arXiv:2303.14617v1 [cs.DB] 26 Mar 2023

神经图推理：复杂逻辑查询回答与图数据库的结合

Hongyu Ren* [1] *hyren@cs.stanford.edu*

Mikhail Galkin* [2] *mikhail.galkin@intel.com*

Michael Cochez[3] *m.cochez@vu.nl*

Zhaocheng Zhu[4 5] *zhaocheng.zhu@umontreal.ca*

Jure Leskovec[1] *jure@cs.stanford.edu*

* **同等贡献**[1]

斯坦福大学 英特尔人工智能实验室 [2] [3]阿姆斯特丹自由大学和爱思唯尔发现实验室，荷兰阿姆斯特丹 [4]*Mila*—魁北克人工智能研究所 [5]蒙特利尔大学

摘要

复杂逻辑查询回答（CLQA）是图机器学习领域新近出现的一项任务，它超越了简单的单跳链接预测，旨在潜在空间中对海量且可能不完整的图进行更为复杂的多跳逻辑推理。该任务在学术界引起了广泛关注；众多工作从理论和实践两个方面拓展了这一领域，以高效系统处理不同类型的复杂查询和图模态。本文全面综述了 CLQA，并构建了详细的分类体系，从多个角度考察该领域，包括图类型（模态、推理域、背景语义）、建模方面（编码器、处理器、解码器）、支持的查询（算子、模式、投影变量）、数据集、评估指标和应用。

在进一步完善 CLQA 任务的基础上，我们引入神经图数据库（NGDB）的概念。神经图数据库扩展了图数据库（图数据库）的理念，由神经图存储和神经图引擎组成。在神经图存储中，我们设计了图存储和特征存储，并通过编码器将信息进一步嵌入潜在嵌入存储中。给定查询后，神经查询引擎会学习如何进行查询规划和执行，并通过与神经图存储交互，高效检索正确结果。与传统图数据库相比，神经图数据库能够利用嵌入存储，对不同模态的特征进行灵活且统一的建模。此外，当图不完整时，它们可以稳健地检索答案，而普通图数据库则无法恢复这些答案。最后，我们指出了神经图数据库未来研究中值得探索的方向、尚未解决的问题及其应用。

目录

1 引言 **3**

2 预备知识 **4**

2.1 图的类型. 5

2.2 基本图查询回答.72.3 基本近似图查询回答.82.4 图查询回答.82.5 近似图查询回答.122.6 三角范数与余范数.122.7 图表示学习.13

3 NeuralGraph 数据库 **13**

3.1 神经图存储.163.2 神经查询引擎.17

4 图 21

4.1 模态.214.2 推理域.214.3 背景语义.22

5 建模 **25**

5.1 编码器.265.2 处理器.285.3 解码器.355.4 计算复杂度.36

6 查询37

6.1 查询算子.376.2 查询模式.436.3 投影变量.45

7 数据集和 M 指标47

7.1 评估设置.477.2 查询类型.477.3 训练.487.4 推理.497.5 指标.51

8 应用 **52**

9 总结与未来机遇 53

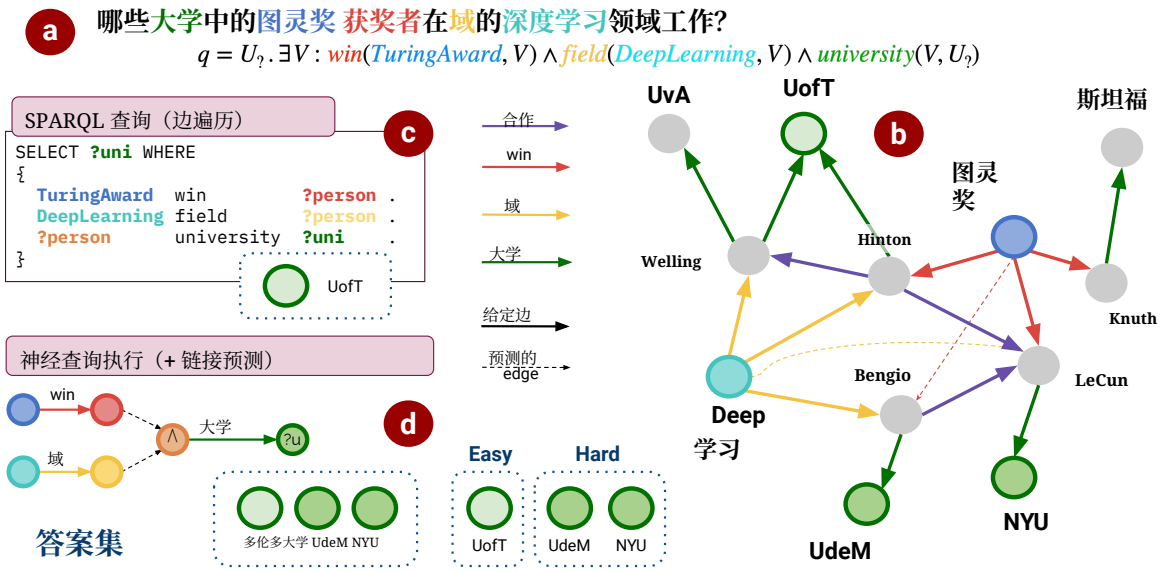


图 1：复杂逻辑查询 及其在不完整图上的执行过程 。SPARQL 等符号引擎 执行边遍历，并直接检索图中可达的不完整易 答案 集，即 {UofT}。神经查询执行 恢复缺失的真实边（虚线），并返回一组额外的难 答案 {UdeM、 NYU这些答案无法通过符号法获得。

1 引言

图数据库是捕获、组织和浏览现实世界实体间结构化关系信息的关键架构。与将信息存储在具有严格模式的表中的传统关系型数据库不同，图数据库以异构图的形式存储信息，其中节点表示实体，边表示实体之间的关系。在图数据库中，关系（即实体之间的异构连接）是核心要素。凭借图结构和更灵活的模式，图数据库能够以更高效、更具表现力的方式处理相距较远实体之间的高阶关系，尤其适用于遍历多跳层级结构。传统数据库需要执行代价高昂的连接操作来检索信息，而图数据库则可直接遍历图，并借助邻接矩阵更高效地沿链接进行导航。凭借这些能力，图数据库成为许多关键工业应用的基础，包括虚拟助手中的问答（Flint，2021年；Ilyas 等人，2022年）、市场平台中的推荐系统（Dong，2018年；Hamad 等人，2018年）、移动应用中的社交网络（Bronson 等人，2013年），以及金融行业中的欺诈检测（Tian 等人，2019年；Pourhabibi 等人，2020年）。

对于给定的下游任务，图数据库最重要的任务之一是执行复杂查询回答。其目标是从图数据库中检索给定输入查询的答案。给定查询后，图数据库首先通过查询规划器将查询转换并优化为更高效的图遍历模式，然后使用查询执行器在图数据库上执行该模式，从图存储中检索答案。存储会将图压缩为适合快速表查找的符号索引。因此，在完整性假设下，即存储的图中不存在缺失边时，查询能够快速且高效地完成。

然而，众所周知，大多数现实世界的图都不完整。例如，在 Freebase 中，93.8% 的人没有出生地，78.5% 的人没有国籍（Mintz 等人，2009 年），约 68% 的人没有任何职业（West 等人，2014 年）；而在维基数据中，约 50% 的艺术家没有出生日期（Zhang 等人，2022a），已知建筑物中只有 0.4% 有高度信息（Ho 等人，2022年）。在数据不完整的情况下，天真地遍历图会导致大量相关结果遗漏，而且随着跳数增加，这一问题会进一步加剧。这从根本上阻碍了图数据库的应用程序。

链接预测是一项具有挑战性的任务。以往的研究通过学习每个链接的潜在表示来预测链接（Bordes 等人，2013年；Yang 等，2015年；Trouillon 等人，2016；Sun 等人，2019年），或通过挖掘规则来预测（Galárraga 等人，2013年；Xiong 等人，2017年；Lin 等人，2018年；Qu 等人，2021年）。然而，这始终是在可能不完整的结果与可判定性之间进行权衡：图越稠密，某些 SPARQL 蕴涵制（Hawke 等人，2013年）就越无法保证查询执行能在有限时间内终止。

另一方面，图机器学习的最新进展使人们能够在潜在空间中对大型图进行富有表现力的推理，而不受可判定性瓶颈的制约。由Hamilton 等人(2018年)开展的开创性工作——图查询嵌入（GQE）——为在不完整知识图谱上求答复杂的、类似数据库的逻辑查询奠定了基础。在查询执行期间，通过使用可学习向量表示和神经网络对实体、关系及逻辑运算符进行参数化，可以推断缺失链接。对于（图1）中的不完整知识图谱，给定复杂查询“深度学习领域的图灵奖获奖者在哪些大学工作？”，传统符号图数据库（类似 SPARQL 或 Cypher）只能返回一个答案（`UofT`），该答案可通过边遍历获得。相比之下，神经查询嵌入使用嵌入空间中的可学习向量对图和查询进行参数化。神经查询执行类似于在嵌入空间中遍历图并执行逻辑运算符，从而推断缺失链接，并在答案集中补充两个相关答案 `UdeM` 和 `NYU`，而符号数据库无法获得这些答案。

自那以后，该领域受到越来越多的关注，支持的逻辑运算符、查询类型、图模态和建模方法也不断改进。在我们看来，这些改进相当零散，缺乏总体目标。目前仍缺少一个统一的框架来整理现有工作并指导未来研究。为此，我们提出开展该领域首批全面研究之一。我们设计了一套分类体系，从三个主要维度对现有工作进行分类：即，**(i) 图**(第4节——底层图及其模式背后的逻辑形式体系) **(ii) 建模**(第 5 节——用于回答查询的神经方法) **(iii) 查询**(第 6 节——可回答哪些查询)。随后，我们讨论**数据集与指标**(第 7 节——如何衡量 NGDB 引擎的性能)。这些维度中的每一个都进一步细分为更精细的方面。最后，我们列出 CLQA 应用（第 8 节），并总结未来研究面临的开放挑战（第 9 节）。

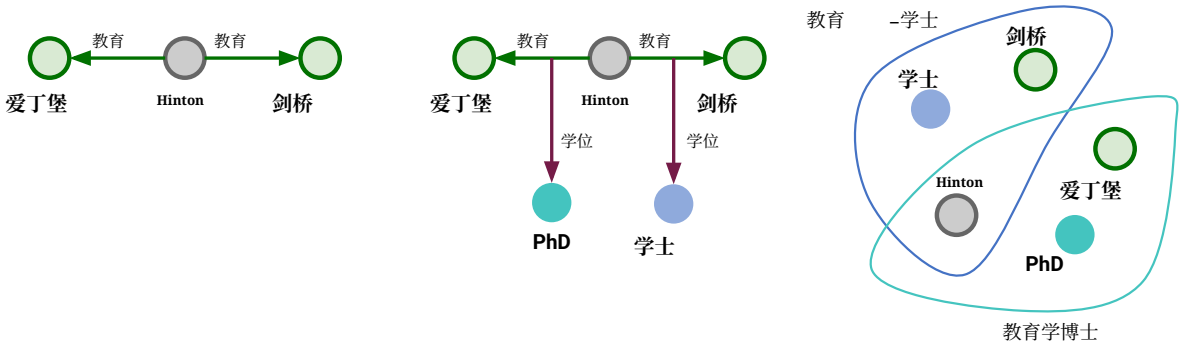
我们进一步提出了一种新颖的框架 **神经图数据库** （NGDB，第 3 节），其中复杂的近似查询求答（第 2 节）是核心任务，并通过神经查询嵌入方法加以处理。NGDB 由神经图存储和神经查询引擎组成，分别对应于常规图数据库中两个对应组件的神经版本。除了存储图结构以及多模态节点级/边级特征的图存储和特征存储之外，神经图存储还假设存在一个编码器，用于将原始信息进一步压缩到语义保持的潜在嵌入空间中，即，相似的实体/关系会被映射到相似的嵌入。在检索过程中，这种设计选择可以立即同时支持精确身份匹配和近似最近邻搜索。作为另一个关键组件，神经查询引擎负责在嵌入空间中对给定的输入查询进行优化、规划和执行，并最终通过与神经图存储交互来返回查询答案。我们针对 NGDB 提供了详细的概念框架，涵盖两个组件的不同设计原则和假设层次。我们希望这能阐明 NGDB 当前的研究现状，并为未来关于神经图数据库的研究提供路线图。

相关工作。 尽管已有关于通用图机器学习（Chami 等人，2022年）、知识图谱中的简单链接预测（Ali 等人，2021年；Chen 等人，2023年）以及基于逻辑的链接预测（Zhang 等人，2022b；Delong 等人，2023年）的深入综述，但迄今为止，复杂查询回答领域仍未得到综述。我们的工作填补了这一空白，并全面展现了这一新兴领域的现状。

2 预备知识

在介绍神经图数据库之前，我们先讨论数据库和图学习文献中常见的基础术语和预备知识。这包括不同类型图的定义、图查询及其向 SPARQL 等结构化语言的映射、近似图查询回答的形式化、逻辑运算符和模糊逻辑。完整的定义层次结构见 图 3。

2.1 图的类型



基于三元组的知识图谱

超关系知识图谱
（带有关系-实体限定词）

超图知识图谱
（使用临时定义的超边类型）

图 2：知识图谱模态示例。仅含三元组的图在节点之间具有有向二元边。超关系图允许在边上附加键值形式的关系-实体属性。超图由多个节点组成的超边构成。

这里介绍与本研究领域相关的不同类型的图。它们是各种查询推理工作的核心数据结构，也是我们的神经图存储模块（第3.1节）的核心数据结构。我们的定义改编自（Hogan 等人，2021年，第 2 章）。

我们首先定义一个集合，其中包含图中使用的所有元素

定义 2.1（Con） *Con*是一个无限的常量集合。[1]

定义 2.2（**（标准/基于三元组的）知识图谱**）我们将知识图谱（*KG*）$G = (E, R, S)$定义为：其中，$E \subset Con$表示节点（实体）集合，$R \subset Con$表示关系（边类型）集合，而$S \subset (E \times R \times E)$是边（三元组）集合 [2]。知识图谱 G 上的每条边 $s \in S$ 都表示一个陈述（或事实）(e_s, r, e_o)，其采用三元组格式，或采用 $r(e_s, e_o)$ 形式作为一阶逻辑（*FOL*）中的公式，其中 $e_s, e_o \in E$ 表示主语实体和宾语实体，而 $r \in R$ 表示主语与宾语之间的关系。通常，图会受到限制，即节点和边的集合必须是有限的；除非另有说明，我们都将作此假设。

例如，图1 中的知识图谱上有一条陈述(Hinton, university(UofT) 或university(Hinton, UofT)。请注意，知识图谱中的所有关系都是二元的，也就是说，每种关系恰好涉及两个实体。这也是资源描述框架（RDF）采用的方式（Brickley 等人，2014年），该框架将三元组定义为事实的基本表示。不过，RDF 允许对三元组作更宽泛的定义，其中宾语可以是字面量，包括数字和时间戳（详见下文）。

超关系知识图谱通过允许边带有关系-实体限定词，对三元组知识图谱进行了泛化。我们将其定义如下：

定义 *2.3*（超关系知识图谱——根据 *Alivanistos* 等（*2022*）改编）设 E 和 R 的定义同定义 *2.2*，令 $\Omega = 2^{(\mathcal{R} \times \mathcal{E})}$，我们将超关系知识图谱定义为 $G = (E, R, S)$。此时，边集 $S \subset (E \times R \times E \times \Omega)$ 由带限定符的陈述构成，其中每条陈述 $s = (e_s, r, e_o, qp)$ 都包含 qp，它们为主关系 r提供上下文信息。集合 $qp = \{q_1, \ldots\} = \{(qr_1, qe_1), (qr_2, qe_2) \ldots\} \subset R \times E$ 是限定符对的集合，其中 $\{qr_1, qr_2, \ldots\}$ 是限定关系，而 $\{qe_1, qe_2, \ldots\}$ 是限定实体。

我们注意到，如果 $\mathcal{E}$ 和 $\mathcal{R}$ 是有限集，那么 Ω 也有限，并且可能的（r，qp）组合只有有限种。因此，在这些条件下，通过定义一种规范的

