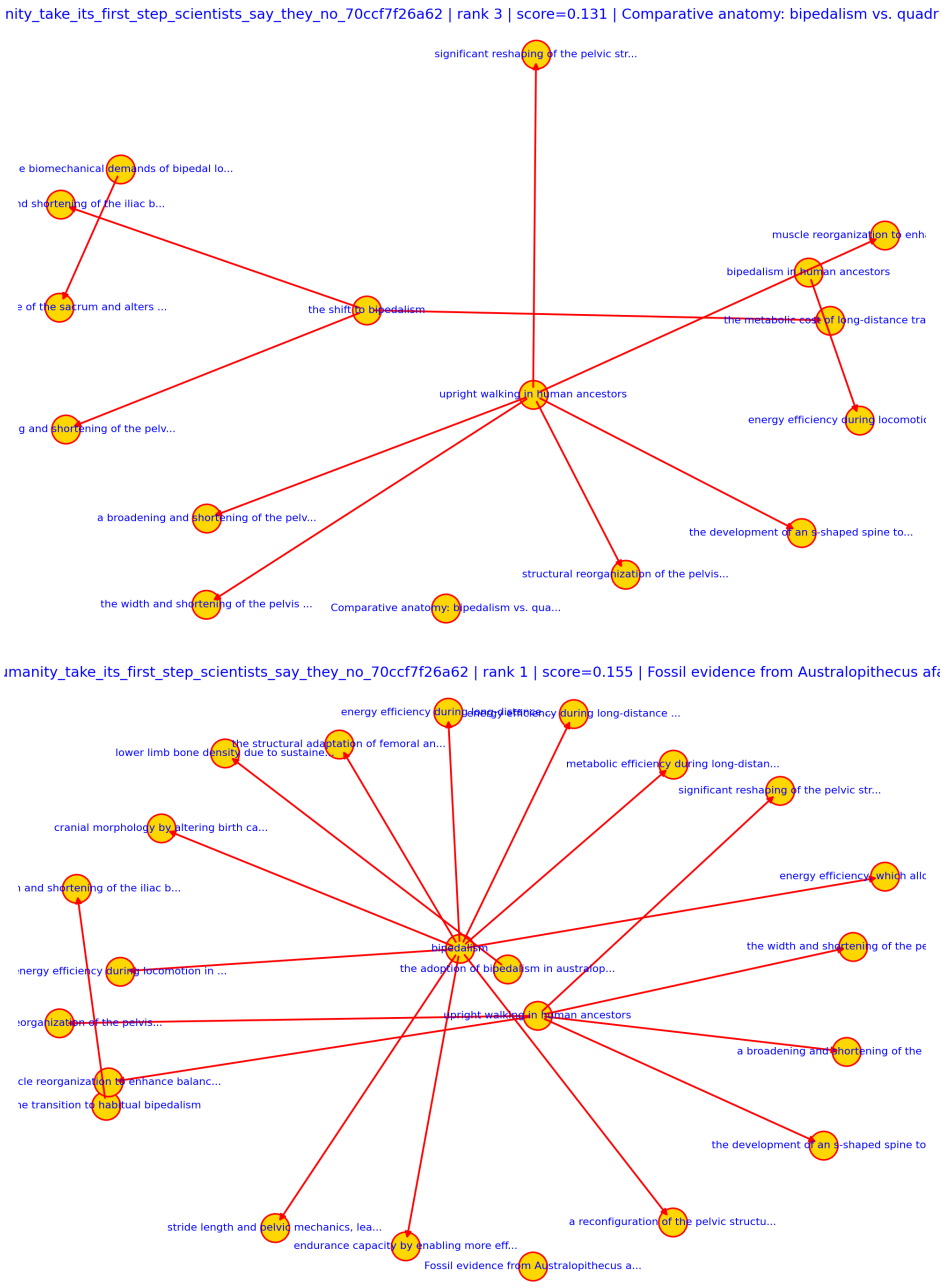


图2：根据《华盛顿邮报》关于700万年前人类祖先双足行走起源的文章构建的两个因果模型。Democritus 从 PDF文档中自动构建数千个此类模型，然后由 Csql 将其转换为因果数据库。

- 人类祖先的直立行走——导致→ 形成S形脊柱，以便在骨盆上方平衡身体

在本文中，我们改为将局部因果模型（如图2所示）转换为因果数据库，以便对其执行类SQL查询。接下来，我们将详细介绍 CSQL 数据模型。

3 CSQL 数据模型

3.1 概述

Csql 将从文档集合中提取的因果知识表示为关系数据库。该数据库并非存储孤立的事实或实体关系，而是编码因果结构 以及汇总自多个局部因果模型的实证支持。该数据模型旨在满足三项要求：（i）可使用标准 SQL 进行查询；（ii）保留源文档中存在的不确定性和分歧；（iii）支持组合式因果查询，例如影响力排序、枢纽检测和多步因果路径。

3.2 核心关系

Csql数据库由以下核心关系组成。

节点。 每个不同的因果概念都表示为一个节点：

node_id 64位标识符 label_canon 规范化概念标签 label_examples 表层形式示例 deg_in 传入因果边数 deg_out 传出因果边数

边。 因果关系表示为加权有向边：

edge_id 唯一边标识符 src_id 源节点 dst_id 目标节点 rel_type 关系类型（例如，导致、影响）polarity 增加/递减/未知 support_lcms 支持性局部模型数 support_docs 支持文档数 score_sum 汇总可信度总量 score_mean 每个模型的平均支持度 score_max 单模型最高得分

边支持。细粒度的来源信息记录在支持表中：

边ID 引用的因果边 文档ID 源文档 LCM实例ID 支持性局部模型 原始分数 模型原始分数 耦合 特定于模型的耦合强度

3.3 派生关系

Csql 还 会从核心表中计算并物化派生关系。这些关系包括 :

因果模块 （强连通分量）。在因果图上计算强连通分量（强连通分量） :

scc_id 组件标识符 n_nodes 节点数 n_edges 边数
support_docs 支持文献数 top_nodes 组件中的高阶节点

3.4 查询语义

Csql 通过标准 SQL 查询实现因果分析。例如：

- **骨干提取：** 按分数总和对边排序，以识别最可信的因果关系。
- **因果枢纽：** 识别出向得分总量较大的节点。
- **下游影响：** 对边进行传递性连接，分析多步因果路径。
- **分歧分析：** 比较 score_max 和 score_mean，以检测脆弱主张。

所有此类查询均使用确定性 SQL 在物化表上执行，而非通过神经推理或基于提示的生成来完成。

4 使用 CsQL查询因果数据库

将文档编译成 Csql数据库后，即可使用标准 SQL 查询进行因果分析。与神经网络或基于提示的系统不同，Csql 查询会在物化关系上确定性地执行，这些关系编码了因果结构、不确定性和来源信息。

本节展示 Csql 支持的若干典型查询模式。所有示例均使用未经修改的 SQL 在 Parquet 支持的表上执行，可在任何现代分析型数据库引擎（例如 DuckDB）中运行。

4.1 骨干提取

一项常见的分析任务是找出文档集合中最可信的因果关系。在 CSQL 中，这相当于按聚合支持度对因果边排序。

```
SELECT
    e.rel_type,n1.label_canon AS
src,n2.label_canon AS
dst,e.support_lcms,e.score_sumFROM
atlas_edgese连接 atlas_nodesn1ONe.src_id =
n1.node_id连接 atlas_nodesn2ONe.dst_id =
n2.node_idORDER BY e.score_sumDESCLIMIT
20;
```

此查询返回一个因果骨干：在许多高分局部因果模型中反复出现的关系。在人类双足行走领域，此查询会揭示诸如双足行走影响代谢效率 以及双足行走提高运动效率这样的关系；这些关系在提取出的论述中始终得到支持。

4.2 因果枢纽

Csql 支持识别因果枢纽：在多个模型中聚合后下游影响较大的变量。

```
SELECT
    n.规范标签 作为 源，求和(e.分
数总和) 作为 输出总量，计数(*) 作
为出度来自 atlas_edgese
```

```
连接 atlas_nodesn条件e.源ID =n.节点 ID按以下
字段分组 n.节点 ID, n.规范标签按以下字段排序
输出总量 降序限制 10;
```

此查询按变量出边的可信度总量进行排名。在实践中，这可以识别某个领域中的主要解释性驱动因素。例如，在双足行走语料库中，双足行走脱颖而出，成为一个核心枢纽，对能量学、骨骼结构和运动产生广泛的下游效应。

4.3 局部机制探索

Csql 可通过筛选特定源概念，帮助用户聚焦探索单个因果变量。

```
SELECT
    e.rel_type,n2.label_canon AS
effect,e.support_lcms,e.score_sumFROM
atlas_edgeseJOIN atlas_nodesn1ONe.src_id =
n1.node_idJOIN atlas_nodesn2ONe.dst_id =
n2.node_idWHERE n1.label_canon =’双足行走’
ORDER BY e.score_sumDESC;
```

此查询按可信度排序，列出给定原因产生的最强下游效应。这类查询有助于深入探究与特定概念相关的机制假设，并可与其他筛选条件组合使用（例如关系类型或极性）。

4.4 溯源与可审计性

Csql 中的每条因果边都会保留溯源信息，将其关联到支持性文档和局部因果模型。

```
SELECT
    s.文档ID, s.LCM实例ID, s.原始分数, s.耦
合来自 atlas_edge_supports其中 s.边ID
 = 10262796833405321330;
```

此查询会准确揭示哪些文档和局部模型参与形成了给定的因果主张。这类溯源查询支持审计、按来源筛选，并可针对不同的聚合策略进行敏感性分析。

4.5 环与反馈结构

Csql 支持通过派生关系（例如强连通分量（SCCs））检测反馈和循环影响。

```
SELECT
    scc_id, 节点数, 边
数, 支持文献数, 顶层
节点来自atlas_scc按
n_nodes 降序排列限制
5;
```

强连通分量可识别紧密耦合的因果模块，这些模块可能代表反馈回路、共同演化的变量或更高层次的机制。这些结构难以在基于有向无环图的系统中表达，却自然地出现在从语言中提取的因果话语中。

5 Csql数据库示例

图3展示了 Csql 如何将从文档集合中提取的因果知识表示为关系数据库，并以人类祖先的直立行走为例。atlas_edges 表中的每一行都对应一条因果关系，这些关系是在多个局部因果模型（LCM）中发现的，并基于源文章所引出的同一话语框架进行聚合。

在此示例中，概念 双足行走 成为一个高度数因果枢纽。该表显示，双足行走对多个下游变量具有显著的出向因果影响，包括长距离运动期间的代谢效率、运动期间的能量效率以及耐力。这些关系并非依据单个句子或模型断言；相反，每条边都汇总了许多独立生成的局部因果模型（LCM）中的证据，这体现在 support_lcms 和 score_sum 等列中。数值越高，表示因果假设之间的冗余度和一致性越强。

关键的是，Csql 通过标准 SQL 展示这一因果结构。例如，查询并选取 score_sum 最大的边，可直接回答诸如“双足行走受到哪些最强的因果影响？”这类问题。同样，按源节点分组并汇总出向得分总量，可以识别下游影响范围大的因果枢纽。这些操作不需要专门的因果语言或图查询引擎，只需使用关系连接和聚合即可。

这一示例凸显了 Csql 的核心设计目标：使从文本中抽取的因果基础知识能够通过熟悉的数据库抽象加以访问。Csql 不会返回抽取三元组的扁平列表或静态知识图谱，而是具体化出一个具有因果意义的模式，在其中可以使用 SQL 对可信度、冗余度、极性和结构角色（例如枢纽、路径、环）进行查询、筛选和组合。以下各节将形式化这一数据模型，并展示如何通过对这些表执行关系查询，自然而然地产生日益复杂的因果推理模式。

这些示例说明了 Csql 如何将文档集合转换为具有因果基础的关系数据库。通过标准 SQL 展示因果结构、不确定性和来源信息，Csql 使分析人员能够借助熟悉的数据库工具开展因果探索、审计和假设生成。

5.1 CsQL数据库的定量属性

除了支持因果 SQL 查询之外，Csql 还构建了一个结构化关系数据库，其图论属性和统计属性可以简洁地加以概括。这些属性对于以下任务十分有用：(i) 诊断文档引起的主题漂移；(ii) 比较不同领域；以及 (iii) 为下游应用选择阈值（例如，筛除低支持度边，或选择“枢纽”概念进行探索）。

模式级摘要。 对于每个文档集合，Csql 会生成两个主要关系：（i）atlas_nodes（规范化概念对象）和（ii）atlas_edges（规范化因果生成器）。边同时携带支持度 （包含该边的局部因果模型数量）和质量（例如 score_sum、score_mean、score_max），这些值是在模型实例上汇总得出的。这为边排序提供了有原则的依据，而无需强加固定本体。

重尾支持度与得分质量。 在各个领域中，我们观察到边支持度和得分质量呈现出一致的重尾分布：少数边占据了总得分质量中不成比例的份额，而大多数边的支持度和质量都较低。这在话语衍生因果图中是意料之中的：少数机制会在话语邻域中被反复陈述（或转述），而许多外围论断则很少出现。在实践中，这意味着可以进行大幅压缩：对于许多分析任务，仅保留按 score_sum 排名靠前的k 条边，就能保留大部分可解释骨干。

枢纽集中度。 Csql 明确指出，影响力集中在少数枢纽周围。定义节点x的出向质量为

$$M_{\text{out}}(x) = \sum_{(x \xrightarrow{r} y) \in E} \text{score_sum}(x, r, y).$$

(a) 骨干边（按得分质量排名靠前）

```
SELECT e.关系类型, n1.规范标签 AS 源节点, n2.规范标签 AS 目标节点,
e.LCM 支持度, e.分数总和
FROM 读取 Parquet('atlas_edges.parquet') e
连接 读取 Parquet('atlas_nodes.parquet') n1 ON e.源ID=n1.节点 ID
连接 读取 Parquet('atlas_nodes.parquet') n2 ON e.目标ID=n2.节点 ID
按 e.分数 排序       _总和 降序
限制 8;
```

```
影响   |双足行走  |代谢效率 ...                                  | 15 | 1.4624
增加 | 双足行走  |能量效率.....                                | 13 | 1.3282
增加 | 双足行走  |能量效率.....                                | 13 | 1.2644
影响   |双足行走  |耐力.....                                     | 13 | 1.2557
增加 | 双足行走  |能量效率 ...                                 | 12 | 1.2093
影响   |双足行走  |结构适应...| 11 | 1.2010
影响   |双足行走  |步幅                              ...   | 11 | 1.1029
影响   |森林覆盖减少  |选择 ...                                    | 6 | 0.4637
```

(b) 因果枢纽（下游质量最大的来源）

```
SELECT n.label_canon AS 源, SUM(e.score_sum) AS out_mass, COUNT(*) AS 出度
FROM 读取 Parquet('atlas_edges.parquet') e
连接 读取 Parquet('atlas_nodes.parquet') n 上 e.源ID=n.节点 ID
按 n.node 分组   _id，n.label   –规范化
按出向质量排序 _降序
限制 8 条;
```

```
双足行走  | 8.8239 | 10
森林覆盖减少 | 0.4637 | 1
环境条件变化.....                                | 0.4026 | 1
股骨转子位置异常 | 0.2853 | 2
地猿化石结构  | 0.1358 | 1
为保持平衡而发生的肌肉重组.....                | 0.1283 | 1
灵长类较大的脑容量  | 0.0814 | 1
股骨的侧向移位 ...                               | 0.0763 | 1
```

(c) 局部机制：来自枢纽的顶级出边

```
SELECT e.关系类型, n2.规范标签 AS dst, e.LCM 支持度, e.分数总和
FROM 读取 Parquet('atlas_edges.parquet') e
连接 读取 Parquet('atlas_nodes.parquet') n1 ON e.源ID=n1.节点 ID
连接 读取 Parquet('atlas_nodes.parquet') n2 ON e.目标ID=n2.节点 ID
WHERE n1.规范标签='双足行走'
ORDER BY e.分数 _sum DESC
LIMIT 7;
```

```
影响  | 代谢效率 ...                          | 15 | 1.4624
增加  |能量效率 ...                         | 13 | 1.3282
增加  |能量效率 ...                         | 13 | 1.2644
影响  | 耐力...                               | 13 | 1.2557
增加  |能量效率...                          | 12 | 1.2093
影响  | 结构适应...                           | 11 | 1.2010
影响  | 步幅                    ...   | 11 | 1.1029
```