[1]如果常量中不包含实数的表示，那么这个集合将是可数无限的。[2]$\mathcal{E}$ 和$\mathcal{R}$之间可能存在重叠，也就是说，某些常量可能既用作边，也用作节点。

排序方式对 Ω 进行排序，我们可以通过为每种组合创造一个新谓词r_{qp}，使用一阶逻辑表示超关系图。于是，定义中的陈述可以写作 $r_{qp}(e_s, e_o)$。
例如，(图2) 中超关系知识图谱上的一条语句是 (Hinton, education, Cambridge,{(degree, Bachelor)})。限定符对 {(degree, Bachelor)} 为主三元组提供了额外的上下文，并有助于将其与其他 education 事实区分开来。如果满足上述条件，我们可以将该语句写作 $\texttt{education}_{\{(\texttt{degree:Bachelor})\}}(\texttt{Hinton}, \texttt{Cambridge})$。

超关系知识图谱可以利用图中事实的额外限定符来捕获更多样化的事实集合。这一概念与 Hartig 等人 (2022年) 引入的 RDF-star（此前称为 RDF*）格式密切相关，它是 RDF 的扩展。RDF-star 明确将元数据（上下文）与数据（语句）分离。RDF-star 支持超关系图的超集；它扩展了宾语和值限定符位置上允许使用的值，将字面量也纳入其中，我们将在下文讨论这一点。

定义*2.4*（超图知识图谱）设 E 和 R 的定义如定义 *2.2* 所述，超图知识图谱是一个图 $G = (E, R, S)$，其中语句 $S \subset (R \times 2^{\mathcal{E}})$ 是超边。这样的超边$s = (r, e) = (r, \{e_1, \ldots, e_k\})$ 具有一个关系类型 r，并将 k 个实体连接在一起，其中这些实体的顺序并不重要。 k， e 的大小称为该超边的元数。

图2中的一个超边示例如下：图2 为 (education_degree, {Hinton, Cambridge, Bachelor})。这是一个三元陈述（由三个实体组成）。与超关系知识图谱不同，超边仅包含一种关系类型，无法自然地纳入更细粒度的关系组合。相反，超边的类型是陈述关系的合并集合，在超图知识图谱中，每一种组合都会产生一种新的关系类型。因此，超图知识图谱的组合性不如超关系知识图谱。也就是说，例如，在描述 major 事实的 education(Hinton, Cambridge) 关系时，超关系知识图谱可以简单地将其用作限定符，并保留两种关系；而超图模型则必须提出一种新的超边类型 education_major。随着关系数量的增加，这种创建新超边关系类型的策略可能导致组合爆炸。然而，超图模型更适合表示元数各异、且所有实体贡献相同的关系，例如 partnership(companyA, companyB, companyC)。

上文介绍的每种图类型都可以扩展为支持字面量。实现方式是允许节点或限定符值包含字面值。有些图（如属性图）允许节点包含属性，但这相当于创建额外的节点来存储这些属性值，并添加与之相连的关系。理论上，边类型也可以是字面量，但本文不考虑这种情况。

定义 *2.5*（带字面量的知识图谱）设 E 和 R 与上述某一种图类型中的定义相同，则相应的带字面量知识图谱 $G_{\mathcal{L}} = (E, R, S_{\mathcal{L}}, L)$ 还包含一个额外的字面量集合 $L \subset Con$，用于表示数值型、类别型、文本、图像、声波或其他连续值（L 与 E 和 R 不相交）。如果将标准知识图谱扩展为 *RDF* 图，字面量只能用于宾语位置，即 $S_{\mathcal{L}} \subset (E \times R \times (E \cup L))$。在 *RDF-star* 图中，字面量可以作为宾语或限定符值，即 $\Omega = 2^{(\mathcal{R} \times (\mathcal{E} \cup \mathcal{L}))}$ 和 $S_{\mathcal{L}} \subset (E \times R \times (E \cup L) \times \Omega)$。[3]对于这两种图，我们也可以根据需要定义在三元组其他位置包含字面量的图类型，或者在三元组的元素中引入更复杂的子结构（参见 *Cochez*（*2012*））。在超图知识图谱中，字面量可以引入超边的元素中，$S_{\mathcal{L}} \subset (R \times 2^{\mathcal{E} \cup \mathcal{L}})$

如果带字面量的知识图谱的可能陈述集合与不带字面量的知识图谱具有相同的有限性特征，那么关于使用一阶逻辑表达它的性质也不会改变。
一个宾语为字面量的三元组示例（图 10）是 (Cambridge, established, 1209)。字面量包含数值、类别、离散时间戳或文本数据，无法像实体那样轻易离散化为有限的可能值集合。与 Web 本体语言（OWL, Motik 等人（2009年）区分对象属性（连接两个实体）和数据类型属性（连接一个实体和一个字面量）类似，通常将连接两个实体的边称为关系，将连接属性的边

[3]RDF 和 RDF-star 还允许在主语和宾语位置使用空白节点，以表示身份未被明确指定的实体Brickley 等人 (2014年)。此外，它们还支持命名图（在某些情况下也支持四元组）。我们的形式化体系并未显式支持这些内容，但所有这些内容都可以使用超关系图进行建模。

称为连接实体与字面量的属性。例如，在 education(Hinton, Cambridge), education 是关系，而 established(Cambridge, 1209) 是属性。

上述知识图谱定义并不全面。还可以创建具有其他属性的图。例如，可以创建具有无向边、不带边标签、具有时间特征、具有概率关系或模糊关系等的图。此外，也可以使图或边具有组合特征。例如，可以定义带有限定信息的超边。鉴于选项如此繁多，我们决定将范围限制在上述有限的选项集合内。下一节将介绍如何查询这些图。

2.2 基本图查询求答

我们基于（Hogan 等人，2021年，第 2.2.1 节基本图模式）中的基本图查询定义，但将其调整为适用于我们的图形式化。

定义 2.6（项与变量集合）给定一个知识图谱 $G = (E, R, S)$（或等价的 $G = (E, R, S, L)$），我们定义变量集合 $Var = \{v_1, v_2, ...\}$，其取值来自 Con，但与其严格不相交。我们将变量与常量的并集称为项：$Term = Con \cup Var$

有了这些概念，基本图查询定义如下：

定义 2.7（基本图查询）基本图查询是一个四元组 $Q = (E', R', S', S')$（等价地，也可以表示为一个五元组 $Q = (E', R', S', S', L')$，其中包含字面量），其中 $E' \subset Term$是节点项集合，$R' \subset Term$ 是关系项集合，$S', S' \subset E' \times R' \times E'$，是两个边集（或等价地，采用带字面量的图边定义）。该查询看起来像两个（小型）图：一个由 S'，中的边构成，另一个由 S'中的边构成。前一个集合包含必须在图中匹配的边，以获得查询的答案；后一个集合则包含不得 匹配的边（即原子否定）。

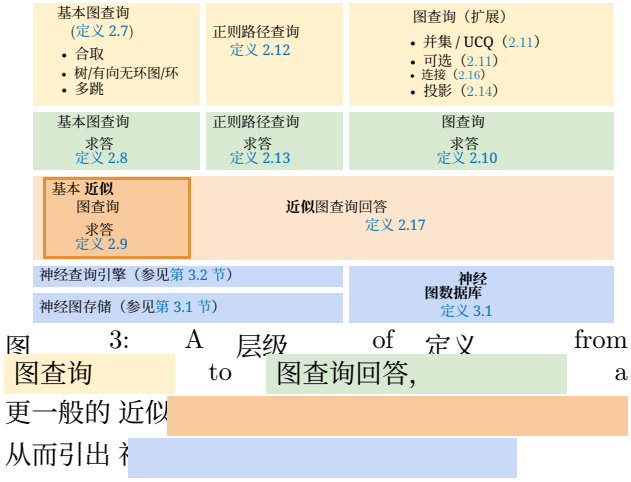


图 3: A 层级 of 定义 from 图查询 to 图查询回答, a 更一般的 近似 从而引出 神

在 Va 中$\mathtt{r}_Q = \mathcal{E}' \cap \mathtt{Var}$（查询中的所有变量），该查询的答案定义如下 .

定义 2.8（基本图查询回答）给定一个知识图谱 G 和一个如上所定义的查询 Q，该查询的答案是满足以下条件的任意映射 μ: $Var_Q \to Con$：将 S' 中边里的每个变量 v_i 替换为 $\mu(v_i)$ 后，得到的结果是 S 中的一条边；并且将 S' 中边里的每个变量 v_j 替换为 $\mu(v_j)$后，得到的结果是不在 S 中的一条边。该查询的所有答案构成所有此类映射的集合。

对于基于三元组的知识图谱，边集$\mathcal{S}$（分别为$\mathcal{S}'$）中的每条边（a, R, b）都可视为关系投影 $R(a, b)$（分别为 $\neg R(a, b)$），即二元函数。由于这些关系投影的合取（即全部）必须为真（否定情形亦然），我们也将这类带原子否定的查询称为合取查询（CQ$^{\text{neg}}$）。SPARQL 查询语言定义了基本图模式（BGP）。这些模式与我们针对 RDF 图的基本图查询非常相似，但$\mathcal{S}' = \emptyset$即不支持原子否定，仅支持合取查询（CQ）。我们也可以为其他类型的知识图谱类似地创建 CQ 和 CQ$^{\text{neg}}$ 类。

在图1 中 我们给出了一个以 SPARQL 查询形式表达的基本图查询示例。这与我们的形式化体系相对应：

$$
\begin{aligned}
Q = (&\{\texttt{TuringAward}, ?person\ \texttt{DeepLearning}, ?单\}, (\texttt{\# Node Terms}\ E' &&)\\
&\{win, 域, 大学\}, (\texttt{\# Relation Terms}\ R' &&)\\
&\{(\texttt{TuringAward}, 获胜, ?person), (\texttt{DeepLearning}, 域, ?person), (?person, 大学, ?uni)\}, \emptyset) && (\mathcal{S}')\\
& && (\overline{\mathcal{S}'})
\end{aligned}
$$

此查询的一个可能答案如下，这是一个部分映射： $\mu_1 = \{(?person\ \texttt{Hinton}), (?uni\ \texttt{UofT})\}$这也是唯一的答案，因此所有答案的集合为 $\{\mu_1\}$

如果我们想排除与 Welling 合作过的人，那么可以按如下方式修改查询：

查询 $=$ ({TuringAward, ?person DeepLearning, ?uni}, {获胜, 域, 大学}, {(TuringAward, 获胜, ?person), (DeepLearning, 域, ?person), (?person, 大学, ?uni)}, {(?person, , , 合作 **Welling**)} {(?person 合作)} {(?person 合作)})

请注意，这里的关键区别在于，我们将 {(?person, 合作, Welling)} 添加到了$\mathcal{S}'$。此查询的答案集为空。

2.3 基本近似图查询应答

到目前为止，我们假设知识图谱是完整的，也就是说，查询可以得到精确回答。然而，我们关注的是无法获取完整图的情形。该情形可描述如下：给定一个知识图谱 $\mathcal{G}$ （它是某个完整但不可观测的图 ˆ$\mathcal{G}$的子集）以及一个基本图查询 Q。基本近似图查询应答的任务是在无法Q的情况下回答该查询，而ˆ无需访问 $\mathcal{G}$。根据具体设定，该方法可以访问 $\mathcal{G}$，也可以访问一组示例（查询，答案）对，并据此生成答案（另见第7.3节）。在图1的示例中，我们还画出了几条虚线边。这些边确实存在，但ˆ仅位于图的不可观测部分$\mathcal{G}$中。

ˆ 目标是在已知$\mathcal{G}$ 的情况下找到答案集。在此情况下，完整答案集变为{μ_1，{(?person Bengio)，(?uni UdeM)}，{(?person LeCun)，(?uni NYU)}}，其中包括非近似版本的答案 μ_1。

如果我们的图是完整的，包含否定的查询将具有以下答案：{{(?person Bengio)，(?uni UdeM)}，{(?person LeCun)，(?uni NYU)}}，其中不包括 μ_1。

近似查询回答系统为每一个可能的映射[4]提供一个分数 $\in \mathbb{R}$。因此，这些系统提供的答案是一个从映射（即， μ）到其在 $\mathbb{R}$中相应分数的函数。

定义 2.9（基本近似图查询回答）给定一个知识图谱 G（它是完整但不可观测的知识图谱ˆG 的子图）、一个基本图查询 Q 以及评分域 $\mathbb{R}$，查询 Q 的一个基本近似图查询答案是一个函数 f，它将每个可能的映射（μ： $Var_Q \to Con$）映射到 $\mathbb{R}$。

目标是使根据图 ˆ$\mathcal{G}$ 得出的正确答案获得更高的分数（排名更靠前、概率更高或真值更高），而错误答案的分数较低。不过，无法保证可观测图 $\mathcal{G}$ 中的答案会获得高分。

对于我们的示例，一种可能的映射如表 1所示。表中的每一行对应一个映射 μ。理想情况下，所有正确的映射都应排在前面，而其他映射排在后面。不过，在这个示例中，我们看到有几个正确答案排名靠前，但也有两个错误答案。

2.4 图查询求答

[4]该分数可以表示映射的排名、似然或真值，也可以是表示取值的二进制值。

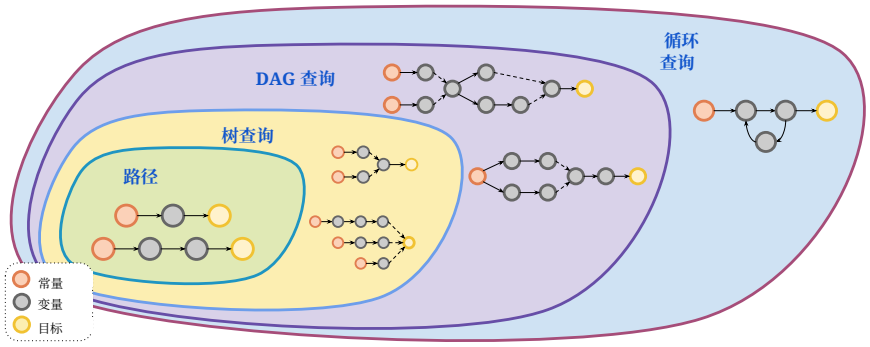


图 4：查询模式及其相对表达能力空间。多跳推理系统处理最简单的 路径 查询。现有的复杂查询回答系统支持 树 查询，而 有向无环图和循环 查询仍无法由当前的神经模型回答。

在前面的章节中，我们介绍了基本图查询，以及如何在精确或近似图查询的设置下对其进行求答。然而，不同方法能够回答的查询类型各不相同：有些方法仅支持基本图查询的一个子集，而另一些则支持更复杂的查询。在本节中，我们将再次聚焦于精确（非近似）查询回答，先探讨一些可能的限制，再考察一些扩展。我们还将重点介绍与一阶逻辑片段的关联。

表 1：示例查询的有序评分映射。两个错误的映射以红色显示。

?人物	?uni	分数
Hinton	UofT	40
Bengio	UdeM	35
Welling	UofT	34
UdeM	Hinton	33
LeCun	NYU	32
...	...	...