图3：**Csql 实战（DuckDB）**。基于 Parquet 的因果图谱上的 SQL 查询示例（atlas_nodes.parquet、atlas_edges.parquet）。（a）按汇总的可信度总量提取骨干因果关系。（b）按出向得分总量识别因果枢纽。（c）深入某个特定枢纽，找出其得分最高的下游影响。对于已编译的图谱表，所有结果均具有确定性。9

```
-- 基线：该对象最常见的出向生成器
    hub "双足行走 "
SELECT
  e.rel_type,
  n2.label _规范化 作为 目标，
  e.LCM 支持度,
  e.分数 _sum
来自 读取 Parquet('atlas_edges.parquet')e
连接 读取 Parquet('atlas_nodes.parquet')n1上e
    .src_id = n1.节点 _id
连接 读取 Parquet('atlas_nodes.parquet')n2位于e
    .dst_id = n2.节点 _id
其中 n1.规范标签 =       '双足行走 '
按 e.分数          _sum 降序
限制 10;
```

```
-- 硬干预（do-cut）：删除所有
    从“bipedalism”发出的边
使用 干预边作为(
  SELECT e.*
  FROM 读取 Parquet('atlas_edges.parquet')e
  连接 读取 Parquet('atlas_nodes.parquet')n1ON
      e.src _id = n1.节点 _id
  WHERE n1.规范标签 <>      '双足行走 '
)
SELECT
  e.rel_type,
  n1.label _规范化 AS src ,
  n2.label _规范化 AS dst ,
  e.support_lcms,
  e.分数 _sum
来自 干预边 e
连接 读取 Parquet('atlas_nodes.parquet')n1位于e
    .src_id = n1.节点 _id
连接 读取 Parquet('atlas_nodes.parquet')n2位于e
    .dst_id = n2.节点 _id
按e.score          _sum 降序
限制 10;
```

图 4：**Csql 将反事实推理作为 SQL 查询重写**。左：基线查询从枢纽“双足行走”中提取最强的出向因果生成器。右：通过将硬干预（*do-cut*）实现为一个视图，移除所有源节点为“双足行走”的边，从而揭示图谱中次强的枢纽和机制。所有操作都是对图谱表执行的确定性转换。

在我们的双足行走案例研究中，头部枢纽（双足行走）的出向质量远远超过其他枢纽。我们使用简单的集中度比率（例如，前 1 和前 5 占总出向质量的比例）来量化这种集中程度。

关系类型构成与极性质量。边被归一化为一组较少的关系类型（例如，导致、影响、增加、减少）和极性类别（增加、递减、未知）。我们报告关系类型和极性质量的分布，将其作为话语框架的粗略指标。例如，有些领域强调影响（较柔和、解释性的话语），而另一些领域则强调导致或带符号的关系（增加/减少）。

环与强连通分量。不同于仅限有向无环图的表示，所生成的因果数据库并不受无环性的约束。我们基于图谱边计算强连通分量（SCCs），以识别反馈回路和闭合的影响模式。在文档尺度上，强连通分量可能较为稀疏；在语料库尺度上，强连通分量模块则成为识别非平凡因果连通性和潜在干预回路的有用信号。在可用时，我们会将强连通分量摘要存储在 atlas_scc 中。

5.2 通过 SQL 视图重写进行反事实查询

Csql 通过将干预视为对图谱边关系的确定性重写，支持干预式推理。在最简单的硬干预中，我们对概念X应用“do-cut”算子，删除所有源节点为X的出向因果生成器。随后，通过比较基准图谱与干预后图谱视图的查询结果差异，来评估反事实查询。这提供了一种轻量但具体的反事实推理形式：我们可以看到，当枢纽的出向影响被移除时，哪些下游影响会消失（或排名下降）。图 4 展示了反事实查询的 Csql 规范。表 2 展示了将其应用于本示例所用的 Csql 数据库所得的结果；该数据库是根据《华盛顿邮报》关于双足行走的文章构建的。

rel	dst（基线顶部）	支持	分数总和_
影响	长距离行进中的代谢效率	15	1.462
增加	早期人族运动过程中的能量效率	13	1.328
增加	长距离运动过程中的能量效率	13	1.264
影响	耐力（散热等）	13	1.256
增加	能量效率（代谢资源）	12	1.209
影响	股骨/胫骨的结构适应	11	1.201
影响	步幅和骨盆力学	11	1.103

反事实（*do-cut*）：这些边会从图谱视图中消失。

表 2：**硬干预的反事实差异**。基线：来自双足行走的最强出向生成因素。在 do-cut(bipedalism) 下，出边按构造被移除，因此这些排名靠前的后果会从查询结果中消失。

CSQL 图谱摘要：WaPo人类起源（双足行走）		
图谱规模		
节点（概念）数量		501
唯一因果边的数量		65
边支持行（LCM证据）		717
枢纽集中度		
头部枢纽		双足行走
头部枢纽出边质量占比		0.748
前五大枢纽出边质量占比		0.857
边权重分布（分数_总和）		
中位数（p_{50}）		0.0337
第90百分位数（p_{90}）		0.847
第99百分位数（p_{99}）		1.377
尾部比率（p_{99}/p_{50}）		40.8
关系类型质量细分		
关系类型	极性	总得分质量
影响	unk	6.19
增加	inc	5.13
导致	unk	0.30
减少	dec	0.14
导致 _TO	unk	0.03
影响	unk	0.01

表3：基于《华盛顿邮报》一篇关于人类双足行走起源的文章构建的 Csql 图谱定量摘要。该图谱呈现出明显的枢纽集中度、重尾边权重，以及以影响型和增加型因果关系为主的特征。所有统计数据均通过 DuckDB 对基于 Parquet 的 Csql 表直接执行 SQL 查询计算得出。

也可以按如下方式在 Csql 中实现“软”干预。

```
WITH soft_doAS(SELECTe.*,CASE WHEN n1.label_canon='双足行走'
THEN0.2*e.score_sumELSEe.score_sumENDASscore_sum_doFROM read_parquet('atlas_edges.parquet')eJOIN
read_parquet('atlas_nodes.parquet')n1ONe.src_id=n1.node_id)SELECT ...
ORDERBYscore_sum_doDESCLIMIT10;
```

5.3 CsQL图谱的定量概述

为补充定性示例，我们报告了基于《华盛顿邮报》一篇关于双足行走起源的文档构建的 Csql 图谱的汇总统计数据。这些统计数据直接通过 DuckDB SQL 查询对图谱 Parquet 表进行计算。统计结果列于表3。

图谱规模。 该图谱包含 501 个规范概念节点和 65 条唯一的规范因果边。此外，边支持表包含 717 行，每行记录一条图谱边在特定文档中由某个局部因果模型（LCM）支持的实例。这种对（i）唯一规范边 和（ii）支持实例 的区分至关重要：图谱表存储提炼后的因果骨干，而边支持表则保留来源信息，以确保可审计性和可复现性。

枢纽集中度。 我们通过汇总每个源概念的输出因果质量（对出边的 score_sum 求和）来衡量枢纽性。在这一领域，头部枢纽是双足行走。值得注意的是，单是头部枢纽就约占图谱总出向得分总量的 74.8%，而前五大枢纽则占 85.7%。这表明，在该文档诱导的话语框架中，因果图谱高度集中：大多数获得强支持的因果关系都从少数几个话语支点向外辐射。

重尾的边强度。 Csql 中的边强度呈现出显著的重尾分布。以 score_sum 作为边权重时，中位数（第50百分位数）边权重为 0.0337，而第90百分位数为 0.847，第99百分位数为 1.377。比值 $p_{99}/p_{50} \approx 40.8$ 量化了这种极端集中现象：少数边主导着因果骨干，而大多数边所承载的权重相对较小。这与 Democritus v2.0 中观察到的经验行为一致：因果话语会产生庞大的假设空间，但只有很小一部分假设得到了冗余支持。

关系类型与极性质量。 我们还根据归一化关系类型（rel_type）和极性（polarity）对图谱边进行汇总。在此图谱中，大多数边属于两个广泛的类别：INFLUENCES（未知极性）和 INCREASES（正极性）。按质量计算，INFLUENCES 贡献了 6.19 个单位的得分质量，而 INCREASES 贡献了 5.13 个单位；CAUSES、REDUCES 以及少数其他关系的贡献则较小。这种分布反映了源文档的语言特征以及因果话语中隐含的不确定性：新闻报道和古人类学主张通常被表述为影响和关联，而不是带有符号且经过量化的效应。

顶级边。 最后，我们列出权重最高的边（按 score_sum 排序）。质量最高的边以双足行走为源，并将其连接到代谢效率、耐力、步幅和运动能量效率等机制性后果。这些边共同构成了归纳图谱的骨架。

要点。 这些定量总结说明了 Csql 图谱为何有用：归纳模式并非只是抽取三元组的扁平列表，而是一个加权因果数据库，具有（i）强大的枢纽结构、（ii）重尾的边强度，以及（iii）按关系类型和极性进行的明确分解。这种结构可以通过标准 SQL 直接查询，从而支持下游分析（例如识别主导驱动因素、追踪多步影响链，或定位高不确定性边），而无需手工设计的本体或固定的领域模式。

总之，Csql 不仅能生成可查询的因果数据库，还能生成一组稳定的定量信号（重尾、枢纽集中度、关系类型混合），从而支持语料库诊断和下游系统设计。

5.4 数据库规模与结构

表4总结了从《华盛顿邮报》一篇关于人类双足行走起源的文章构建的因果数据库规模。尽管该数据库仅源自一份文档，所诱导的话语框架仍扩展到数百个因果变量和数百条有支持的因果关系。

数量	值
节点（概念）数量	501
因果边数量	65
边支持记录数量	717

表4：从单份文档构建的 Csql因果数据库的规模。边支持行对应于支持每条聚合因果边的单个局部因果模型（LCM）。

头部枢纽	前1名质量占比	前5名质量占比
双足行走	0.748	0.857

表5：Csql 图谱中的因果枢纽主导性。影响力最大的单个节点（双足行走）占输出因果质量的近75%，而前五大枢纽合计占比超过85%。

5.5 枢纽主导性与因果集中化

图谱中的因果影响高度集中在少数枢纽周围。我们通过衡量最具影响力的源节点所占的输出因果质量占总量的比例，来量化这种集中程度。

5.6 重尾因果强度

Csql 中的因果边强度呈现明显的重尾分布。我们使用聚合边分数分布的经验分位数来刻画这一特征。

5.7 关系类型构成

因果数据库包含多种关系类型，因果质量在不同谓词之间的分布并不均衡。

6 CSQL数据模型

Csql 将从文档集合中提取的因果基础知识表示为一小组具有明确定义语义的关系表。与传统的信息抽取流程不同，其模式并非预先手工设计，而是通过 Democritus 因果建模流程从语言中自动归纳得出。

从宏观上看，Csql 由四个核心关系组成：

- atlas_nodes：规范化的因果概念，
- atlas_edges：汇总后的因果关系，
- atlas_edge_support：将边与局部因果模型及文档关联起来的来源信息，
- atlas_scc：用于表示反馈结构的强连通分量。

这些表 共同定义了一种具有因果意义的关系模式，支持结构分析和证据感知型查询，并使用标准 SQL。

6.1 节点：规范化因果概念

atlas_nodes 中的每一行都表示从语言中归纳出的一个规范化因果概念，例如 双足行走、运动过程中的能量效率，或长距离行进过程中的代谢效率。节点通过对所有提取出的话语三元组中的主语和宾语短语进行规范化与去重获得。

关键列包括：

- 节点 ID：一个稳定的标识符，
- 规范标签：规范化的概念字符串，

p_{50}	p_{90}	p_{99}	p_{99}/p_{50}
0.034	0.847	1.377	40.8

表 6：因果边得分分布的分位数。极端的p_{99}/p_{50}比率证实，因果假设具有强烈的重尾能量景观。

关系类型	# 边	总质量
影响	19	6.19
增加	27	5.13
导致	9	0.30
减少	7	0.14
导致 _TO	1	0.03
影响	2	0.01

表 7：因果关系类型的分布。影响型关系在数量和因果总量方面都占主导地位，而严格因果谓词则较为少见，也更加保守。

- 标签示例：具有代表性的表层形式，
- deg_in、deg_out：因果图谱中的入度和出度。

节点充当诱导因果结构中的对象。无需外部本体或受控词汇表；对象空间直接从文本中学习。

6.2 边：聚合的因果关系

atlas_edges 中的每一行都表示跨多个局部因果模型（LCM）聚合得到的因果关系。一条边对应一个规范三元组

$$(\texttt{src_id}, \texttt{rel_type}, \texttt{dst_id}),$$

其中，rel_type 的取值范围包括 INCREASES、REDUCES、CAUSES 或 INFLUENCES 等归一化关系。除结构信息外，每条边还携带定量证据：

- support_lcms：该边出现的 LCM 数量，
- support_docs：提供证据的文档数量，
- score_sum、score_mean、score_max：聚合可信度分数，
- 用于区分效应递增、递减和未知的极性质量字段。

这些值编码了不同情况下的一致性：在许多高分假设中反复出现的边会累积更多得分质量，因此具有更高的因果可信度。

6.3 边支持与来源信息

Csql 通过 atlas_edge_support 表保留细粒度的来源信息，该表将每条聚合边关联回：

- 特定的 LCM 实例，
- 源文档，
- 原始逐模型分数和耦合值。

这种设计实现了可审计性和可追溯性。用户可以从高层次的因果主张逐步深入，查看支持该主张的具体模型和文本证据。

6.4 强连通分量

atlas_scc表记录了因果图谱中的强连通分量。这些分量代表反馈回路或紧密耦合的因果子系统，而这些结构无法用基于DAG的因果形式体系表示。

强连通分量为推理循环因果性、相互强化和动力学机制提供了自然的切入点；这些正是我们超越有向无环图、采用表达能力更强的因果表示的关键动因。

7 因果推理即 SQL

Csql 的一项核心贡献是，可以直接将因果推理表示为针对归纳模式的 *SQL* 查询。无需使用专门的图查询语言或因果 DSL。

7.1 识别因果骨干

文档集合中最有力的因果主张对应于聚合得分总量最大的边。例如：

```
SELECT
    e.rel_type, n1.label_canon 作为 src,
n2.label_canon 作为 dst, e.support_lcms,
e.score_sum来自 atlas_edgese连接
atlas_nodesn1其中e.src_id =n1.node_id连接
atlas_nodesn2其中e.dst_id =n2.node_id按以下
字段排序 e.score_sum 降序限制为 50;
```

此查询提取话语框架的因果骨干 ——即在许多高分假设中都存在的关系。

7.2 因果枢纽与下游影响

因果枢纽是输出得分总量较大的节点：

```
选择
    n.label_canon 作为 src, 求和
(e.score_sum) 作为 out_mass来自 atlas_edgese
连接 atlas_nodesn其中e.src_id =n.node_id分组
依据 n.label_canon按以下字段排序 out_mass 降序;
```

在直立行走的例子中，双足行走 成为主要枢纽，对代谢效率、耐力和运动产生强大的下游影响。

7.3 因果组合

多步因果链对应于关系连接：

```
SELECT
    n1.label_canon AS a, e1.rel_type AS r1,
n2.label_canon AS b, e2.rel_type AS r2,
n3.label_canon AS c, (e1.score_sum +
e2.score_sum) AS path_score FROM atlas_edges
e1 连接 atlas_edges e2 ON e1.dst_id =e2.src_id 连
接 atlas_nodes n1 ON e1.src_id =n1.node_id 连接
atlas_nodes n2 ON e1.dst_id =n2.node_id 连接
atlas_nodes n3 ON e2.dst_id =n3.node_id
ORDER BY path_score DESC;
```

此类查询直接在 SQL 中实现因果组合。

数量	值
规范节点（|节点|）	839
规范边（|边|）	450
边支持行（|边_支持|）	1311
文档（|文档|）	10
合并图谱（|图谱|）	7

表8：合并 Csql数据库的语料库级规模统计。节点和边是规范化（字符串归一化）的概念和关系；边支持行以（文档、局部因果模型）实例的粒度存储来源信息。

枢纽源（src）	出向质量	出边	LCM支持
接触草甘膦	105.03	3	24
黑巧克力消费	64.70	5	20
帖子影响的严重程度	44.76	2	17
研究人员的影响	33.68	3	15
草甘膦的施用	30.58	2	9

表9：合并后的 Csql 语料库数据库中出向得分总量最高的因果枢纽 e.

7.4 环与反馈

环可通过自连接或强连通分量表检测。例如，相互影响：

```
SELECT
    n1.规范标签 作为 a, e1.关系类型 作为 r1, n2.规
范标签 作为 b, e2.关系类型 作为 r2来自
atlas_edgese1连接 atlas_edgese2在 e1.源ID =e2.
目标ID AND e1.目标ID =e2.源ID连接
atlas_nodesn1在e1.源ID =n1.节点 ID连接
atlas_nodesn2在e1.目标ID =n2.节点 ID;
```

这使 得可以显式查询基于有向无环图的因果系统中不存在的反馈结构 s.