定义 2.10（图查询求答）给定知识图谱 G、查询形式体系，以及根据该形式体系定义的图查询 Q，查询的一个答案是满足查询形式体系要求的任意映射 $\mu: \mathit{Var}_Q \to \mathit{Con}$。所有答案的集合就是所有此类映射组成的集合。

对于我们在 定义 2.7中介绍的基本图查询，查询形式体系规定了图中哪些边必须存在、哪些边不得存在（定义 2.8）。在该上下文中，我们已经提到合取查询，它们可以带有否定（CQ$^{\text{neg}}$），也可以不带否定（CQ）。如果满足使用一阶逻辑（FOL）表示图的条件，那么我们可以等价地用一阶逻辑表示基本图查询。查询中的每个变量都变为存在量词化变量，而公式则变为以下两部分的合取：1）由 S'中的三元组形成的项；2）由 S'中的三元组形成的项的否定。如果查询图的边上存在变量，那么我们可以将二阶公式改写为一阶逻辑，方法是将其解释为有限个选项中所有可能谓词上的析取。上文的示例查询变为以下一阶逻辑句子。

$$q = \exists ?person, ?uni: \text{获胜}(\texttt{TuringAward}, ?person) \wedge \text{域}(\texttt{DeepLearning}, ?person) \wedge \text{大学}(?person, ?uni)$$

查询的答案是变量赋值函数。对于若干限制性更强的片段，使用这种一阶逻辑语法来表述查询很有用。

限制 首先可以引入的限制是**多跳查询**，这在自然语言处理和知识图谱文献中早已为人所知（Guu 等人，2015年；Das 等人，2017年；Asai 等人，2020年），主要出现在问答语境中。形式上，多跳查询（或路径查询）是形成链式结构的合取查询（CQ），其中一个投影的尾部是下一个投影的头部，也就是说，

$$q_{\text{path}} = V_k, \exists V_1, \ldots, V_{k-1}: r_1(v, V_1) \wedge r_2(V_1, V_2) \wedge \cdots \wedge r_k(V_{k-1}, V_k)$$

其中 $v \in \mathcal{E}$、$\forall i \in [1, k]$： $r_i \in \mathcal{R}$ $V_i \in \texttt{Var}$，且所有 V_i 都是存在量化变量。换句话说，路径查询不包含分支交集，可以通过获取节点的邻居来迭代求解。也可以定义允许否定的多跳查询。

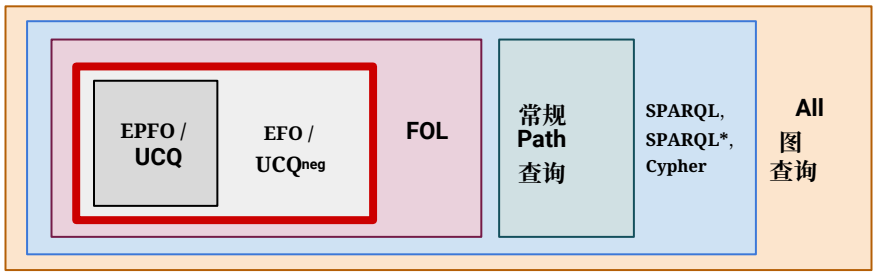


图 5：当前查询回答模型涵盖存在正一阶逻辑（EPFO）和存在一阶逻辑（EFO）逻辑片段（以红色矩形标出）。EPFO 和 EFO 分别等价于合取式并集（UCQ）以及带原子否定的合取式并集（UCQ$^{\text{neg}}$）。反过来，这些逻辑片段又是一阶逻辑（FOL）的子集。反过来，一阶逻辑查询也只是图数据库语言所能回答的查询的子集。例如，正则路径查询无法用一阶逻辑表达。SPARQL、SPARQL* 或 Cypher 等语言涵盖了所有查询类型以及更多类型。

限制 CQ 和 CQ 的其他方式$^{\text{neg}}$也存在，这些方式产生的查询比多跳查询具有更强的表达能力。可以定义形如树、DAG，并允许包含循环部分的逻辑查询族。如图 4所示，路径（多跳）查询构成表达能力最弱的逻辑查询集合。树查询通过交集运算符连接更多推理分支；DAG 查询取消查询树这一要求，允许查询呈现为有向无环图；最后，循环查询进一步取消无环这一要求。请注意，这些查询不允许在谓词位置使用变量。此外，所有实体（在此上下文中称为锚点）都必须在拓扑排序中位于所有变量之前。

我们将在第 6.2 节 中进一步阐述这些查询类型，并指出，在第 5 节 所调查的神经复合逻辑查询问答方法中，大多数仍然局限于树状查询，在表达能力方面落后于许多图数据库查询语言。弥合这一差距是未来工作的重要方向。

扩展 我们引入的第一个扩展是**并集**。

定义 2.11（映射集合的并集）给定两个映射集合（例如 μ 定义定义 *2.8*中的映射集合），我们可以通过取它们的并集来创建一个新的映射集合。这个并运算符满足交换性和结合性，因此我们也可以讨论三个或更多映射集合的并集。输入集合中的映射，其领域可以不同。

我们可以对两个或更多底层查询的结果应用并操作，从而定义一种新的查询类型。如果这些底层查询是基本图查询，我们就称这种新型查询为**带否定的合取查询并集（UCQ$^{\text{neg}}$）**；如果基本查询不包含否定，则称为**合取查询并集（UCQ）**。这些类在一阶逻辑中也很常见，并且确实对应于合取式析取。

例如，以下查询属于 UCQ$^{\text{neg}}$，因为它是由合取项构成的析取，这些合取项由原子式 a_i 组成，而原子式是关系投影或其否定：

$$q = v_?.\exists v_1, \ldots, v_n : \Big(\underbrace{r_1(c, v_1)}_{a_1} \wedge \underbrace{r_2(v_1, v_2)}_{a_2}\Big) \vee \Big(\underbrace{\neg r_3(v_2, v_3)}_{a_3}\Big) \vee \cdots \vee \underbrace{r_k(v_n, v_?)}_{a_m}$$

此外，还有一些与这些片段等价的一阶逻辑片段。具体而言，**EPFO** 中的所有查询，即存在正一阶逻辑句子，都有一个等价的 UCQ 查询；而**EFO** 中的所有查询，即存在一阶逻辑句子，都有一个等价的 UCQ 查询$^{\text{neg}}$。原因在于，EPFO 和 EFO 句子可以写成析取范式（DNF），即合取项的并集。

某些查询语言（如 SPARQL）允许在查询中加入可选部分。在我们的形式化体系中，可以使用并运算符定义查询的**可选**部分。假设查询中有 n 个可选部分，则创建 2^n 个不同的查询，其中移除可选部分的其他组合。然后，查询的答案是

所有这些查询的答案的并集。如果查询语言本身已经支持并集，那么可选项不会使其表达能力更强。

除了这些扩展之外，还可以进一步扩展到**能够用一阶逻辑表达的所有查询**，这需要全称量化或合取的否定。不过，这些仍不是所有可能的图查询。有一种有趣的查询不属于一阶逻辑，即**合取正则路径查询**。这类似于我们上文讨论的路径查询，但不指定长度。

定义 2.12（正则路径查询）正则路径查询是一个三元组(s, R, t)，其中 $s \in Term$ 是路径的起始项，$R \in Term$ 是路径的关系项，$t \in Term$ 是路径的结束项。该查询表示一条从 s出发、经过任意数量的 R 类型边，最终到达 t的路径。

由于这类查询是任意长度的合取，因此无法用一阶逻辑表示。若要表达固定长度的路径，可以使用上文所述的多跳路径。若要表达最大长度，则可以将所有允许的长度取并集。在后两种情况下，查询仍可用 $EPFO$ 表达。

定义 2.13（正则路径查询应答）给定一个知识图谱 G 和一个正则路径查询 $Q = (s, R, t)$。查询的一个答案是任意映射 $\mu: Var_Q \to Con$，如果将 Q 中的所有变量 v 替换为 $\mu(v)$，得到 $(\hat{s}, \hat{}R, \hat{}t)$，那么图中存在一条路径，从节点 $\hat{s}$ 出发，经过一条或多条类型为 $\hat{}R$ 的边，并终止于 $\hat{}t$。该查询的所有答案构成的集合，就是所有此类映射的集合。

正则路径查询有多种变体，也可以与上述查询类型结合，形成新的查询类型。在图 5中，我们展示了这些片段与其他查询类之间的关系。大多数方法都严格属于存在一阶逻辑片段，即，它们仅支持带有上述限制的子集。我们将在第 6 节进一步讨论这些方法的局限性。

这里我们想强调的最后两个方面是投影 和连接。

定义 2.14（查询答案的投影）给定一个查询答案 μ，以及一组变量 $V \in Var$。查询答案在变量 V 上的投影为 $\{(var, val) | (var, val) \in \mu, var \in V\}$。

换言之，它是将答案限制在一组特定变量上的结果。上述介绍的查询形式都可以添加投影，以便只获取特定变量的值。这对应于 SPARQL 中的 `SELECT ?var` 子句。或者，也可以投影 $\mathcal{V}$ 中的所有变量，这等价于 `SELECT *` 子句。不包含任何投影变量的查询是一个布尔子图匹配问题，等价于 SPARQL 中的 `ASK` 查询。

连接用于将两个独立查询的结果合并为一个。具体来说，

定义 2.15（查询答案的连接）给定两个查询答案 μ_A 和 μ_B，以及 Var_A和 Var_B，分别表示 μ_A 和 μ_B 中出现的变量。这两个答案的连接仅在它们共有的所有变量取值相同时存在，即 $\forall var \in Var_A \cap Var_B$：$\mu_A(var) = \mu_b(var)$。在这种情况下，连接$(\mu_A, \mu_B) = \mu_A \cup \mu_B$。

两个答案集的连接定义如下。

定义 2.16（查询答案集的连接）给定两个查询答案集 A 和 B，它们的连接是一个新集合 $\text{join}(A, B) = \{\text{join}(a, b) | a \in A, b \in B,$ 以及 $\text{join}(a, b)$ 存在$\}$。

此操作使我们能够将多个底层查询（可能属于多种类型）组合成一个查询。例如，给定上文基本图查询示例中的答案集：

$A = \{\{(?\text{人物 Hi nton}), (?\text{单 UofT})\}, \{(?person\ \texttt{Bengio}), (?\text{单 UdeM})\}, \{(?person\ \texttt{LeCun}), (?\text{单 NYU})\}\}$

以及另一个答案集

$B = \{\{(?person\ \texttt{Hinton}), (?\text{出生}, 1947)\}, \{(?person\ \texttt{Bengio}), (?\text{出生}, 1964)\}, \{(?person\ \texttt{Welling}), (?\text{出生}, 1968\text{年})\}\}$

这些结果连接后为：

连接(A, B) $=\{\{(?person$ Hinton$), (?$单 UofT$),(?$出生$, 1947)\}, \{(?person$ Bengio$), (?$单 UdeM$), (?$出生$,1964)\}\}$

我们将在第6.1节进一步讨论连接，在那里我们将使用这些基本构件来定义更广泛的一组查询算子，旨在涵盖 SPARQL 中存在的所有操作。这包括克莱尼加号/星号 (+/*)，用于构建属性路径、FILTER、OPTIONAL，以及不同的聚合函数。

2.5 近似图查询回答

现在我们已经定义了什么是基本近似图查询回答，可以进一步定义更一般的近似图查询回答。近似图查询回答的定义与基本情形相同，但它并非只提供基本查询的答案，而是涉及更广泛的查询类型。

定义 2.17（近似图查询回答）给定一个知识图谱 G，它是一个ˆ完整但不可观测的知识图谱 G 的子图、一个查询形式体系、根据该形式体系的任意图查询 Q，以及评分域 $\mathbb{R}$。

查询 Q 的近似图查询答案是一个函数 f，它将每个可能的映射（μ: $Var_Q \to Con$）映射到 $\mathbb{R}$。

请注意，变量并不总是映射到图中出现的节点。查询很可能包含一个聚合函数，其结果为字面值。

在相关文献中，我们可以找到简单 答案和困难 答案的概念。这指的是答案是否仅通过访问$\mathcal{G}$ 即可找到。

定义 2.18（简单答案和困难答案）给定知识图谱 $\mathcal{G}$、更大且不可观测图 ˆ$\mathcal{G}$ 的子图，以及查询 $\mathcal{Q}$。简单答案和困难答案根据精确查询回答（定义 *2.10*）进行定义。 简单答案集合是从 $\mathcal{G}$ 获得的答案与从 ˆ$\mathcal{G}$ 获得的答案的交集。 ˆ困难答案集合是从 G 获得的答案与从 $\mathcal{G}$获得的答案之间的集合差。

ˆ请注意定义中的不对称性。简单答案是在 $\mathcal{G}$ 和 $\mathcal{G}$ 中都可以找到的答案。困难答案是只能在 ˆ$\mathcal{G}$ 中找到、但不在 $\mathcal{G}$ 中的答案。对于基本图查询（定义 2.7），所有简单答案也都可以从 ˆ$\mathcal{G}$ 中找到。然而，对于某些更复杂的查询类型（例如，允许使用否定的查询），在 $\mathcal{G}$ 中还可能存在不在 ˆG 中的答案。我们将这些答案称为在 **G** ˆ **上进行查询回答时的** $\mathcal{G}$假阳性。

2.6 三角范数与余范数

逻辑查询的求答意味着执行逻辑运算符。反过来，近似查询回答意味着输入为连续向量，输出真值不一定是二值的。此外，这些方法通常要求逻辑运算符平滑且可微。三角范数（T-范数）和三角余范数（T-余范数）分别将逻辑合取和析取泛化为真值连续空间上的函数，并实现模糊逻辑运算。

T-范数定义了一个连续函数 $\top$: $[0,1] \times [0,1] \to [0,1]$，其具有以下性质: $\top(x, y) = \top(y, x)$（交换性），$\top(x, \top(y, z)) = \top(\top(x, y)\ z)$（结合性），以及 $y \leq z \to \top(x, y) \leq \top(x, z)$（单调性）。此外，1 是 $\top$ 的单位元，即 $\top(x, 1) = 1$。T-范数的目标是用连续函数泛化逻辑合取。T-余范数可视为 T-范数的对偶，它同样定义了一个函数 $\perp$，其定义域和值域相同: $\perp$: $[0,1] \times [0,1] \to [0,1]$。T-余范数使用连续函数 $\perp$ 将析取泛化到模糊逻辑中。函数 $\perp$ 满足与 $\top$ 相同的交换性、结合性和单调性，唯一的区别是 0 为单位元，即 $\perp(x, 0) = x$。

存在许多源自相应逻辑形式化体系的三角范数、余范数和模糊否定（Klement 等人，2013年；van Krieken 等人，2022年），例如，（1）*Gödel* 逻辑定义三角范数:

$\top_{\min}(x,y) = \min(x,y)$，三角余范数： $\perp_{\max}(x,y) = \max(x,y)$；（2）乘积逻辑的三角范数： $\top_{\text{prod}}(x,y) = x \cdot y$，三角余范数： $\perp_{\text{prod}}(x,y) = x + y - x \cdot y$；（3）在卢卡西维茨逻辑中，三角范数： $\top_{\text{Łuk}}(x,y) = \max(x+y-1, 0)$，三角余范数： $\perp_{\text{Łuk}}(x,y) = \min(x+y, 1)$。使用模糊否定 $N(x) = 1 - x$，很容易验证$\perp(x,y) = N(\top(N(x)\ N(y)))$（德摩根定律）自然得到一对（$\top, \perp$）。

2.7 图表示学习

图表示学习（GRL）是机器学习的一个子领域，旨在学习图或其元素（例如单个节点（Hamilton，2020年）的低维向量表示。例如，$\mathbf{h}_v \in \mathbb{R}^d$ 表示与节点 d关联的维向量 v。从概念上讲，我们希望图中具有某些相同结构和语义特征的节点拥有相似的向量表示（其中相似度通常通过距离函数或其变体来衡量）。

浅层嵌入 早期的图表示学习方法专注于学习浅层节点嵌入，即为每个节点直接学习一个唯一向量，并将其用于优化任务。对于同质（单关系）图，DeepWalk（Perozzi 等，2014年）和 node2vec（Grover 和 Leskovec，2016）通过预测图中的游走来训练节点嵌入；而在多关系图中，TransE（Bordes 等，2013年）采用自编码器方式，通过重构邻接矩阵来训练节点和边类型嵌入。

图神经网络 图神经网络的 图神经网络 (GNNs)（Scarselli 等人，2008年）的思想是，通过执行邻域聚合，在给定（或可学习的）节点特征之上添加一个具有共享参数的神经网络编码器并进行学习。该框架可以推广到消息传递 （Gilmer 等人，2017年），在每一层中， t 一个节点 v 都会接收来自其邻居的消息（可能还会加入边特征和图特征），以置换不变的方式聚合这些消息，并更新其表示：

$$\mathbf{h}_v^{(t)} = \textsc{Update}\Big(\mathbf{h}_v^{(t-1)}, \textsc{Aggregate}_{u \in \mathcal{N}(v)}\big(\textsc{Message}(\mathbf{h}_v^{(t-1)}, \mathbf{h}_u^{(t-1)}, \mathbf{e}_{uv})\big)\Big)$$

这里，$\mathbf{h}_u$ 是邻居节点 u的特征，$\mathbf{e}_{uv}$ 是边特征。消息函数从节点 u 向节点 v 构建消息，并可通过神经网络参数化。由于邻居集合$\mathcal{N}(v)$是无序的，聚合函数通常采用置换不变函数，例如求和 或求均值。更新函数将节点之前的状态与邻居聚合后的消息作为输入，生成该节点在第 t 层的最终状态 v，并且也可以通过神经网络参数化。