通过 SQL 视图重写进行反事实查询。Csql 通过将干预视为对图谱边关系的确定性重写，支持干预式推理。在最简单的硬干预中，我们对概念X 应用“do-cut”算子，删除所有源节点为X的出向因果生成器。随后，通过比较基准图谱与干预后图谱视图中的查询结果差异来评估反事实查询。这提供了一种轻量但具体的反事实推理方式：移除某个枢纽的出向影响后，我们可以观察哪些下游影响消失（或排名下降）。

7.5 Csql 语料库数据库的定量概要

表8报告了由多个文档衍生图谱合并构建的 Csql 数据库的语料库级摘要统计信息。该数据库包含规范化概念节点和关系边，以及来源信息（将边与文档及局部模型实例关联起来的边支持行）和聚合可信度统计（例如得分质量和极性质量）。由此得到的分数分布呈重尾特征，且出向得分总量集中在少数因果枢纽上，反映了该系统倾向于在话语流形中显现核心驱动因素。

解读。合并后的 Csql 数据库呈现出两个反复出现的特性。首先，聚合可信度信号（得分质量）呈重尾分布：少数边和来源占据了总质量的很大一部分。其次，这些质量围绕少数几个因果枢纽（表5）组织起来，它们充当话语流形中的核心驱动因素（例如，接触草甘膦和黑巧克力消费）。这些枢纽恰恰是最适用于交互式因果查询的节点：它们有助于

关系_类型	极性	边数	得分质量
影响	unk	135	353.78
导致 _TO	unk	80	188.11
减少	dec	75	181.83
增加	inc	89	156.30
导致	unk	50	89.99
影响	unk	21	75.71

表10：合并语料库中按归一化关系类型和极性划分的边权重分布。

通过 SQL 查询揭示影响重大的下游后果、比较相互竞争的机制，并确定哪些主张在多个局部因果模型中保持稳定。最后，关系权重分布（表10）显示，归纳模式集中为一小套可重复使用的因果谓词（影响/导致_增加/增加/减少），这对于在不同语料库之间实现可移植性至关重要，且无需手工构建本体。

8 数据库 CsQL 的算法构建

本节介绍确定性编译步骤，该步骤将一组局部因果模型（LCM）转换为一个因果 *SQL* 数据库 （Csql）。此阶段的输入是由上游系统（例如 Democritus）生成的 LCM 目录，并可选地附带评分元数据（例如 scores.csv）。输出是一小组 Parquet 表，可直接在 DuckDB（或任何支持 Parquet 的 SQL 引擎）中查询。

8.1 输入约定

我们假设每个文档d 都有一组以 JSON 文件形式存储的局部因果模型（LCM）。一个 LCM 表示为有向标记多重图$M = (V_M, E_M)$，并带有一个焦点字符串和可选的元数据。每条边都是一条记录

$$e = (\texttt{src}, \texttt{rel}, \texttt{dst})$$

其中 src 和 dst 是表层字符串，rel 是自由格式的关系短语（例如“增加”“减少”“影响”）。此外，每个 LCM 可以附带一个标量评分和用于聚合的辅助字段（半径、模型大小、耦合等）。

8.2 CsQL模式

Csql 会具体化三个核心表（另加一个可选的分析表）：

- 节点：从 LCM 节点字符串中发现并规范化的概念（实体/变量）。
- 边：汇总所有局部因果模型并去重后的规范化因果边。
- 边支持：溯源表，用于将每条规范边关联到支持它的文档和 LCM 实例（包括局部模型评分）。
- 强连通分量（可选）：关于诱导图的强连通分量摘要，用于揭示环/模块。

这种分离是有意为之，也体现了标准的数据库规范化实践：edges 为查询提供稳定的事实表，而 edge_support 则保留来源信息，并支持审计、筛选和语料合并。

8.3 规范化与键构建

为了使数据库能够跨文档查询和合并，Csql 会对字符串和关系进行规范化。定义如下：

规范节点标签。 规范化函数 canon(·) 将原始节点字符串映射为规范形式（例如转换为小写、规范化空格和标点、统一破折号等）。规范节点的键构建方式如下：

$$\texttt{node_id} = H(\text{canon}(x))$$

其中 H 是一个稳定的 64 位哈希。

算法1 BuildAtlas：将局部因果模型编译为 Csql数据库**需要：** 包含按文档划分的局部因果模型文件夹的文档运行根目录 R；可选的评分元数据 ata（scores.csv）；最小边筛选器 τ；关系白名单/黑名单。**确保：** Parquet 表：nodes.parquet、edges.parquet、edge_support.parquet（以及可选的 nal **scc.parquet**）。**1：初始化空映射：NodeMap：node_id $\mapsto$节点记录，EdgeAgg：edge_id $\mapsto$聚合记录 2：初始化空列表 SupportRows 3：对于 R 下的所有文档 d，执行 4：为 d 加载一组 LCM JSON 文件 {$\mathbf{M}_i$}（可按分数/大小筛选）5：对于所有局部因果模型 $\mathbf{M}_i$，执行 6：提取元数据：文档ID、LCM实例ID、分数、原始分数、耦合 7：对于所有边 $\mathbf{e} = (\mathbf{u}, \mathbf{r}, \mathbf{v}) \in \mathbf{E}_{M_i}$，执行 8：如果 $|\mathbf{E}_{M_i}| < \tau$则继续 9：结束 if 10：$\mathbf{u}_c \leftarrow$ 规范化(u)；$\mathbf{v}_c \leftarrow$ 规范化(v) 11：源ID $\leftarrow \mathbf{H}(\mathbf{u}_c)$；目标ID $\leftarrow \mathbf{H}(\mathbf{v}_c)$ 12：(关系类型, 极性) $\leftarrow$ 关系类型(r) 13：边ID $\leftarrow \mathbf{H}$(源ID $\|$ 关系类型 $\|$目标ID) 14：使用 u、v 的示例更新 NodeMap 15：更新 EdgeAgg[edge_id]：支持计数、分数质量总和、极性质量 16：向 SupportRows 追加支持行：17：(边ID、文档ID、图谱ID、LCM实例ID、分数、原始分数、耦合) 18：结束 for 19：结束 for 20：结束 for 21：根据 NodeMap 生成节点表，并根据边计算度统计 22：根据 EdgeAgg 生成边表及汇总字段：LCM 支持度、支持文献数、分数总和、平均分数、最高分数、正向极性质量、负向极性质量、未知极性质量、争议 23：根据 SupportRows 生成 edge_support 表 24：（可选）根据边构建诱导有向图；计算强连通分量；写入 scc.parquet 25：将所有表写入 Parquet 26：返回**

规范关系类型。 关系规范化函数 reltype(·) 将原始关系短语映射到一个规模较小的受控词汇表（例如，导致、影响、增加、减少、影响、导致_）。我们还会提取一个粗略的极性 标签 $\in$ {增加,递减, 未知}。

规范边。 规范边由以下内容标识：

$$\texttt{edge_id} = H(\texttt{src_id} \| \texttt{rel_type} \| \texttt{dst_id})$$

这使得在文档内部及不同文档之间都能稳定去重。

8.4 AtlasBuilder：从局部因果模型到 Parquet

算法1描述了 Csql 编译过程。输出表存储：（i）规范对象和箭头（节点、边），（ii）来源信息和评分（边支持），以及（iii）可选的图分析结果（强连通分量）。

聚合与"争议"。每条规范边都会聚合来自多个 LCM实例的证据。令w_i 表示 LCMi 对一条规范边贡献的得分质量（若有调整后得分，则使用调整后得分）。极性质量定义如下：

$$m_{\texttt{inc}} = \sum_{i:\,\texttt{polarity=inc}} w_i, \quad m_{\texttt{dec}} = \sum_{i:\,\texttt{polarity=dec}} w_i, \quad m_{\texttt{unk}} = \sum_{i:\,\texttt{polarity=unk}} w_i.$$

算法2 MergeAtlases：将逐文档 Csql数据库合并为语料库数据库 要求：包含图谱文件夹 {A/} 的目录，每个文件夹均包含 nodes、edges、edge_support。确保：合并后的语料库 Csql数据库：nodes、edges、edge_support。1：加载并串接所有 edge_support 表；可选择在 doc_id 前添加图谱ID，以便区分 2：按 edge_id 对 edge_support 分组，重新计算边聚合指标（支持计数、分数总和、极性质量）3：根据聚合分组构建合并后的边 4：通过收集所有引用的 src_id、dst_id，并根据合并后的边重新计算度，构建合并后的节点 5：将合并后的表写入 Parquet 6：返回

简单的争议指标如下：

$$\text{controversy} = \frac{\min(m_{\text{inc}}, m_{\text{dec}})}{m_{\text{inc}} + m_{\text{dec}} + \epsilon}.$$