经典的图神经网络架构，如图卷积网络（Kipf & Welling，2017年）、图注意力网络（Velickovic 等，2018年）和 GIN（Xu 等，2019a）旨在处理同质单关系图。此后，若干研究开发了适用于具有多重关系的异构图的图神经网络架构（Schlichtkrull 等，2018年；Vashishth 等，2020年；Zhu 等人，2021年）。图神经网络和消息传递为几何深度学习(Bronstein 等人，2021年的发展铺平了道路；几何深度学习利用输入数据中的对称性和不变性，将其作为构建深度学习模型的归纳偏置。

3 神经图数据库

传统数据库（包括图数据库）围绕两个关键模块进行设计：用于存储数据的存储层，以及用于处理针对已存储数据的查询的查询引擎。从这一角度来看，神经数据库本身并不是一个新颖的术语，许多机器学习系统在数据被编码到模型参数中时，实际上已经采用了这一范式；此时，查询等价于一次前向传播，可以为下游任务输出新的表示或预测。神经数据库最早的实例之一是向量数据库。在向量数据库中，存储模块由数据的领域无关向量表示组成，这些数据可以是多模态的，例如文本段落或图像。向量数据库属于面向存储的系统，通常构建在近似最近邻库（ANN）之上，例如 Faiss（Johnson 等人，2019年）或 ScaNN（Guo 等人，2020年），用于求答基于距离的查询（例如最大内积搜索，即 MIPS）。由于与编码器无关（也就是说，任何能够生成向量表示的编码器都可以作为来源），向量数据库缺乏图推理和复杂查询回答能力。不过，

正如我们将在下文介绍的，近似最近邻系统是实现神经图数据库某些层的便捷选择。

随着大规模预训练模型（即基础模型（Bommasani 等人，2021年））的兴起，我们见证了它们在自然语言处理和计算机视觉任务中的巨大成功。我们认为，这类基础模型也是神经数据库的典型范例。在基础模型中，“存储模块”可能直接由模型参数表示，也可能外包给外部索引；后者常用于检索增强模型（Lewis 等人，2020年；Guu 等人，2020年；Alon 等人，2022年），因为即使将所有世界知识编码进数十亿个模型参数中也很困难。“查询模块”通过在编码器模型（BERT 或 T5 风格）中填空，或通过仅解码器模型（GPT 风格）中的提示来执行上下文学习；这些模型可涵盖多种模态，例如用于视觉应用的可学习标记（Kolesnikov 等人，2022年），甚至还可以调用外部工具（Mialon 等人，2023年）。在文本模态方面，Thorne 等人（2021a;b）设计了自然语言数据库（NLDB），其中基本元素是通过预训练语言模型（LM）编码为向量的文本事实。发送给 NLDB 的查询采用自然语言话语的形式，这些话语会被编码为向量，而查询处理则采用检索器—阅读器方法。首先，稠密神经检索器返回一组候选事实作为支持集，随后，细粒度神经阅读器对这些候选项执行连接操作（作为序列到序列任务）。

图结构数据的规模庞大，涵盖众多领域，例如 Freebase（Bollacker 等人，2008年）、DBpedia（Lehmann 等人，2015年）、维基数据（Vrandecic 和 Krötzsch，2014年）、YAGO（Pellissier Tanon 等人，2020年）等所包含的常识，ConceptNet（Speer 等人，2017年）、ATOMIC（Hwang 等人，2021年）等所包含的常识知识，以及 Bio2RDF（Callahan 等人，2013年）和 PrimeKG（Chandak 等人，2023年）等所包含的生物医学知识。随着数据规模不断扩大，这些图谱的不完整性也同时加剧。在如此大规模下，符号方法难以提供有效的方式来处理不完整性。因此，我们认为，神经推理和图表示学习方法能够大规模应对不完整性，同时保持较强的表达能力，并支持超越简单链接预测的复杂查询。我们提出在神经图数据库（NGDB）框架下研究这些方法。NGDB 的概念将神经数据库的理念扩展到图领域。NGDB 融合了传统图数据库的优势（将图作为一等公民、高效存储和统一的查询接口）与现代图机器学习的优势（几何和受物理学启发的向量表示、默认能够处理不完整且含噪的输入，以及针对下游应用的大规模预训练和微调）。与 Besta 等人（2022年）提出的 LPG2Vec 工作不同，后者提出了一种针对带标签属性图的特征化策略，以便随后用于标准图数据库流水线；我们将 NGDB 概念设计为神经优先。

根据定义 近似图查询回答（定义 2.17），我们将神经图数据库定义如下.

定义 3.1（神经图数据库，NGDB）神经图数据库（见图6）是一个元组$(S,\ E,\ f_\theta)$。 S是神经图存储（见第 3.1 节），E是神经查询引擎（见第 3.2 节），而f_θ是参数化的近似图查询应答函数，其中 θ 表示一组参数。

具体而言，我们的神经图数据库设计议程包括：

- **数据不完整性假设**，即，底层数据在节点级、链接级和图级可能存在缺失信息，而我们希望在查询回答中对其进行推断和利用；
- **归纳性与可更新性**，即，与支持更新和即时查询的传统数据库类似，用于构建图潜在表示的表示学习算法必须具有归纳性，并能以零样本（或少样本）的方式泛化到未见数据（推理时的新实体和关系），以避免代价高昂的重新训练（例如，重新训练浅层节点嵌入）；
- **表达能力**，即，潜在表示能够编码数据中的逻辑关系和语义关系，类似于一阶逻辑（或其片段），并在查询回答中加以利用。实际而言，神经推理所支持的逻辑运算符集合应接近或等同于 SPARQL 或 Cypher 等标准图数据库语言；

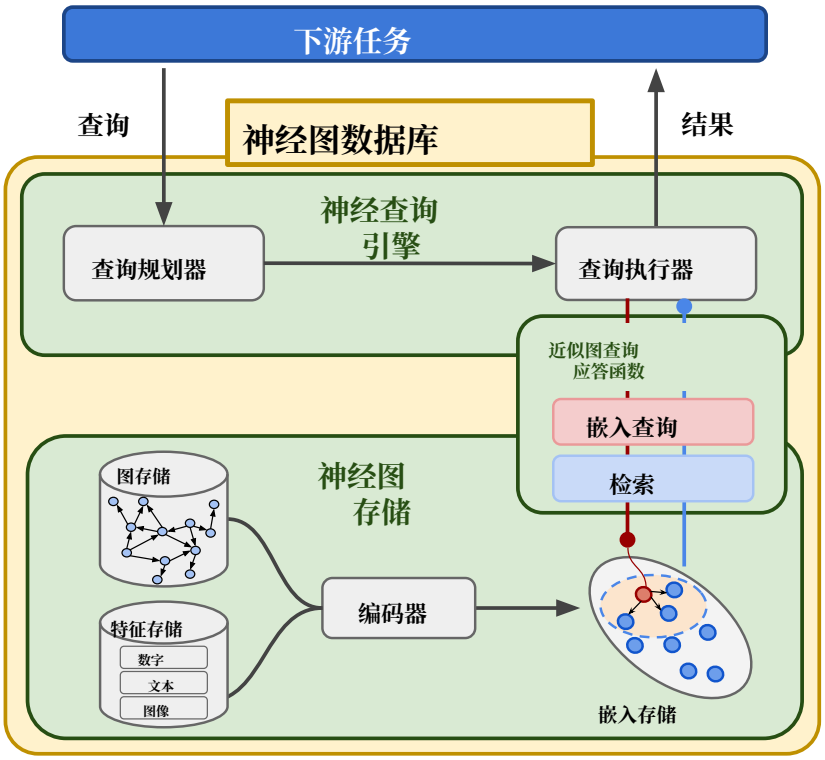


图6：神经图数据库的概念方案。输入查询由神经查询引擎处理，其中规划器推导出查询的计算图，而执行器则在潜在空间中执行查询。神经图存储利用图存储和特征存储，在嵌入存储中获取潜在表示。执行器通过近似图查询应答函数 f_θ （定义 *3.1*）与嵌入存储交互，以检索并返回结果。

- **多模态性** 不仅限于知识图谱，即，任何可作为节点或记录存储在传统数据库中的图结构数据（例如由图像、文本、分子图或带时间戳的序列构成），只要能够赋予其向量表示，都是神经图存储和神经查询引擎的有效来源。

解决神经图数据库设计议程的关键方法如下：

- **以向量表示作为基本元素**，即，传统图数据库会将邻接矩阵（或边列表）哈希到多个索引中，而不完备性假设意味着，给定的边**和**图潜在表示（向量表示）都成为事实来源，并存储在神经图存储中；

- **在潜在空间中进行神经查询执行**，即，由于不完备性假设，边遍历等基本操作无法仅通过符号方式执行。因此，神经查询引擎同时处理邻接关系和图潜在表示，将可能缺失的数据纳入查询回答；

NGDB 的概念架构如图6所示。从更高层次来看，NGDB 包含两个主要组件： 神经图存储 和 神经查询引擎 ，我们将在以下第 3.1 节和第 3.2 节中分别进行介绍。处理流程从某个应用程序或下游任务发送的查询开始，该查询已经采用结构化格式（例如，如果初始查询采用自然语言，则可通过语义解析（Drozdov等，2022年）获得）。查询首先到达神经查询引擎，具体而言，

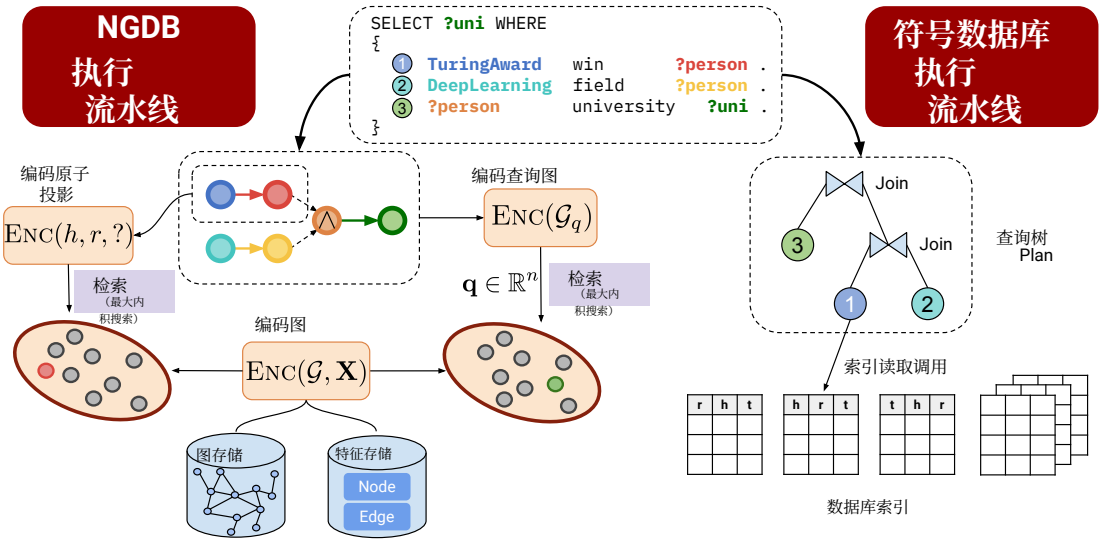


图 7：神经图数据库（左）和传统数据库（右）的存储与执行流水线。传统数据库将图存储在一组查找索引中，查询树计划中的每个查询模式都由其中一些索引来回答。在神经图数据库中，由于图并不完整，首先将图及其特征编码到潜在空间中。查询（或其原子模式）也会被编码到潜在空间中，并通过检索模块进行探测（例如，可使用最大内积搜索实现）。

到查询规划器 模块。查询规划器的任务是根据查询复杂度、预测任务以及底层数据存储（例如可能的图分区），推导出由原子操作（例如投影和逻辑操作）构成的高效计算图。我们将在第 3.2 节中详细说明该规划机制与经典数据库中标准查询规划器的相似之处和差异。随后，推导出的计划会被发送至查询执行器，后者将查询编码到潜在空间中，在底层图及其潜在表示上执行原子操作，并将原子操作的结果聚合为最终答案集。执行通过检索 模块完成，该模块与神经图存储进行通信。存储层由以下部分组成：(1) 用于以节省空间和时间的方式存储多关系邻接矩阵的图存储 （例如，采用 COO 和 CSR 等各种稀疏格式）；(2) 用于存储与底层图相关联的节点级和边级多模态特征的特征存储；(3)嵌入存储，该存储利用编码器 模块，根据底层邻接关系及相关特征，在潜在空间中生成图表示。检索模块查询编码后的图表示，为原子操作构建潜在答案的分布。在以下小节中，我们将更详细地介绍神经图存储和神经查询引擎。

3.1 神经图存储

在传统图数据库中，存储设计通常取决于图建模范式。最流行的两种范式是资源描述框架（RDF）图（Brickley 等人，2014年）和带标签属性图（LPG）（Besta 等人，2022年）。虽然对这些框架进行详细比较超出了本文的讨论范围，但其主要区别在于：RDF 是一种基于三元组的模型，具备适用于符号推理的某些形式语义；而 LPG 允许在边上添加字面量属性，但不具备形式语义。我们认为，新的 RDF-star 范式（Hartig et al.，2022年）将成为图建模的融合方向，兼具两者之长，即，既允许在边上添加属性（从而能将其他节点和关系作为键值属性），又具备丰富的形式语义。我们注意到，超关系知识图谱（定义 2.3）和维基数据陈述模型（Vrandecic 和 Krötzsch，2014年）在概念上与 RDF-star 范式相近。在物理层面，图数据库会使用各种索引和数据结构来存储边，并针对读取（例如，B+ 树（Neumann &Weikum，2008年）或 HDT（Fernández 等人，2013年））或写入应用（例如日志结构合并树（Sagi 等人，2022年））进行优化。

然而，与上述方法不同，我们提出了神经图存储（图 7）的概念，其中输入图及其向量表示都是事实来源。从物理结构上看，它由以下部分组成：(1) 图存储，用于存储（多关系）邻接矩阵，其基本形式可通过稀疏 COO、CSR 或更高效的压缩格式实现；(2) 特征存储，用于存储各种格式和模态的节点级与边级特征向量，例如数值、类别、文本数据或已有向量；(3)嵌入存储，利用编码器根据底层邻接关系及相关特征，在潜在空间中生成图表示。嵌入存储是神经图存储与传统图数据库对应方案之间最大的区别之一。嵌入过程可以视为一种压缩步骤，但实体/关系的语义和结构相似性得以保留。嵌入空间中实体/关系之间的距离应与其语义/结构相似性呈正相关。编码器的架构有多种选择。

首先，我们可以将编码器实现为矩阵查找，其中每一行对应一个实体/关系的嵌入。这种建模的优点是灵活性最高，并且自由参数数量最多——嵌入矩阵中的任何条目都可以训练。然而，这种方法的代价是难以编辑存储。我们无法轻松地向存储中添加新实体/关系，因为任何新实体/关系都可能需要一个必须从头开始学习的新嵌入。如果移除某个实体后又将其添加回来，这种嵌入矩阵建模也无法恢复原先学到的嵌入。

编码器的另一种选择是图神经网络（GNN），可以从消息传递和邻域聚合的角度对其进行分析（第 2.7 节）。其思路是通过聚合实体/关系的邻居信息来学习其嵌入。图神经网络参数在图中的所有实体和关系之间共享。这种建模具有明显优势，其泛化能力远胜于浅层嵌入矩阵。

神经图存储中的一个重要过程是**检索**步骤。传统图数据库直接通过索引执行基于标识的精确匹配检索。相比之下，在神经图存储中，由于我们将每个实体和关系存储在潜在空间中，除了通过实体/关系 ID 执行检索外，用户还可以输入一个向量，检索过程则可以通过测量嵌入空间中的距离，返回“相关”的实体/关系。检索过程可以视为在嵌入空间中针对输入向量进行最近邻搜索。与传统存储相比，神经图存储有三个直接优势。第一个优势是，检索在嵌入空间中通过预定义的距离函数执行，因此每个检索到的项目都会自然附带一个分数，该分数可能表示输入向量在检索步骤中的置信度/相关性。此外，神经图数据库支持对潜在空间和距离函数进行不同定义（详见第5.2节），因此具有灵活性，用户可以根据不同的期望属性和用户需求，自定义潜在空间和距离函数。最后，借助有关高效最近邻搜索的大量研究成果，我们有机会以高效率和可扩展性，在包含数十亿个节点和边的超大型图上实现检索步骤。现有框架包括 Faiss（Johnson 等人，2019年）、ScaNN（Guo 等人，2020年），它们提供了可扩展的最近邻搜索实现。然而，一个主要局限是，现有框架仅适用于 L1、L2 和余弦距离函数。如何为更复杂的距离函数（例如 KL 散度）设计高效且可扩展的最近邻搜索算法，仍然是一个开放的研究问题；这样，我们就能针对不同的 CLQA 方法，以高得多的效率执行检索。

3.2 神经查询引擎

在传统数据库中，典型的查询引擎（Neumann & Weikum，2008年；Endris 等人，2018年）会执行三项主要操作。（1）查询解析，用于验证语法正确性（通常还会结合对查询词项进行更深入的语义分析）；（2）查询规划与优化，用于推导出高效的查询计划（通常为由关系运算符构成的树），以最小化计算成本；（3）查询执行，根据查询计划扫描存储并处理中间结果。

在神经图数据库中，事实来源既包括图邻接关系（可能不完整），也包括潜在图表示（例如，实体和关系的向量表示）。同样，查询（或其原子操作）也会被编码到潜在空间中。为了在潜在空间上回答编码后的查询，我们设计了包含两个模块的神经查询引擎：(1) 查询规划器，用于推导出由原子操-

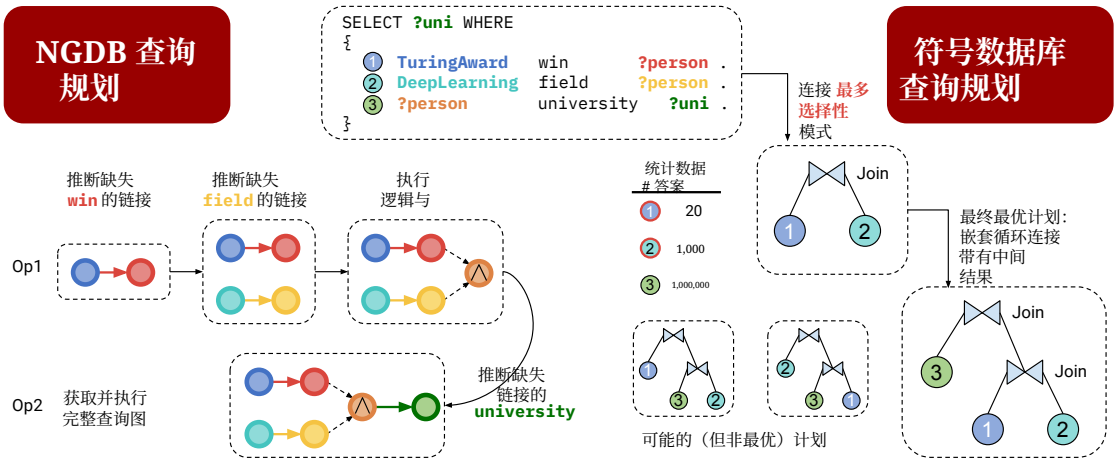


图 8：神经图数据库（左）与传统图数据库（右）中的查询规划。神经图数据库的规划（假设图不完整）可以逐步自回归地执行（1），也可以一次性生成（2）。传统数据库的规划基于成本，并借助元数据（假设图完整且从图中提取），例如中间答案的数量，来构建连接算子树。

作组成的高效查询计划（例如投影和逻辑算子），在考虑查询复杂度、预测任务以及事实来源划分的基础上，最大限度地提高完整性（必须返回现有边上的所有答案）和推理能力（动态预测缺失边）；(2) 查询执行器，用于将推导出的计划编码到潜在空间中，在图及其潜在表示上执行原子操作（或整个查询），并将中间结果聚合为最终答案集，同时进行可能的后处理。

总体而言，在深度学习领域，提示大语言模型（LLM）可视为一种神经查询引擎，其中查询是非结构化文本。近年来，思维链（Wei 等人，2022年）或思维程序（Chen 等人，2022a）等提示技术，通过提供少量“常识”示例，展示如何逐步解决给定问题，在自然语言推理方面取得了显著进展（Qiao 等人，2022年）。这类逐步指令类似于由提示工程师手动设计并针对特定大型语言模型优化的查询计划；其中，查询执行被表述为语言模型的目标（预测接下来的若干个词元）。提示通常无需对大型语言模型进行额外微调或训练，并且可在推理阶段使用（通常称为上下文学习）。因此，预训练大型语言模型可视为神经数据库，可以通过一系列提示进行查询，这些提示按领域专用查询语言组织（Beurer-Kellner 等人，2022年）。最近，一种类似的提示技术（Drozdov等，2022年）展现出强大的系统泛化能力，成功解决了一个长期存在的语义解析任务：将自然语言问题转换为 SPARQL 查询。这类查询技术有几个主要缺点：（1）上下文窗口有限（通常少于 8192 个词元），无法通过提示提供整个数据库内容；（2）生成的响应存在事实准确性问题，且容易产生幻觉。

查询规划器 在神经图数据库中，输入查询预期以结构化形式给出，例如，形式可以通用到一阶逻辑——我们不讨论可能的查询语言，因为这超出了本工作的范围；本工作着重强调通用语言和特定领域语言的广度。不过，为了说明示例查询，我们采用 SPARQL 表示法，它是图数据库的一种标准查询语言。总体而言，符号查询规划与神经查询规划的主要区别如图 8所示。传统图数据库是符号式和演绎式的，而神经图数据库是神经式和归纳式的（即，能够泛化到未见数据）。由于我们假设图数据是不完整的，神经推理预计需要处理更多中间结果，因此，查询规划对于推导出最高效的计划就更加重要。

查询规划器的任务是根据查询复杂度、预期预测任务和存储配置，推导出最高效的查询计划（即原子操作和逻辑操作的执行顺序），从而优化执行过程。我们认为，对于无法完全存储在主存中的超大型图，存储配置将发挥重要作用。典型的配置会采用

分布式地将图和潜在表示划分到多个设备上。因此，规划器必须确定将输入查询的哪些部分发送到相应分区。在分布式环境中训练和推理机器学习模型（可能受到隐私限制）是联邦学习 (Li 等人， 2020年）和 差分隐私 (Dwork 等人， 2006的核心；我们认为，它们将成为超大型神经图数据库的重要组成部分。

用于评估计划质量的常见指标是执行时间（以时间为单位）和吞吐量（单位时间内的查询数量）。在传统图数据库中，不同的查询计划会返回相同的答案，但所需的执行时间不同（Endris 等人，2018年），这是由于连接算子的顺序（例如，嵌套循环连接或哈希连接）、索引调用次数以及需要处理的中间结果数量不同所致。相比之下，神经图数据库在潜在空间中结合链接预测执行查询，并且根据所采用的方法，原子操作（关系投影和逻辑运算符）的时间复杂度可能取决于隐藏维度 $\mathcal{O}(d)$ 或实体数量 $\mathcal{O}(\mathcal{E})$ （我们将在第 5.4 节详细讨论复杂度）。

截至目前，对于神经查询回答方法而言，查询规划仍然是一项挑战，因为所有现有方法（第 5 节）都会执行一个预先给定的算子序列，而所有现有数据集（第 7 节）都提供了这样一个未经优化的硬编码序列。此外，一些现有方法，例如，CQD（Arakelyan 等人，2021年）容易受到这样一个问题的影响：改变查询计划可能会导致不同的答案集。

我们假设，考虑缺失边和不完整图来推导高效查询计划的原则性方法，可以被视为神经程序合成(Nye 等人，2020年)，这种方法常用于推导查询计划，以帮助大型语言模型解决复杂数值推理任务（Gupta 等人，2020年；Chen 等人，2020年）。在复杂查询回答领域，潜在执行引导推理（LEGO）框架首次尝试应用神经程序合成来生成查询计划（Ren 等人，2021年）。LEGO 通过从关系投影和逻辑运算符构成的空间中采样，迭代扩展初始查询树的根节点。采样依据可学习启发式方法和剪枝机制进行。

查询执行器查询计划确定后，查询执行器模块会将查询（或其部分）编码到潜在空间，与图存储及其检索模块进行交互，并将中间结果聚合为最终答案集。根据定义 2.18，存在简单答案和困难答案。查询执行器应返回完整的简单答案集，并恢复缺失边，以返回缺失的困难答案。

第 5 节介绍了神经查询执行的两种常见机制：（1）原子机制，类似于传统数据库：按顺序执行查询计划，先编码原子模式（例如关系投影），检索其答案，再将逻辑运算符作为中间步骤执行；（2）全局机制：将完整查询图编码到潜在空间中，并在一步内完成执行。例如（图 7），给定查询计划 $q = U_?.\exists V$：$\text{win}(\texttt{TuringAward}, V) \wedge \text{field}(\texttt{DeepLearning}, V) \wedge \text{university}(V, U_?)$，原子机制会分别执行关系投影，例如按顺序执行 $\text{win}(\texttt{TuringAward}, V)$；而全局机制则会编码完整查询图，并在图的潜在空间中对其进行探测。

所选机制会直接影响计算复杂度（第 5.4 节）以及所支持的逻辑运算符（第 6 节），即，完全基于潜在空间的机制大多仅限于合取查询和交集查询，并且泛化能力较差。迄今为止，大多数方法都遵循原子机制。

神经查询执行的主要挑战在于，使查询表达能力与 SPARQL 或 Cypher 等符号语言的表达能力相匹配。该挑战包含三个方面：(1) 处理表达能力更强的查询算子，例如 `FILTER` 或聚合操作，例如 `COUNT` 或 `SUM`；(2) 支持比树更复杂的查询模式；(3) 支持多个投影变量及其上的操作。我们将在 第 6 节中详细讨论。

查询推理方法分类体系 在接下来的章节中，我们将构建一个查询回答方法分类体系，作为 神经查询引擎（*NQE*）的组成部分。我们从三个主要方向对现有及未来的方法进行分类：**(i) 图** ——回答查询所依据的底层结构是什么；**(ii)建模** ——如何回答查询，以及采用了哪些归纳偏置；**(iii)查询** ——我们要回答什么、查询结构是什么，以及预期答案是什么。该分类体系见 图 9。在接下来的章节中，我们将更详细地介绍每个方向，并通过示例加以说明，涵盖现有复杂查询回答领域的全部文献（40 多篇论文）。

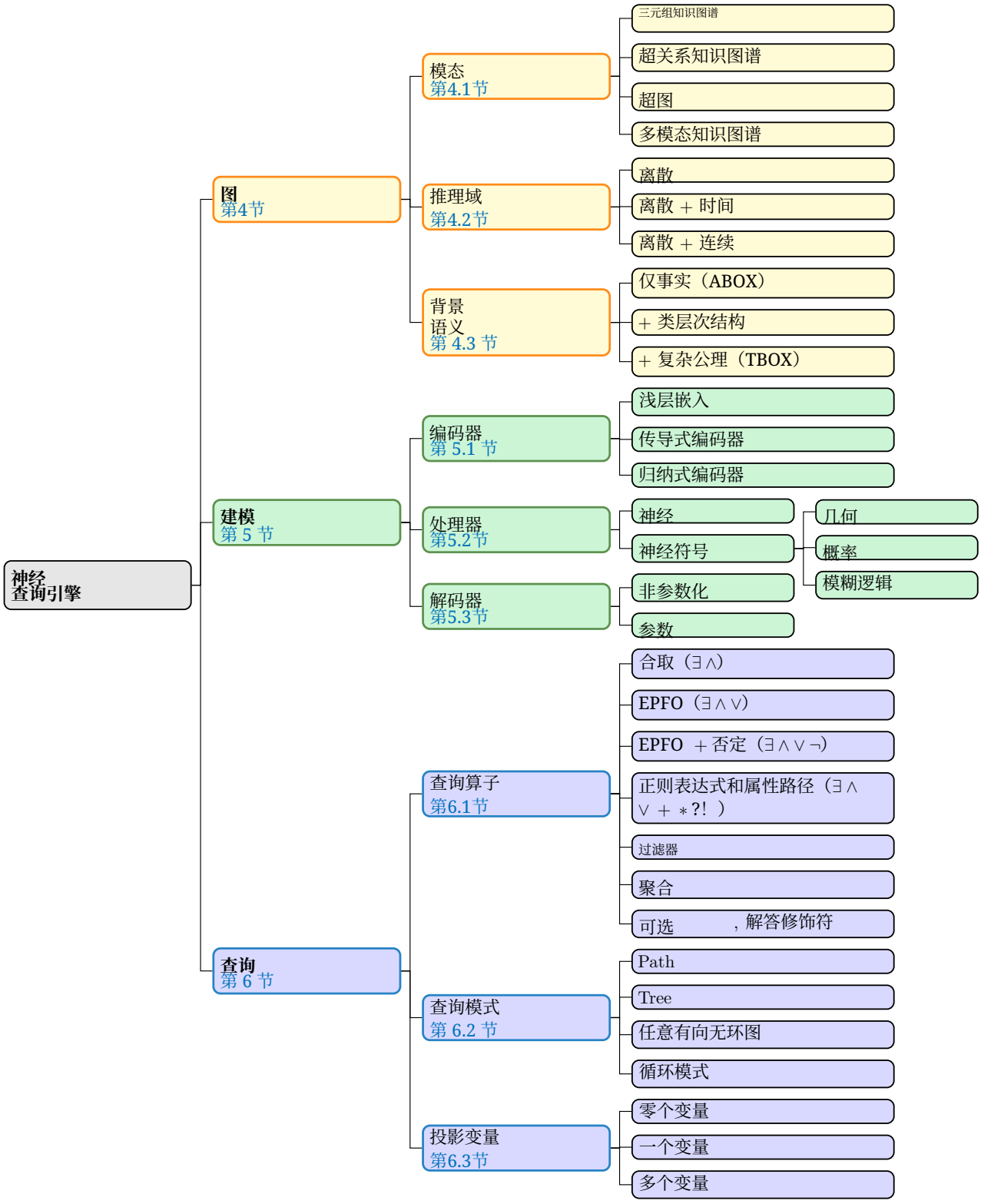


图 9：神经查询引擎分类体系由三个主要分支组成——图、建模和查询。我们将在相关章节中更详细地介绍每个分支，并列举重要示例。

4 图

该图类别涵盖了底层图结构（$\mathcal{G}$ 见定义2.9），复杂查询将针对该结构发出并生成答案。理解图、其内容及建模范式对于设计查询回答模型至关重要。为此，我们从三个方面分析底层图： 模态、推理域和背景语义。

4.1 模态

我们重点介绍知识图谱和图数据库中常见的四种模态：遵循**RDF 数据模型的基于三元组的知识图谱**、遵循 RDF-star 或带标签属性图（LPG）格式的**超关系知识图谱**，以及**超图知识图谱**（我们将在第 3.1 节中详细说明如何选择合适的建模范式）。这三种模态之间的差异如图2所示。此外，我们还概述了**多模态知识图谱**，其不仅包含由节点和边组成的图，还包含文本、图像、音频、视频以及其他数据格式，这些数据格式以显式或隐式方式与底层图相连接。

我们从模态这一方面对相关文献进行了分类，见表 2。迄今为止，大多数查询回答方法仅适用于**基于三元组的**图。在支持**超关系**图的方法中，我们目前只了解到 StarQE（Alivanistos 等人，2022年），它在标记边上引入了实体-关系限定符；其扩展模型 NQE（Luo 等，2023年）也支持此特性。我们认为，超关系模型或可作为时序查询回答方法的理论基础，因为时间属性实际上是连续的键值边属性。迄今为止，我们尚未发现支持**超图**或**多模态**图的复杂查询回答模型。我们认为，这些方向可能成为该领域未来的研究方向。