当只有一个方向得到支持时，该指标会产生争议值 $= 0$；当增加/减少趋于平衡时，该值则接近 0.5。

8.5 语料库合并：CsQL数据库的并集

由于所有标识符都已规范化（基于哈希），合并多个图谱数据库是一种确定性并集操作，随后进行重新聚合。算法 2 概述了合并过程。
通过这种方式，Csql 使用相同的模式支持文档级 分析和语料库级 分析。至关重要的是，来源信息得以保留：语料库查询可以通过筛选 doc_id、atlas_id 或 lcm_instance_id 进行细化，从聚合后的因果主张追溯到局部模型和源文档。

8.6 给用户的实用说明

Csql 旨在兼容各种现成工具：

- 任何支持 Parquet 的 SQL 引擎都可以查询这些表；我们使用 DuckDB。
- 大多数查询通过连接边和节点来获取易读标签。
- 支持数据溯源的查询会连接边支持，以找回文档和 LCM级证据。

这使 Csql 成为一种因果数据库后端，可部署在可视化层、检索增强生成系统或更高层的推理系统（包括 Democritus 本身）之下。

9 Csql源自检索增强生成编译的因果语料库

Csql 的一个关键设计目标是将因果话语编译与因果数据库构建解耦。前文重点介绍了通过 Democritus 进行大语言模型驱动的话语编译，而 CSQL 流水线并不依赖于因果主张的获取方式。在本小节中，我们展示如何直接基于检索增强生成编译的因果语料库构建 CSQL，而无需访问原始文档或使用基于大语言模型的生成流水线。
我们以因果主张检验（*TCC*）数据集 [Garg 和 Fetzer， 2025] 为典型示例。TCC 使用信息抽取和基于检索的方法，从大型经济学论文语料库中提取因果主张，并生成如下形式的结构化记录

$$(\text{cause}, \text{effect}, \text{sign}, \text{method}, \text{document metadata}).$$

我们不将 TCC 视为静态知识图谱，而是将其解读为预编译因果话语层。随后，Csql 将这一层编译为关系型因果数据库，以支持因果聚合、排序和干预式查询。

话语到数据库的编译。 给定一个 RAG 编译语料库（如 TCC），Csql 会执行确定性编译步骤，将提取的因果主张映射到一小组规范化关系表中。规范化后的原因和结果短语成为节点，而类型化因果关系则成为边。对文档中重复出现的主张进行聚合，生成支持计数和得分质量，类似于通过 Democritus 生成的局部因果模型所获得的结果。
重要的是，Csql 不假设固定的本体、预定义的模式或人工设计的知识图谱。相反，模式直接从因果话语本身中推导而来。溯源信息——例如文档标识符、发表年份和因果推断方法——保存在辅助边支持表中，从而支持下游筛选并提高可审计性。

跨编译方法的统一因果查询。编译完成后，基于检索增强生成语料库构建的 Csql数据库，支持与基于 Democritus 构建的数据库相同类型的因果查询。例如：

- 识别下游影响最大的全局因果枢纽，
- 根据冗余度和汇总支持度对因果主张进行排序，
- 检测方向性证据相互矛盾的争议性主张，
- 通过移除选定来源或对其进行条件化来近似干预。

从查询层的角度看，基于LLM生成的话语构建的因果数据库与基于RAG编译语料库构建的因果数据库并无区别。

对规模与互操作性的启示。 这种分离使 Csql 能够在语料库尺度上运行，而无需对数万份文档进行 LLM推理。TCC 等大型精选资源可以直接导入，使 Csql 能够支持对数量级远超仅依靠生成式流水线所能处理的文档进行因果分析。同时，基于RAG的来源和 Democritus 编译而成的数据库可以无缝合并，形成一个涵盖异构文档集合的统一因果数据库。

从抽象层面看，Csql 可作为各种因果话语生成器（包括大语言模型、RAG系统和传统信息抽取流程）的通用编译目标，同时提供统一的、原生支持 SQL 的因果分析接口。

9.1 CSQL 基于 RAG 编译因果语料库的 Csql：因果主张检验（TCC）

我们还在因果主张检验 （TCC）数据集上实例化了 Csql。该数据集已提供从 ~45K 篇经济学论文中抽取的语料库级有向因果陈述。因果主张检验（TCC）项目 [Garg 和 Fetzer， 2025]从大型经济学论文语料库中提取并编目因果主张，生成了一个因果陈述全球索引，并附有符号、统计显著性和推断方法等元数据。在此场景下，Csql 不会运行 Democritus 话语编译器；而是将 TCC 视为预编译的话语层，并将其编译为相同的关系模式（节点、边、边支持）。

生成的数据库包含 295,459 个规范化概念字符串和 260,777 条不同的有向边（265,656 条边支持行），涵盖 44 个发表年份。由于当前的数据导入采用通用关系标签 “ 影响 ”，且尚未将符号／方法元数据投影到 Csql 中，边权重在很大程度上反映频次（支持论文数量），因此形成了近乎退化的重尾分布（例如，支持质量中位数为 1，第99百分位数为 2）。这种表现是意料之中的：TCC 本身已是边列表，而 Democritus 原生图谱会聚合许多相互重叠的局部因果假设，因此形成更丰富的关系词汇和长尾得分质量。尽管存在这一限制，Csql 接口已支持有用的语料库规模查询，例如枢纽发现和骨干提取；在进行中的工作中，我们正将 TCC 的符号和识别方法元数据投影到 Csql 中，以支持极性识别查询和方法条件化分歧分析。

为了在不对数万份 PDF 运行 Democritus 的情况下对 Csql 的规模进行压力测试，我们将因果主张检验 (TCC)数据集视为一个已经编译好的因果话语层。TCC 提供论文级因果边（原因 → 效应）以及年份和所述因果推断方法等元数据。我们通过将节点字符串（原因/效应）规范化为概念标识符、汇总不同论文中的相同边，并将逐篇论文证据行存储在 edge_support 表中，将该语料库编译为 Csql数据库。以下所有统计数据均直接通过 DuckDB 查询生成，查询对象是由此得到的 Parquet 表。

数量	值
主张实例（边支持行）	265,656
唯一论文（文档ID）	45,319
唯一年份	44
不同的方法字符串	1,575
规范概念节点（节点）	295,459
唯一因果边（边）	260,777

表11：TCC–Csql数据库的摘要统计。该语料库被视为一个由检索增强生成/信息抽取编译的因果话语层；Csql 将其编译为可查询的因果数据库，并进行边聚合并提供逐篇论文证据。

枢纽（原因）	出向质量	出边数量
教育	442	374
贸易自由化	391	370
age	358	339
通货膨胀	280	250
货币政策	270	211

表12：TCC–Csql 中排名靠前的语料库枢纽，按出边支持度总量（出边上 support_docs 的总和）排序。这些枢纽概括了经济学语料库中占主导地位的因果话语中心。

规模。 编译后的 TCC–Csql数据库包含**265,656** 条证据行（每个抽取的主张实例对应一条），涵盖**45,319** 篇不同论文、**44** 个发表年份，以及**1,575** 个不同的方法字符串（按语料库中的报告统计）。在模式层面，数据库包含**295,459** 个规范概念节点和**260,777** 条唯一有向因果边。这表明，Csql 无需手工设计本体，即可摄取由检索增强生成/信息抽取编译的大型因果语料库，并将其作为统一、可查询的因果数据库呈现。

枢纽与骨干主张。语料库规模的因果话语的一个显著特征是，枢纽变量会涌现出来，并具有较大的出向质量。在 TCC–Csql 中，出向质量最大的枢纽包括教育、贸易自由化、年龄、通货膨胀和货币政策，每个枢纽都参与数百条不同的出边。例如，对于“什么对通货膨胀的影响最大？”这一查询，返回的反复出现的主张包括货币政策 →通货膨胀、货币增长 → 通货膨胀和产出缺口 → 通货膨胀，并按论文支持度排序。这些结果并不被解读为真实因果关系；相反，它们将语料库操作化为一个可供搜索、筛选和审查的因果主张空间。

时间动态。 由于 edge_support 中的每个主张实例都带有发表年份，Csql 支持时间切片因果分析。按年份汇总主张实例，可以看到其数量随时间显著增长：语料库在 20 世纪 80 年代初每年包含不到 2,000 个抽取的主张实例，到 2010 年代末增至每年 10,000 多个，2020 年则超过 18,000 个。对于某个特定的规范因果主张（例如，货币政策 → 通货膨胀），Csql 可以按年份返回支持性论文的分布，从而纵向追踪特定因果主张在数十年间如何出现和再度出现。

方法字符串与异质性。 TCC 元数据包含一个报告因果推断方法的字段。在当前语料库快照中，该方法字段包含大量异质性较高的字符串（例如，工具变量法、双重差分法、双向固定效应，但也有许多更细分或自由文本形式的变体）。不同方法取值数量众多（1,575）也体现了这一点。Csql 将方法列视为观测属性：用户可以按原样筛选方法或分组，也可以选择将方法字符串规范化为较小规模的受控分类体系，以便进行下游分析。