表 2：按模态分类的复杂查询回答方法。

仅三元组	超关系	超图	多模态
GQE (Hamilton 等人, 2018年),带哈希的 GQE (Wang 等人,2019年), CGA (Mai 等人, 2019年), TractOR (Friedman& Broeck, 2020年), Query2Box (Ren 等人, 2020年), BetaE (Ren 与 Leskovec, 2020年), EmQL (Sun 等人, 2020年), MPQE (Daza & Cochez, 2020年), Shv (Gebhart 等人, 2021年) ,Q2B Onto (Andresel 等人, 2021年), RotatE-Box (Adlakha 等人, 2021年), BiQE (Kotnis 等人, 2021年), HyPE (Choudhary 等人, 2021b) ,NewLook (Liu 等人, 2021年) ÇQD (Arakelyan 等人, 2021年), PERM (Choudhary 等人, 2021a), ConE (Zhang 等人, 2021b), LogicE (Luus 等人, 2021年), MLPMix (Amayuelas 等人, 2022年), FuzzQE (Chen 等人, 2022b), GNN-QE (Zhu 等人, 2022年), GNNQ (Pflueger 等人, 2022年), SMORE (Ren 等人, 2022年), KGTrans (Liu 等人, 2022年), LinE (Huang 等, 2022b), Query2Particles (Bai 等人, 2022a), TAR (Tang 等人, 2022), TeMP (Hu 等人, 2022), FLEX (Lin 等人, 2022b), TFLEX (Lin 等人, 2022a), NodePiece-QE (Galkin 等人, 2022b), ENeSy (Xu 等, 2022), GammaE (Yang 等, 2022a), NMP-QEM (Long 等人, 2022), QTO (Bai 等人, 2022b), SignalE (Wang 等人, 2022a), LMPNN (Wang 等人, 2023c), Var2Vec (Wang 等人, 2023a), CQD$^{\mathcal{A}}$ (Arakelyan 等人, 2023年), Query2Geom (Sardina 等人, 2023年), SQE (Bai 等人, 2023年)	StarQE (Alivanistos 等人, 2022年), NQE (Luo 等, 2023年)	None	None

4.2 推理域

根据定义 2.8，查询 Q 包含常量 Con、变量 Var，并将答案作为映射μ返回： $\texttt{Var}_Q \to \texttt{Con}$。推理域是指可能的常量、变量和答案所构成的空间，也就是查询回答模型能够进行推理的范围。我们重点介绍三种常见领域（离散、离散 $+$ 时间、离散 $+$ 连续），在图 10 中对其进行说明，并在表 3 中对现有研究进行分类。后续每个领域都是前述领域的超集，例如，离散 $+$ 连续包含离散和离散 $+$ 时间的能力，并将空间扩展至连续输入和输出。

在离散域中，常量、变量和答案可以是知识图谱中的实体 E 和（或）关系类型 R， $\texttt{Con} \subseteq \text{E} \cup \text{R}$, $\texttt{Var} \subseteq \text{E} \cup \text{R}$, $\mu \subseteq \text{E} \cup \text{R}$。也就是说，输入查询只能包含实体（或关系），其答案也只能是实体（或关系）。例如，图 10 中的查询 $?x:\ \texttt{education}(\texttt{Hinton}, x)$ 只能返回两个实体 $\{\texttt{Edinburgh}, \texttt{Cambridge}\}$ 作为答案。从概念上讲，该框架也允许将关系类型 $r \in \text{R}$ 设为变量，例如用于描述 SPARQL 图模式

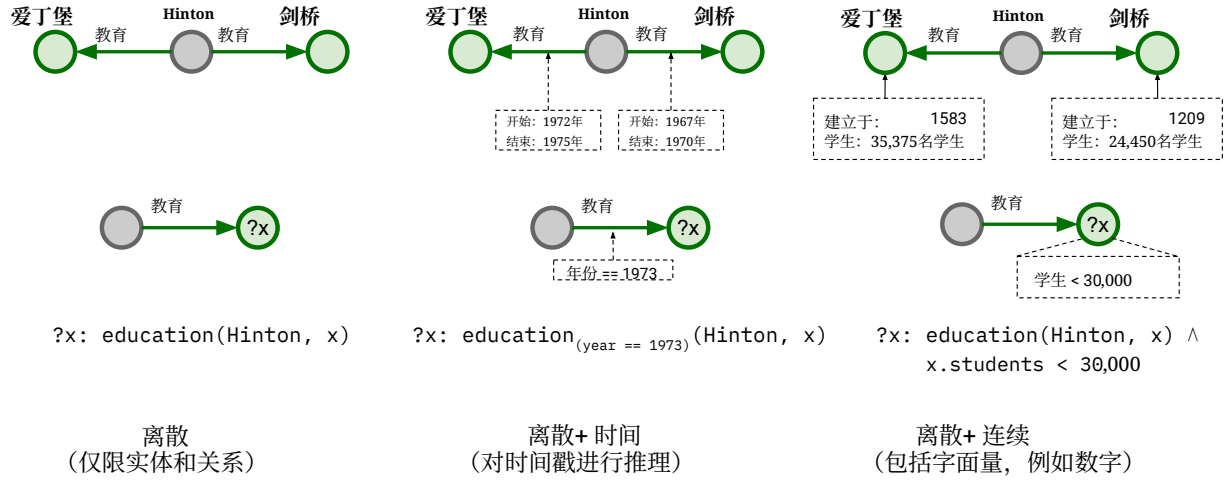


图10：推理域。离散 领域只允许将实体和关系作为常量、变量集合和答案。离散 $+$ 时间 领域将空间扩展到离散时间戳和特定时间算子。离散 $+$ 连续 领域允许连续输入（字面量）和输出。

{Hinton ?r Cambridge} ——或 $?r:\ r(\text{Hinton}, \text{Cambridge})$ 的函数形式，该形式会返回两个节点之间的所有关系类型 Hinton 和 Cambridge。然而，据我们所知，现有的大多数查询回答文献和数据集都将常量、变量和答案的范围限制为仅实体， $\text{Con} \subseteq \mathcal{E}$， $\text{Var} \subseteq \mathcal{E}$， $\mu \subseteq \mathcal{E}$，且所有关系类型都预先给定（我们将在第 6 节中更详细地讨论查询结构）。迄今为止，该领域的大多数文献都属于离散推理域（表 3）。

某些节点和边可能带有 时间戳，这些时间戳来自一组离散时间戳 $t \in \mathcal{TS}$，用于指示某个陈述的有效期。在更一般的情况下，某些子图也可能带有时间戳。当查询包含时序数据时，我们定义离散 $+$ 时间 领域。在该领域中，常量集合扩展为包含时间戳 $\text{Con} \subseteq \mathcal{E} \cup \mathcal{R} \cup \mathcal{TS}$，关系投影也可能在特定时间戳下实例化$R_t(a, b)$。例如（图 10），给定一个带时间戳的图和查询 $?x: \text{education}_{\text{year}==1973}(\text{Hinton}, x)$，答案集仅包含 Edinburgh，因为时间戳 1973 仅落在一条边的有效期内。

也可以将变量和查询答案的领域扩展为包含时间戳， $\text{Var}, \mu \subseteq \mathcal{E} \cup \mathcal{R} \cup \mathcal{TS}$，这样查询便可以将时间戳用作中间的存在量化变量，而可能的答案可以是实体或时间戳。Lin 等人提出的时间特征逻辑嵌入框架（TFLEX）采用了这种方法（Lin 等人 (2022a)），并定义了额外的算子 **之前、之后、之间**，用于处理具有离散时间戳的边。

最后，表达能力最强的领域是离散 $+$ 连续领域，它支持对连续输入（例如数字、文本和连续时间戳）进行推理；这些输入通常作为节点和边属性或字面量存在。形式上，对于数值数据，常量、变量和答案的空间扩展为包含实数$\mathbb{R}$，即 $\text{Con}, \text{Var}, \mu \subseteq \text{E} \cup \text{R} \cup \text{T S} \cup \mathbb{R}$。图 10 中的一个示例查询 $?x: \text{education}(\text{Hinton}, x) \wedge x.\text{students} < 30000$ 包含一个合取项 $x.\text{students} < 30000$，要求对变量 x 的 students属性进行数值推理，以生成答案 Cambridge。类似地，将答案集扩展为连续输出可以被表述为一个回归任务。截至目前，我们尚未发现任何能够在连续领域中工作的复杂查询回答方法。不过，鉴于大多数现实世界知识图谱都高度依赖字面量，对连续数据进行推理的能力对于查询回答至关重要。

4.3 背景语义

关系数据库通常包含一个模式，即对表和列如何组织的规范，它提供了数据库内容的高层概览。在图数据库中，模式同样存在，并且

表 3：按推理域分类的复杂查询回答方法。

离散	离散 + 时间	离散 + 连续
GQE (Hamilton 等人,2018年), 带哈希的 GQE (Wang 等人, 2019年), CGA (Mai 等人,2019年), TractOR (Friedman 和 Broeck,2020年), Query2Box (Ren 等人, 2020年), BetaE (Ren 与 Leskovec, 2020年), EmQL (Sun 等人,2020年), MPQE (Daza 和 Cochez, 2020年), Shv (Gebhart 等人, 2021年), Q2B Onto (Andresel 等人, 2021年), RotatE-Box (Adlakha 等人, 2021年), BiQE (Kotnis 等人, 2021年), HyPE (Choudhary 等人, 2021b), NewLook (Liu 等人,2021年), CQD (Arakelyan 等人,2021年), PERM (Choudhary 等人, 2021a), ConE (Zhang 等人,2021b), LogicE (Luus 等人,2021年), MLPMix (Amayuelas 等人,2022年), FuzzQE (Chen 等人, 2022b), GNN-QE (Zhu 等人, 2022年), GNNQ (Pflueger 等人, 2022年), SMORE (Ren 等人, 2022年), KGTrans (Liu 等人, 2022年), LinE (Huang 等, 2022b), Query2Particles (Bai 等人, 2022a), TAR (Tang 等人, 2022年), TeMP (Hu 等人, 2022年), FLEX (Lin 等人, 2022b), NodePiece-QE (Galkin 等人, 2022b), ENeSy (Xu 等, 2022年), GammaE (Yang 等人, 2022a), NMP-QEM (Long 等, 2022), StarQE (Alivanistos 等人, 2022), QTO (Bai 等人, 2022b), SignalE (Wang 等人, 2022a), LMPNN (Wang 等人, 2023c), NQE (Luo 等, 20 23), Var2Vec (Wang 等, 2023a), CQD^{A} (Arakelyan 等人, 2023), Query2Geom (Sardina 等人, 2023), SQE (Bai 等人, 2023)	TFLEX (Lin 等人, 2022a)	None

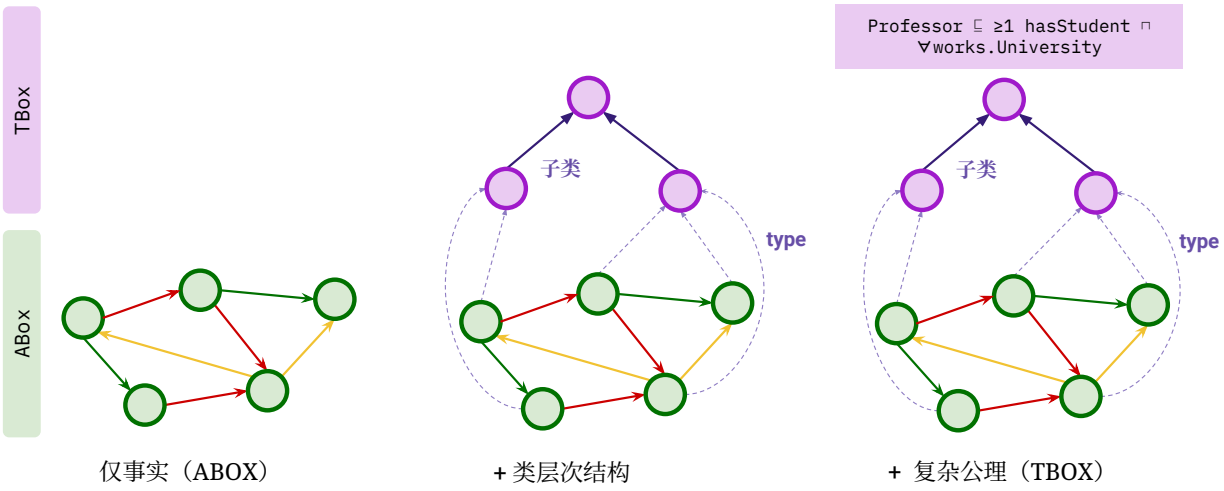


图 11：背景语义。仅事实 图仅包含断言（ABox），没有更高层次的模式。类层次结构 引入节点类型（类）以及类之间的层级关系。最后，复杂公理 添加了更加复杂的逻辑规则（TBox），这些规则受特定 OWL 配置文件的约束，例如，教授是拥有一个或多个学生且工作于大学的人。

例如可以描述带标签属性图（LPG）范式中常见的节点类型。然而，RDF 图可通过采用基于描述逻辑（Baader 等人，2003）的标准，提供额外一层逻辑一致性。由于纳入模式对于设计有效的查询回答机器学习模型至关重要，我们将背景语义（图 11）引入为一种概念，即在纯事实（语句）之上可用的附加模式信息。

仅事实。 在最简单的情况下，不存在背景模式，因此知识图谱仅由语句（事实）组成，也就是说，知识图谱遵循定义 2.2，$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$。从描述逻辑的角度来看，图仅包含 断言 （ABox）。查询根据推理域（第4.2节）的不同，仅涉及实体、关系和字面量。最初的 GQE（Hamilton 等人，2018年）专注于仅事实图，而后续的大多数查询回答方法（表 4）都专门针对无模式知识图谱。

类层次结构。实体类别（或节点类型）是仅事实图的自然扩展，可作为一种基本模式。将定义2.2 扩展为包含一组类型$\mathcal{T}$后，图$\mathcal{G}$定义为$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S}, \mathcal{T})$。每个

实体 e 可能关联一个或多个类类型$(e) = \{t_1, \ldots\ldots, t_k\} | t \in \mathcal{T}$。LPG 和 RDF 图都支持类，不过 RDF 可以使用 RDF Schema（RDFS）指定类之间的层级关系（Brickley 等人，2014年）[5]。形式上，类型成为图中的显式节点，而语句$\mathcal{S}$ 现在可以将类型作为主语或宾语，$\mathcal{S} \subset ((\mathcal{E} \cup \mathcal{T}) \times \mathcal{R} \times (\mathcal{E} \cup \mathcal{T}))$；在其他模态的图中，类型也可以是其他语句元素。实际上，涉及类型的边可能会实际存在于知识图谱中，也可能被视为特定模型的额外监督输入。

迄今为止，我们了解到有三种纳入实体类型的查询回答方法（表 4）。上下文图注意力（CGA，Mai 等人 (2019年）仅将类型用于实体嵌入初始化，并要求每个实体只能有一种类型。查询不以实体类型为条件，答案集仍然只包含实体。也就是说，查询常量、变量和答案集均遵循$\texttt{Con} \subseteq \mathcal{E}$ $\texttt{Var} \subseteq \mathcal{E}$ $\mu \subseteq \mathcal{E}$。

在类型感知消息传递（TEMP，Hu 等人 (2022年）中，类型嵌入用于丰富实体和关系表示，随后将这些表示传递给下游查询回答模型。每个节点可能有多个类型。在归纳场景中（我们将在第 5.1 节中详细说明推理场景），对于推理时的未见节点，类型嵌入和关系嵌入被学习为不变量，以便在训练实体和推理实体之间迁移。就查询而言，常量、变量和答案集仍然仅限于实体。

TBox 与 TBOX 和 ABOX 神经推理器（TAR，Tang 等人（2022年））将类型及其层次结构纳入考量，以改进对实体的预测，并引入了预测答案实体类型的任务，即 $\mu \subseteq$ （$\mathcal{E} \cup \mathcal{T}$）。常量和变量仍然仅限于实体，因此不允许将类型用作查询变量。除了原有的实体预测之外，TAR 还通过三种辅助损失利用类层次结构，即 概念检索 ——预测类型的答案集，包含关系预测 ——预测哪种类型是另一种类型的子类，以及 实例化 ——为每个实体预测一种类型。

对于类层次结构 这一类方法，自然的下一步是将类型以常量和变量的形式纳入查询，$\texttt{Con} \subseteq (\mathcal{E} \cup \mathcal{T})$ $\texttt{Var} \subseteq (\mathcal{E} \cup \mathcal{T})$。

复杂公理。最后，模式可能不仅包含类层次结构，还包含一组更复杂的公理，例如涉及关系层次结构、关系上的限定约束或复合类。如今，这种复杂模式可以视为扩展图 G =(E, R, S, O) 定义的本体 O。在图 11 中，公理 $\texttt{Professor}\,\mathrm{v} \geq 1$ 有学生 u ∀工作于.University 描述了这样一个事实：教授是拥有一名或多名学生并在大学工作的人员。用描述逻辑的术语来说，图还包含一个额外的术语组件（TBox）。TBox 的表达能力直接影响符号推理引擎的复杂度；对于大多数表达能力最强的片段，复杂度可达指数时间（ExpTime）。为缓解这一问题，本体语言（如 Web 本体语言 OWL）规定了表达能力较弱但计算上可行的配置文件，例如，OWL 2 引入了三个配置文件：OWL EL、OWL QL 和 OWL RL（Motik 等人，2009年）。反过来，OWL-QL 与 DL-Lite$_\mathcal{R}$ 本体语言存在较为宽松的联系（Artale 等人，2009年）。

在图表示学习中，即使对于简单的链接预测模型，纳入复杂的本体公理也并非易事（Zhang 等人，2022b）。在查询回答文献中，Andresel 等人（2021年）首次尝试纳入复杂公理。在 Q2B Onto（O2B）中，即 Query2Box（Q2B，Ren 等人（2020年）的扩展版本，所考虑的复杂公理属于$DL\text{-}Lite_\mathcal{R}$ 片段，支持类的层次结构（子类）、关系的层次结构（子属性），以及关系的值域和领域。模型架构不直接以公理为条件，因此仍保持 Query2Box 的原始架构。相反，公理会影响图结构、查询采样和辅助损失；具体而言，系统使用查询重写机制，为原始查询补充更多答案，使其效果如同在完整图上执行（演绎闭包），类似于数据增强。在优化过程中，辅助正则化损失旨在将一个专门的查询框 q 纳入该查询更一般的版本中 q'。

不过，即使是在 O2B 的查询采样中纳入复杂公理这一昂贵过程，主要受益的仍然是演绎查询回答能力，即推断出图$\mathcal{G}$ 和本体 $\mathcal{O}$已经蕴含的答案，泛化能力并不会因此提升，尤其是在缺失边无法通过本体公理推断出来时。我们将在演绎、泛化以及其他设置将在第 7 节中进行详细阐述。

[5]RDFS 具有更强的表达能力（例如，关系层次结构），但我们将其留给<style id='4'>复杂公理

未来工作的另一个方向，是更好地理解图神经网络（GNN）编码器应用于多关系知识图谱时的理论表达能力。关于非关系图的早期工作（Barceló 等，2020）将其表达能力对应到带有两个变量和计数量词的 FOC_2 一阶逻辑子集，以及对应到最大元数为 B 的超图的 FOC_B （Luo 等，2022）。在关系图中，Barceló 等（2022）通过关系 Weisfeiler-Leman（RWL）检验对关系图神经网络的表达能力进行了量化，证明 RWL 比经典 WL 检验（Weisfeiler 与 Leman，1968）具有更强的表达能力，并且常见的关系图神经网络架构，如 R-GCN（Schlichtkrull 等，2018）和 CompGCN（Vashishth 等，2020），其能力均受限于 1-RWL。Huang 等（2023）利用 RWL 推导出，以查询节点为条件的图神经网络族（例如神经 Bellman-Ford 网络（Zhu 等，2021））受限于非对称局部 2-RWL，并且具有 $\mathrm{rGFO}^3_{\mathrm{cnt}}$ 的表达能力，即受限的、包含三个变量和计数量词的守卫一阶逻辑片段。与此同时，Gao 等（2023）将知识图谱视为双重置换等变结构（分别对节点和边类型进行置换），并将其表达能力对应到全称量化的实体-关系（UQER）Horn 子句。然而，图神经网络能否捕获类 OWL 公理，并将其作为复杂查询回答中的归纳偏置加以利用，仍是一个悬而未决的问题。

Tab表 4：按背景语义分类的复杂查询回答方法 .

仅事实（ABOX）	+ 类层次结构	+复杂公理（TBOX）
GQE (Hamilton 等人，2018年)，带哈希的 GQE (Wang 等人，2019年)，TractOR (Friedman 和 Broeck，2020年)，Query2Box (Ren 等人，2020年)，BetaE (Ren 与 Leskovec，2020年)，EmQL (Sun 等人,2020年)，Shv (Gebhart 等人，2021年)，RotatE-Box (Adlakha 等人，2021年)，MPQE (Daza 和 Cochez，2020年)，BiQE (Kotnis 等人，2021年)，HyPE (Choudhary 等人,2021b)，NewLook (Liu 等人,2021年)，CQD (Arakelyan 等人，2021年)，PERM (Choudhary 等人，2021a)，ConE (Zhang 等人，2021b)，LogicE (Luus 等人，2021年)，MLPMix (Amayuelas 等人，2022年)，FuzzQE (Chen 等人，2022b)，GNN-QE (Zhu 等人，2022年)，GNNQ (Pflueger 等人，2022年)，SMORE (Ren 等人，2022年)，KGTrans (Liu 等人，2022年)，LinE (Huang 等，2022b)，Query2Particles (Bai 等人，2022a)，FLEX (Lin 等人，2022b)，TFLEX (Lin 等人，2022a)，NodePiece-QE (Galkin 等人，2022b)，ENeSy (Xu 等，2022年)，GammaE (Yang 等，2022a)，NMP-QEM (Long 等<;style id='3'>，2022年)，StarQE (Alivanistos 等，2022年)，QTO (Bai 等人，2022b)，SignalE (Wang 等人，2022a)、LMPNN (Wang 等人，2023c)、NQE (Luo 等，2023年)、Var2Vec (Wang 等人，2023a)、$\mathrm{CQD}^{\mathcal{A}}$ (Arakelyan 等人，2023年)、Query2Geom (Sardina 等人，2023年)、SQE (Bai 等人，2023年)	CGA (Mai 等 al.，2019年)，TeMP (Hu 等人、 2022年)，TAR (Tang 等人,2022年)	Q2B Onto (Andresel 等人，2021年)

5 建模

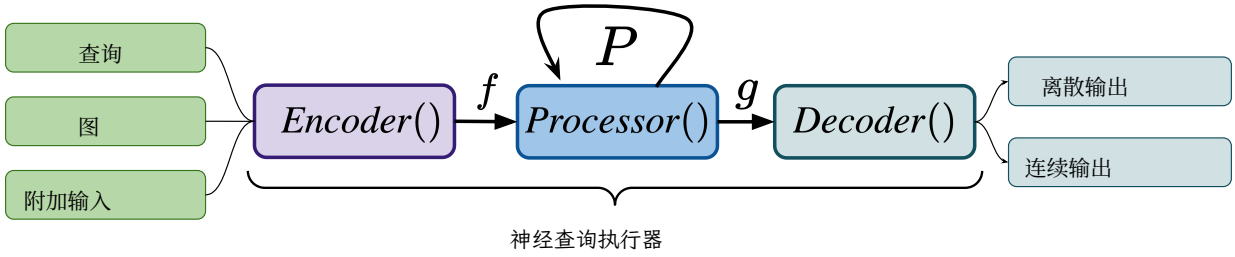


图12：通过神经查询执行 的编码器-处理器-解码器 模块。编码器函数 f构建输入（查询、目标图、辅助数据）在潜在空间中的表示。处理器 P在其他输入的条件约束下，使用逻辑运算符对图执行查询。解码器函数 g 构建所请求的输出，这些输出可以是离散的，也可以是连续的。

本节将从建模的角度探讨相关文献。按照常见的方法论（Battaglia 等人，2018年），我们从建模方法的编码器-处理器-解码器

模块（如图 12所示）。(1) 编码器 $\textsc{Enc}()$ 接收输入查询 q、目标图 $\mathcal{G}$ 及其实体和关系，以及辅助输入（例如节点、边、图特征），以在潜在空间中构建它们的表示。(2) 处理器 P 利用所选的归纳偏置，在潜在空间或符号空间中处理带有其逻辑运算符的查询表示。(3) 解码器 $\textsc{Dec}()$ 接收处理后的潜变量，并构建所需的输出，例如离散实体上的分布，或连续任务情况下的回归预测。通常，编码器、处理器和解码器可以使用神经网络 θ 进行参数化，也可以是非参数化的。最后，我们分析现有处理器的计算复杂度。

5.1 编码器

我们从编码器开始介绍建模部分，即，介绍不同方法如何对知识图谱中的实体和关系进行编码和表示。编码方法分为三类： 浅层嵌入、传导式编码器和归纳式编码器，每一类都代表了生成实体/关系神经表示的一种不同方式。不同的编码方法适用于不同的推理设置（详见第7.4节），并且可能需要采用不同的逻辑运算符方法（详见第5.2节）。图13展示了三种常见的编码方法。

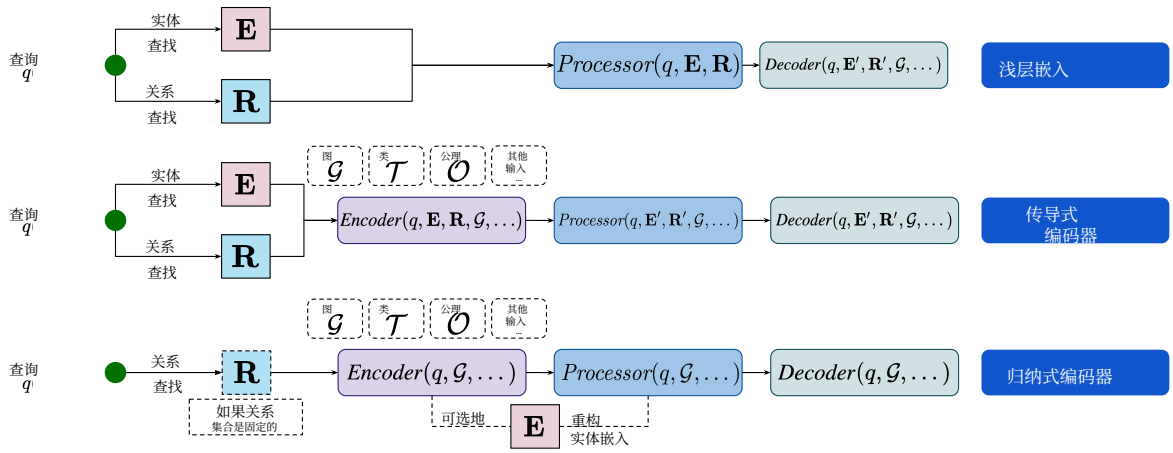


图13：编码器的分类。浅层编码器通过查找实体和关系嵌入，并将其传递给处理器。传导式编码器还会利用查询、图、类或其他潜变量来丰富表示。归纳式编码器不需要可学习的实体嵌入。

表5：按编码器分类的复杂查询求答方法。

浅层嵌入	传导式编码器	归纳式编码器
GQE (Hamilton 等人，2018年)，带哈希的 GQE (Wang 等人，2019年)，CGA (Mai 等，2019年)，TractOR (Friedman 与 Broeck，2020年)，Query2Box (Ren 等人，2020年) BetaE (Ren 与 Leskovec，2020年)，EmQL (Sun 等人，2020年)，Shv (Gebhart 等人，2021年)，Q2B Onto (Andresel 等人，2021年)，RotatE-Box (Adlakha 等人，2021年) HyPE (Choudhary 等人，2021b)，NewLook (Liu 等人，2021年)，CQD (Arakelyan 等人，2021年)，PERM (Choudhary 等人，2021a)，ConE (Zhang 等人，2021b)，LogicE (Luus 等人，2021年)，FuzzQE (Chen 等人，2022b)，SMORE (Ren 等人，2022年)，LinE (Huang 等，2022b)，Query2Particles (Bai 等人，2022a)，TAR (Tang 等人，2022年)，FLEX (Lin 等人，2022b)，TFLEX (Lin 等人，2022a)，GammaE (Yang 等人，2022a)，NMP-QEM (Long 等人，2022年)，QTO (Bai 等人，2022b)，SignalE (Wang 等人，2022a)，Var2Vec (Wang 等人，2023a)，$\text{CQD}^{\mathcal{A}}$ (Arakelyan 等人，2023年)，Query2Geom (Sardina 等人，2023年)	MPQE (Daza & Cochez，2020年)，BiQE (Kotnis 等人，2021年)，KGTrans (Liu 等人，2022年)，StarQE (Alivanistos et al.，2022年)，MLPMix (Amayuelas et al.，2022年)，ENeSy (Xu 等，2022年)，LMPNN (Wang et al.，2023c)，NQE (Luo 等，2023年)，SQE (Bai 等人，2023年)	NodePiece-QE (Galkin 等人，2022b)，GNN-QE (Zhu 等人，2022年)，GNNQ (Pflueger 等人,2022年)，TeMP (Hu 等人，2022年)

浅层嵌入。 第一类方法将图中的每个实体/关系编码为低维向量，从而得到实体嵌入矩阵 $\boldsymbol{E}$ 和关系嵌入矩阵 $\boldsymbol{R}$。

实体嵌入矩阵的形状为 $|\mathbf{E}| \times d$ ($|\mathbf{R}| \times d$)，其中 d 是嵌入的维度。浅层嵌入方法假设图中所有节点的表示彼此独立。这一独立性假设赋予模型很大的自由度，留下许多自由参数供其学习。这种建模源自知识图谱补全文献，其思路是通过优化图中所有边上的预定义距离/评分函数来学习实体和关系嵌入矩阵，例如三元组事实 $\texttt{dist}(\mathbf{e}_s, \mathbf{r}, \mathbf{e}_o)$。大多数查询回答文献遵循相同范式，采用各种不同的嵌入空间和距离函数来学习实体和关系嵌入矩阵。目前已有多种嵌入空间被提出。例如，GQE（Hamilton 等，2018）和 Query2Box（Ren 等，2020）将实体嵌入到 $\mathbb{R}^d$ （欧几里得空间中的点向量）；FuzzQE（Chen 等，2022b）将实体嵌入到值域为 $[0, 1]$ 的实数空间（模糊逻辑评分）；BetaE（Ren 与 Leskovec，2020）使用 Beta分布这一概率嵌入空间；而 ConE Zhang 等（2021b）则将实体嵌入为单位圆上的一个点。每种设计选择都为执行逻辑运算符提供了归纳偏置（详见第5.2节）。有些方法直接采用已在简单链接预测任务上预训练的浅层实体和关系嵌入，并在其基础上应用带有非参数逻辑运算符的查询回答解码器。例如，CQD（Arakelyan 等，2021）、LMPNN（Wang 等，2023c）、Var2Vec（Wang 等，2023a）和 CQD$^{\mathcal{A}}$ （Arakelyan 等，2023）采用复数空间 $\mathbb{C}^d$ 中的预训练嵌入，并分别使用非参数 t-范数和 t-余范数来建模交集与并集。QTO（Bai 等，2022b）则更进一步，在预处理阶段根据预训练的实体和关系嵌入，将所有可能三元组的分数完全物化到一个 $[0, 1]^{|\mathcal{R}|\times|\mathcal{E}|\times|\mathcal{E}|}$ 矩阵中。

尽管浅层方法是主流的设计选择，但其缺点在于：(1) 由于所有实体/关系的参数都是从头学习的自由参数，浅层嵌入不使用任何关于实体或其邻近结构的归纳偏置和先验知识；(2) 它们不适用于归纳式推理设置，因为这些方法在设计上并不具备那些未见的新实体的表示/嵌入。一种可能的解决方案是为新实体随机初始化一个嵌入向量，并通过采样涉及该新实体的查询来微调该嵌入向量（详见第7.3节）。然而，这种解决方案需要在推理期间进行梯度步，因此并不理想。

传导式编码器。与浅层嵌入方法类似，传导式编码器方法学习相同的实体嵌入矩阵 $\mathbf{E}$。此外，它们在查询 q、实体嵌入矩阵和关系嵌入矩阵（以及可选的其他可用输入）之上学习一个额外的编码器 $\text{Enc}_\theta(q, \boldsymbol{E}, \boldsymbol{R}, \dots)$（由 θ 参数化）。其目标是将编码器应用于查询 q 中实体的嵌入，以捕捉图中相邻实体之间的依赖关系。具体而言，该额外的编码器可以将特征矩阵的若干行作为输入，并进一步施加变换。例如，BiQE（Kotnis 等，2021）和 kgTransformer（Liu 等，2022）将查询图 $\mathbf{G}_q$ 线性化为一个序列，并应用一个 Transformer（Vaswani 等，2017）编码器，该编码器关注查询中的所有其他嵌入，并将 $\texttt{[MASK]}$ 标记的最终表示作为目标查询。MPQE（Daza 与 Cochez，2020）和 StarQE（Ali-vanistos 等，2022）在查询图 $\mathbf{G}_q$ 之上运行消息传递图神经网络架构，以丰富实体嵌入和关系嵌入，并提取最终节点表示作为查询嵌入。这些方法与浅层嵌入具有相似的优点和缺点。也就是说，该方法中有许多需要训练的自由参数。不同于浅层嵌入，额外的编码器利用实体与其邻近实体或查询中其他实体之间的关系归纳偏置，从而获得更好的学习到的实体表示和泛化能力。然而，由于该方法的核心仍然基于大型实体嵌入查找矩阵，因此它仍然具有同样的缺点：所有此类方法都无法直接应用于可能观察到新实体的归纳式设置。