10 相关工作

本文处于以下领域的交叉点：（i）从文本中进行因果关系抽取；（ii）因果知识库构建；（iii）以数据库和图为中心的知识表示；以及（iv）因果发现与推断。我们将 Csql 置于各领域既有工作的背景下，重点说明其与现有知识图谱、检索增强生成（RAG）系统和因果抽取流水线的不同之处。

影响通货膨胀的主张	支持（论文）	Mass
货币政策 → 通货膨胀	15	15
货币增长 → 通货膨胀	12	12
产出缺口 →通货膨胀	11	11
货币政策冲击 →通货膨胀	8	8
财政政策 → 通货膨胀	7	7

表 13：示例查询结果：以论文支持度排名的、指向通货膨胀的语料库主张。在 Csql 中，此类查询可直接通过对边和 edge_support 执行 SQL 连接与筛选来表达。

10.1 文本因果关系抽取

关于识别自然语言中的因果关系，已有大量研究，从早期使用提示短语（例如“因为”“导致”）的基于模式的方法，到如今训练用于对文本片段之间的因果关系进行分类的神经模型和基于 Transformer 的模型，研究脉络十分丰富 [Radinsky et al., 2012, Yang et al.,2022, He et al., 2025]。这些系统通常在单个句子或较短上下文的层面运行，并输出成对标注关系。

Csql 在范围和目标上有着根本不同。我们不将抽取出的关系视为最终结果，而是将其视为嘈杂的语篇样本，并据此构建、评估和汇总多种相互竞争的因果假设。输出并非抽取三元组的扁平列表，而是一个关系数据库，其中编码了整个文档集合中的因果结构、证据质量和模型一致性。

10.2 因果知识库与图构建

有几种系统致力于从大型文本语料库中构建因果知识库或因果图 [Hassan- zadeh 等， 2020]。这些资源将提取出的因果元组汇总到图结构存储中，通常采用领域本体或人工设计的模式。

Csql 在三个关键方面有所不同。首先，它的模式是从语言中自动归纳得出的，而非预先固定。其次，因果边并不被视为原子事实，而是被视为由许多局部因果模型支持的集合。第三，最终表示采用关系型而非纯图结构，因此可以使用标准 SQL 查询来表达因果分析。

10.3 知识图谱、本体与 GraphRAG

知识图谱（KG）和本体驱动表示为存储结构化断言并支持演绎推理提供了成熟的基础 [Hogan 等，2021，Group，2012，Knublauch 和 Kontokostas， 2017]。近来，GraphRAG 风格系统将图结构数据与检索增强生成流程相结合 [Lewis 等， 2020]。

这些方法主要呈现文本中所断言的内容。它们缺乏对干预、复合、反馈和下游影响等因果操作的内在语义。相比之下，Csql 表示因果假设，并同时记录这些假设在相互竞争的模型中获得的支持程度。Csql 中的因果推理通过使用 SQL 聚合和复合关系来完成，而不是检索或排列图中的节点。

10.4 因果主张检验与语料库规模的因果数据库

因果主张检验（TCC）项目 [Garg 和 Fetzer， 2025]从大型经济学论文语料库中提取并编目因果主张，生成一个因果陈述全球索引，并附有符号、统计显著性和推断方法等元数据。

Csql 在功能上具有互补性，但在表示方式和工作流程上有所不同。Csql 并非生成一份涵盖整个语料库的统一主张目录，而是针对每篇文档构建多个局部因果模型 ，对其进行评估，并将其结构汇总为可查询的因果数据库。从概念上看，TCC 注重某一学科内的广度与标准化，而 Csql 则注重文档内部及文档之间的深度和基于模型的一致性。

领域	节点	边	头部枢纽	最高分数
人类起源（《华盛顿邮报》）	412	683	双足行走	1.46
巧克力与衰老（《华盛顿邮报》）	527	912	黑巧克力摄入量	1.12
草甘膦（《纽约时报》）	489	801	农达使用	2.04
南极洲（《纽约时报》）	366	594	冰川退缩	1.87

表 14：根据不同文档领域生成的 Csql数据库的汇总统计数据。

10.5 用大语言模型进行因果发现

越来越多的研究探讨大语言模型（大语言模型）在因果发现和推理中的应用 [Shen 等人, 2023年, Le 等人, 2024, Kosoy 等人, 2023]。这些方法通常依赖大语言模型直接提出因果方向或图结构。

Csql 采用了不同的设计。我们将大语言模型视为假设生成器，用自然语言提出候选因果陈述。此后的模型构建、评估、聚合和查询均以确定性方式进行。这种分离使因果分析能够独立于语言模型进行审计、复现和执行。

10.6 因果发现与统计推断

经典因果发现方法以数值数据为基础，使用基于分数、基于约束或混合搜索，在有向无环图或等价类上进行搜索 [Spirtes 等人，2000年，Chickering，2002年，Zheng 等人， 2018]。虽然这些方法无法直接应用于以文档为中心的场景，但原则上 Csql 与它们兼容。Csql 生成的因果数据库可作为假设生成器、先验结构或解释框架，用于下游统计因果分析。

总结。 与先前工作相比，Csql 有三项显著贡献：(i) 将非结构化文档编译为关系型因果数据库，(ii) 将因果主张表示为对相互竞争的局部模型进行聚合的结果，而非孤立的提取结果，以及 (iii) 支持使用标准 SQL 作为查询语言进行因果推理。

11 CSQL 与其他领域

我们已在多个不同领域对 Csql 进行了试验（见表 1）。下面将介绍所分析的领域。

1.《华盛顿邮报》最近刊登了一篇报纸文章，探讨黑巧克力与衰老之间可能存在的因果联系。 [4]。本文始终以这个领域作为贯穿全文的示例，说明 Democritus 能够执行何种因果分析。2. 该因果分析基于《华盛顿邮报》的一篇文章，内容涉及人类祖先双足行走的起源，可追溯至700万年前。[5]。3.《纽约时报》最近刊登了一篇文章，讨论除草剂农达（化学名称为草甘膦）引发的争议 [6]。4.《纽约时报》刊登了一篇关于南极洲冰川融化的文章 [7]。高分模型（上方）包含一个密集的边集群，这些边在多个话语邻域中反复得到支持，其中包括冰融化导致淡水流入、海洋环流变化以及全球海平面上升。这些相互重叠的机制带来了较强的模型一致性和较高的评估分数。相比之下，低分模型（下方）包含的、得到充分支持的边较少，语义连贯性也较弱。尽管左伴随函子可以自由生成此类模型，但它们

[4]https://www.washingtonpost.com/wellness/2025年12月26日/dark-chocolate-health-benefits[5] urlhttps://www.washingtonpost.com/science/2026年1月2日/human-ancestor-biped/[6] https://www.nytimes.com/2026年1月2日/climate/草甘膦-农达-retracted-study.html?smid=nytcore-ios-share.[7]https://www.nytimes.com/2025年12月27日/climate/南极洲-thwaites-glacier.html

在评估过程中无法积累足够的证据，因此在伴随重构之前会被抑制。

12 讨论与启示

12.1 与知识图谱和检索增强生成的关系

Csql 与知识图谱和 GraphRAG 风格系统有着根本区别。知识图谱存储已断言的事实；Csql 存储经过评估的因果假设。GraphRAG 检索相关节点；Csql 则通过对相互竞争的因果模型进行聚合来计算可信度。

知识图谱强调陈述了什么，而 Csql 强调通过跨论述的一致性在结构上得到支持的内容 。

12.2 与因果推断的关系

Csql 并不能取代统计因果推断，而是通过处理数值数据不可用、但因果话语丰富的情形来对其加以补充。Csql 的输出可用作假设生成器、先验构造器，或下游因果分析的解释框架。

12.3 CSQL 作为因果编译器

从抽象层面来看，Csql 是一个编译器：

$$\text{Documents} \longrightarrow \text{Causal Models} \longrightarrow \text{Relational Database}.$$

编译目标是一个具有因果意义的 SQL 模式，并配备了严谨的查询代数。这一视角将 Csql 与函子数据库和数据迁移方面的研究联系起来，不过，用户无需了解这些机制，也能有效地使用该系统。

13 局限性与未来工作

尽管 Csql 展示了如何将大型文档集合编译为可通过 SQL 查询的因果数据库，但当前系统仍存在若干重要局限。这些局限也指出了未来研究和系统开发的明确方向。

13.1 局限性

对话语质量的依赖。Csql 依赖大语言模型（LLM）充当话语编译器，生成候选因果陈述。尽管所有下游处理都是确定性的，最终因果数据库的质量和主题连贯性仍取决于提取出的话语质量。嘈杂或偏离领域的话语扩展可能导致因果图谱分布零散、枢纽集中度较低且分数分布更平坦。评分和聚合层旨在抑制缺乏依据的主张，但话语质量仍是主要瓶颈。

缺乏因果真值验证。Csql 所表示的因果关系是从语言中推断得出的，并未通过实验数据或观察数据进行验证。因此，应将 Csql 理解为捕捉了话语中的因果结构——即文本中所声称、暗示或论证的内容——而非经过验证的因果真值。这一区别至关重要：Csql 评估的是话语一致性下的可信度，而非统计因果推断意义上的因果可识别性。

静态因果快照。当前系统从静态文档集合构建因果数据库。因果话语的时间演变——主张如何随时间变化、证据如何累积或撤回，以及争议如何演变——尚未得到明确建模。每个 Csql数据库都代表处理文档时的话语快照。

复合推理能力有限。尽管 Csql 支持通过 SQL 连接执行多步因果查询（例如两跳路径和枢纽分析），但它尚未实现明确的因果代数运算，例如干预、反事实推理或复合图重写。这些运算隐含于底层理论中，但尚未在数据库查询层公开提供。

话语编译的可扩展性。 流水线中计算成本最高的阶段是通过大语言模型进行话语编译。虽然所有后续步骤都能高效扩展，并在 Parquet 和 SQL 引擎上运行，但要在数百万份文档上进行大规模部署，仍需要额外的优化，例如缓存、增量更新或混合检索–生成策略。

13.2 未来工作

时间化和版本化因果数据库。 Csql 的一个自然扩展是支持按时间索引的因果数据库，以跟踪因果主张如何随着文档版本、发布日期或证据更新而演变。这将支持诸如“哪些因果主张随着时间推移获得了或失去了支持？”之类的查询，对于科学、政策和监管分析尤其有价值。

干预式和反事实查询层。 Csql 的未来版本将在 SQL 之上提供更高层次的因果运算符，包括用于干预、删除和假设性修改因果边的抽象。这些操作可以建立在底层因果语义之上，同时仍可通过熟悉的查询接口访问。

与数值数据集成。 尽管 Csql 目前处理的是从文本中提取的因果主张，但在有数值数据集可用时，自然可以扩展以纳入这些数据。例如，因果数据库中的边可以关联统计模型、回归结果或实验性证据，从而实现结合论述与数据的因果分析。

由用户引导的模式优化。 虽然 Csql 有意避免采用人工设计的本体，但未来的工作可以探索轻量级的人机协作机制，用于优化概念规范化、合并等价节点，或突出与特定领域相关的因果变量。此类指导有助于提升可解释性，同时不牺牲通用性。

分布式与增量式编译。 为支持大规模部署，未来的实现将探索增量式图谱构建、分布式论述编译，以及对 Parquet 支持的因果表进行流式更新。这样一来，随着新文档不断到来，Csql数据库便可持续维护。

因果 Csql 作为标准接口。 最后，我们设想 Csql 可以成为面向文本语料库进行因果查询的候选抽象层，类似于 SQL 如何为关系数据访问制定标准。长远目标是定义一种可移植的因果查询接口，可嵌入分析平台、仪表板和决策支持系统，使用户无需直接接触因果建模形式体系。

参考文献

Anthropic。Claude 3 模型说明卡。https://assets.anthropic.com/m/61e7d27f8c8f5919/original/Claude-3-Model-Card.pdf，2024。访问日期：2025-12-28。

David Maxwell Chickering。通过贪心搜索进行最优结构识别。机器学习研究期刊，3：507–554，2002年。

Abhimanyu Dubey 等人。Llama 3 模型群。*arXiv* 预印本 *arXiv:2407.21783*，2024。

Brendan Fong、David I. Spivak 和 Rémy Tuyéras。作为函子的反向传播：监督学习的组合视角。载于 第*34*届 *ACM* 年会/*IEEE* 计算机科学逻辑研讨会，*LICS 2019*，温哥华，不列颠哥伦比亚省，加拿大，*2019*年*6*月*24*日至*27*日，第1–13页。IEEE，2019。doi：10.1109/LICS.2019.8785665。URL https://doi.org/10.1109/LICS.2019.8785665。

Tobias Fritz. 马尔可夫核、条件独立性与充分统计量定理的综合方法。《数学进展》, 370:107239, 2020年8月。ISSN 0001-8708。doi: 10.1016/j.aim.2020.107239。URL http://dx.doi.org/10.1016/j.aim.2020.107239。

Prashant Garg 和 Thiemo Fetzer。经济学中的因果主张检验。*arXiv* 预印本 *arXiv:2501.06873*，2025年。URL https://arxiv.org/abs/2501.06873。基于经济学论文中抽取的因果主张的数据集与分析。Bruno Gavranovi´c、Paul Lessard、Andrew Dudzik、Tamara von Glehn、João G. M. Araújo 和 Petar Veli˘ckovi´c。立场：范畴深度学习是所有架构的代数理论，2024。URL https://arxiv.org/abs/2402.15332。W3C OWL 工作组。*OWL 2 Web* 本体语言文档概述。W3C 推荐标准，2012年。Oktie Hassanzadeh、Debarun Bhattacharjya、Mark Feblowitz、Michael Perrone、Shirin Sohrabi、Kavitha Srinivas 和 Michael Katz。通过大规模文本挖掘抽取因果知识。载于*AAAI* 人工智能大会论文集，第34卷，第13520–13527页，2020年。Xiaomei He 等人。事件因果识别综述：分类体系、资源与挑战。*ACM* 计算调查，2025年。预印本；另见 Tan 等人的 CASE 2022 事件因果识别共享任务。Aidan Hogan、Eva Blomqvist、Michael Cochez 等人。知识图谱。*ACM* 计算调查，54(4)，2021年。Guido W. Imbens 和 Donald B. Rubin。《统计学、社会科学与生物医学科学的因果推断：导论》。剑桥大学出版社，美国，2015年。ISBN 0521885884。Holger Knublauch 和 Dimitris Kontokostas。形状约束语言（SHACL）。*W3C* 推荐标准，2017年。Vladislav Kosoy 等人。大语言模型是否具备因果知识? *arXiv* 预印本 *arXiv:2305.00000*，2023年。Tuan Le 等人。使用大语言模型进行多智能体因果发现。*arXiv* 预印本 *arXiv:2404.12345*，2024。Patrick Lewis、Ethan Perez、Aleksandra Piktus 等人。面向知识密集型自然语言处理任务的检索增强生成。神经信息处理系统进展，2020年。Sridhar Mahadevan。GAIA：生成式人工智能的范畴论基础。*arXiv*，2024。URL https://arxiv. org/abs/2402.18732。Sridhar Mahadevan。托扑因果模型中的直觉主义*j*-do 演算，2025a。URL https://arxiv.org/ abs/2510.17944。Sridhar Mahadevan。由大语言模型构建大型因果模型，2025b。URL https://arxiv.org/ abs/2512.07796。Mistral AI。Mistral Large 2 (24.07)。https://mistral.ai/en/news/mistral-large-2407，2024。访问日期：2025-12-28。OpenAI。GPT-4o 系统卡。https://openai.com/index/gpt-4o-system-card/，2024。访问日期：2025-12-28。OpenAI。ChatGPT — 发行说明。https://help.openai.com/en/articles/6825453，2025。访问日期：2025-12-28。Judea Pearl。《因果关系：模型、推理与推断》。剑桥大学出版社，美国，第2版，2009年。ISBN 052189560X。Kira Radinsky、Sagie Davidovich 和 Shaul Markovitch。学习因果关系以预测新闻事件。载于第*21*届国际万维网大会（*WWW*）论文集，第909–918页，2012年。doi: 10.1145/2187836.2187958。Xinyu Shen 等人。用于因果发现与推断的大语言模型：综述。*arXiv* 预印本 *arXiv:2309.12345*，2023年。Peter Spirtes、Clark Glymour 和 Richard Scheines。《因果、预测与搜索》。麻省理工学院出版社，第2版，2000年。Jie Yang、Soyeon Caren Han 和 Josiah Poon。从自然语言文本中抽取因果关系的综述。知识与信息系统，64(5):1161–1186，2022年。doi: 10.1007/s10115-022-01665-w。Xun Zheng、Bryon Aragam、Pradeep Ravikumar 和 Eric Xing。无环图无泪：用于结构学习的连续优化。神经信息处理系统进展，2018年。