归纳式编码器。为了解决浅层嵌入和传导式编码器面临的上述挑战，归纳式编码器方法旨在避免为固定数量的实体学习嵌入矩阵 $\boldsymbol{E}$ 。相反，归纳式表示通常通过利用某些不变性来计算，也就是在推理时将表示迁移到包含新实体的不同图上时仍保持不变的特征。正如我们在第 7.4 节中所述，归纳式编码器可能采用不同的不变性，尽管大多数归纳式编码器都依赖于关系类型集合 $\mathbf{R}$ 固定这一假设。形式上，遵循定义 2.7，给定由实体项和关系项组成的复杂查询 $\mathbf{Q} = (\mathbf{E}', \mathbf{R}', \mathbf{S}', \mathbf{S}')$，其中实体项和关系项又包含常量 $\texttt{Con}$ 和变量 $\texttt{Var}$；关系投影 $R(a, b) \in \mathbf{S}$（以及可选的

$\bar{\mathcal{S}}'$）、目标图 $\mathcal{G}$ （以及可选的其他输入），归纳式编码器会为每个实体学习一个条件表示函数 $\text{ENC}_\theta(e|\mathcal{E}', \mathcal{R}', \mathcal{G}\ldots)$ 。 $e \in \mathcal{E}$。Galkin 等人 (2022b) 提出了两类归纳式表示，即，（1）归纳式节点表示和（2）归纳式关系结构表示。

归纳式**节点表示**方法将ENC_θ 参数化为固定大小、不变词汇表的函数。例如，NodePiece-QE（Galkin 等人，2022b）采用关系类型这一不变词汇表，并通过每个实体的一组关联关系对其进行参数化。TeMP（Hu 等人，2022年）采用实体类型和类层次结构这一不变词汇表，并将它们的表示注入实体表示中。归纳式节点表示方法会重建新实体的嵌入，可用作与任意处理器方法配套的浅层查找表的直接替代方案，例如，NodePiece-QE 使用 CQD 作为处理器，而 TeMP 则搭配 GQE、Query2Box、BetaE 和 LogicE 处理器进行了测试。

归纳式**关系结构**表示方法将ENC_θ 参数化为相对关系结构的函数，该结构只需学习关系嵌入，并以关系作为不变量。这类方法通常采用各种标记技巧（Zhang 等人，2021a）来标记输入查询$\mathcal{Q}$ 中的常量（锚实体），使得经过消息传递过程后，所有其他节点都能编码相对于起始节点的图结构。具体而言，GNN-QE（Zhu 等人，2022年）用所查询关系的嵌入向量标记锚节点，例如，对于投影查询（h，r，？）节点 h 将以关系 r 的嵌入初始化，而所有其他节点均以零向量初始化。这样，GNN-QE 只需学习关系嵌入 $\boldsymbol{R}$ 和图神经网络权重。GNNQ（Pflueger 等人，2022年）将查询及其变量和关系表示为超图，并通过对超边应用图卷积来学习关系结构。超边以参与关系的多热特征向量进行参数化，因此唯一可学习的参数是图神经网络权重。

不过，归纳式模型仍存在一系列开放问题。由于大多数归纳式方法依赖于学习关系嵌入，因此难以应用于这样一种设置：知识图谱在推理时会更新，加入新的、未见过的关系类型，也就是说，关系不具备不变性。这一事实可能需要探索新的不变性和特征化策略（Huang等，2022a；Gao 等，2023；Chen等，2023）。与浅层模型相比，归纳式模型在训练时需要更多时间和内存，而且目前还难以扩展到大规模图。我们推测，归纳式编码器将成为未来 CLQA 研究的重点，因为在推理时无需重新训练即可泛化到未见实体和图，对于神经图数据库的可更新性至关重要。此外，可更新性可能会提升持续学习的作用（Thrun，1995；Ring，1998），并加剧灾难性遗忘（McCloskey & Cohen，1989）的负面影响，而编码器必须应对这一问题。更大的推理图也带来了一个重要的规模泛化问题（Yehudai 等人，2021；Buffelli 等人，2022；Zhou 等人，2022b）：在小图上训练的图神经网络，在大得多的图上进行推理时，性能会下降。Galkin 等人（2022b）在归纳式复杂查询回答设置中观察到了这一现象。

5.2 处理器

在对查询及其他可用输入进行编码后，处理器 P 在输入图上，于潜在空间（或$\bar{}$符号空间）中执行查询。回顾一下，查询 q定义为 q(E′,R′, S, S)，其中 E′和 R′$\bar{}$项包括常量 `Con` 和变量 `Var`，S 和 S 中的语句包括关系投影 R(a, b)，以及作用于变量之上的逻辑运算符 ops。我们将处理器 P 定义为一组模块，这些模块在给定常量 `Con` 和作用于变量之上的逻辑运算符 ops $\subseteq \{\wedge, \vee, \neg, \ldots\}$ 时执行关系投影 $R(a, b)$`Var` （我们将在第6.1节中详细介绍逻辑运算符）。根据这些模块背后所采用的归纳偏置和参数化策略，我们将处理器分为神经处理器和神经符号处理器（表6a）。此外，我们将神经符号处理器进一步分为几何、概率和模糊逻辑处理器（表6b）。请注意，在本节中，我们省略了纯编码器方法，例如 TeMP（Hu 等，2022）和 NodePiece-QE（Galkin 等，2022b）；这些方法可以与任何神经或神经符号处理器配对使用。为了更形式化地描述处理器模型，我们将 e 表示为实体向量，将 r 表示为关系向量，并将 q 表示为查询嵌入，后者通常是 e 和 r 的函数。我们使用 G_q 表示查询图。

神经处理器。 神经处理器直接在潜在空间中执行关系投影和逻辑运算符 $\mathbb{R}^d$，并使用神经网络对其进行参数化。迄今为止，大多数现有的纯神经方法仅专门处理查询图$\mathcal{G}_q$，只在单个查询中执行算子，并且不进行条件

表6：查询处理器的分类。 表6a 概述了神经、符号、神经符号方法，以及可与任何处理器配对的编码器。表6b 进一步细分了神经符号处理器。

（a）按处理器类型分类的复杂查询回答方法。

任意处理器	神经	神经符号
TeMP (Hu 等人, 2022年), NodePiece-QE (Galkin 等人, 2022b)	GQE (Hamilton 等人, 2018年), GQE 哈希 (Wang 等人, 2019年), CGA (Mai 等人, 2019年), BiQE (Kotnis 等人, 2021年), MPQE (Daza 和 Cochez, 2020年), StarQE (Alivanistos 等人, 2022年), MLPMix (Amayuelas 等人, 2022年), Query2Particles (Bai 等人, 2022a) ,KGTrans (Liu 等人, 2022年), RotatE-m、DistMult-m, ComplEx-m (Ren 等人, 2022年), GNNQ (Pflueger 等人, 2022年), SignalE (Wang 等人, 2022a), LMPNN (Wang 等人, 2023c), SQE (Bai 等人, 2023年)	表6b

（b）神经符号处理器

几何 [神经符号]	概率 [神经符号]	模糊逻辑 [神经符号]
Query2Box (Ren 等人, 2020年), Query2Onto (Andresel 等人, 2021年) ,RotatE-Box (Adlakha 等人, 2021年), NewLook (Liu 等人, 2021年), 知识层 (Gebhart 等人,2021年), HyPE (Choudhary 等人, 2021b), ConE (Zhang 等人, 2021b), Query2Geom (Sardina 等人, 2023年)	BetaE (Ren 与 Leskovec, 2020年), PERM (Choudhary et al., 2021a), LinE (Huang et al., 2022b), GammaE (Yang 等, 2022a), NMP-QEM (Long 等, 2020年)	EmQL (Sun 等人, 2020年), TractOR (Friedman & Broeck, 2020年), CQD (Arakelyan 等人, 2021年), LogicE (Luus 等人, 2021年), FuzzQE (Chen 等人, 2022b), TAR (Tang 等人, 2022年), FLEX (Lin 等人, 2022b), TFLEX (Lin 等人, 2022a), GNN-QE (Zhu 等人, 2022年), ENeSy (Xu 等, 2022年) ,QTO (Bai 等人, 2022b), NQE (Luo 等, 2023年), Var2Vec (Wang 等人, 2023a), CQD$^{\mathcal{A}}$ (Arakelyan 等人, 2023年)

在完整底层图结构上的执行过程 $\mathcal{G}$。由于查询处理是在神经网络所在的潜在空间中进行的，而并集（$\vee$）和否定（$\neg$并没有明确定义，因此大多数神经处理器只实现关系投影 ($R(a,b)$) 和交集 ($\wedge$) 算子。我们从嵌入空间、关系投影的执行方式、逻辑运算符以及最终的解码距离函数等方面，汇总了神经处理器的特征，见表7。我们在图14中展示了两类神经处理器（顺序执行和联合查询编码）之间的区别。

最初的 GQE（Hamilton 等，2018）是神经处理器的第一个实例。也就是说，查询 q、实体 e 和关系 r 都是 $\mathbb{R}^d$ 中的向量。查询嵌入从常量 C（锚节点 e）的嵌入开始，并通过关系投影和交集逐步细化；也就是说，通常假设初始步骤 0 的查询嵌入等同于锚节点的嵌入，即 $\mathbf{q}^{(0)}$ =e。关系投影在潜在空间中通过平移函数 q +r 执行，而交集则由置换不变的 DeepSet（Zaheer，2017年）神经网络建模。若干后续工作改进了 GQE，使其能够处理哈希二进制向量 $\{+1,-1\}^d$ （Wang 等，2019），或者用自注意力取代 DeepSet，并将基于平移的投影替换为矩阵向量乘积（Mai 等，2019）。最近，Ren 等（2022）提出了 DistMult-m、ComplEx-m 和 RotatE-m，它们是针对复杂查询扩展的简单链接预测模型，受 GQE 启发，通过相应的组合函数执行关系投影，并使用 DeepSet 以及可选的 L2 范数对交集算子进行建模。

另一类工作将神经编码器应用于完整查询图$\mathcal{G}_q$，而不显式执行逻辑运算符。根据查询图的表示方式，这类编码器通常是图神经网络、Transformer 或多层感知机。人们假设，神经编码器能够在优化过程中于潜在空间中隐式捕捉逻辑运算符。例如，MPQE（Daza 和 Cochez，2020年）、StarQE（Alivanistos 等人，2022年）和 LMPNN（Wang 等人，2023c）将查询表示为关系图（StarQE 还可选择使用超关系图），其中每条边都是关系投影，而交集则建模为指向同一变量节点的两个入边投影。所有常量$\mathcal{C}$ 和已知关系类型均由相应的嵌入矩阵初始化。所有查询图中的变量节点都使用同一个可学习的 `[VAR]`特征向量初始化，而所有目标节点都使用同一个 `[TAR]` 向量初始化。随后，查询图被

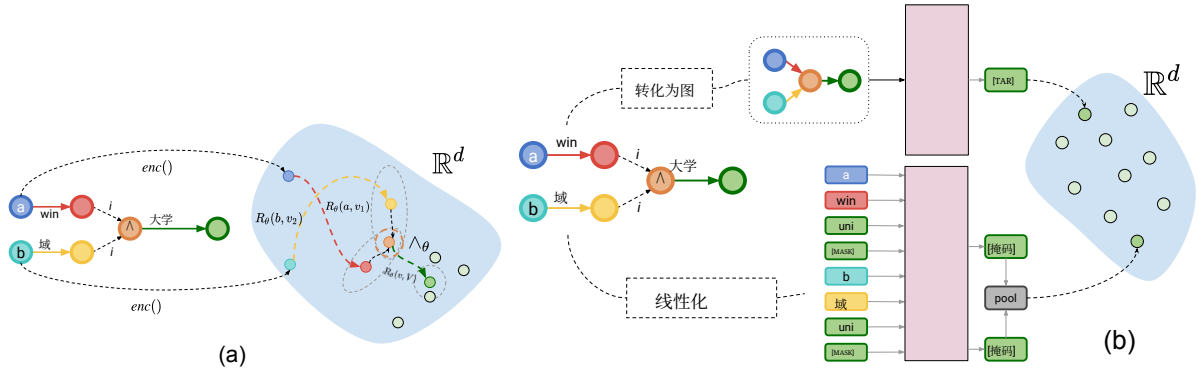


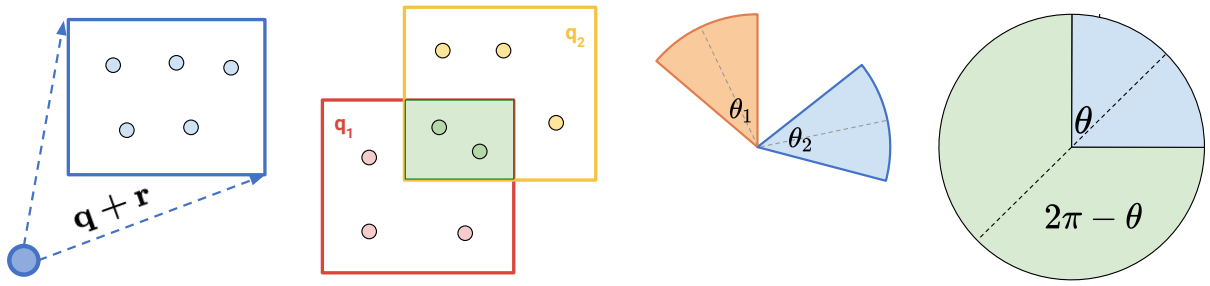
图 14：神经处理器。(a) 关系投影 $R_\theta(a,\ b)$ 和逻辑运算符（非参数化或使用 θ 参数化）在潜在空间中依次执行；(b) 查询被编码为图，或线性化为序列，然后分别通过编码器（图神经网络或 Transformer）。池化后的表示表示查询嵌入。

输入 GNN 编码器（MPQE 使用 R-GCN（Schlichtkrull 等人，2018年），StarQE 使用 StarE（Galkin 等人，2020年），LMPNN 使用 GIN（Xu 等，2019b）），并将 [TAR] 目标节点的最终状态视为可供解码的最终查询嵌入。近期的 LMPNN 在查询图编码中加入了额外的边特征，用以指示给定投影 $R(a,\ b)$ 是否带有否定，并针对 ComplEx 组合函数推导出合并后的投影与否定运算符的闭式解。GNNQ（Pflueger 等人，2022年采取了不同的方法，将查询回答视为子图分类任务。也就是说，输入查询不会直接在给定图$\mathcal{G}$上执行，而是对给定的预计算子图$\mathcal{G}' \subset \mathcal{G}$ 是否满足给定合取查询进行分类。为此，GNNQ 首先使用 Datalog 推导出的三元组扩充该图，并将子图转换为超图，其中只有超边由可学习向量参数化。一方面，这种策略使 GNNQ 具有归纳性，无需学习实体嵌入。另一方面，GNNQ 仅支持合取查询，且尚未定义对并集查询和否定查询的扩展。

Gebhart 等人提出的一种更为新颖的方法 （2021年）基于层理论和代数拓扑（Hansen& Ghrist，2019年）。在该方法中，图被表示为胞腔层，合取查询则被建模为关系链（0-余链）。在查询图上诱导出一个层，相关答案应与所诱导的层和实体嵌入保持一致。优化问题是使用层拉普拉斯算子以及层拉普拉斯算子的舒尔补对 0-余链进行调和延拓。从概念上讲，该方法将投影和交集运算符的执行合并为拓扑结构上的函数。

对于 Transformer 编码器，BiQE（Kotnis 等人，2021年）、kgTransformer（Liu 等人，2022年）和 SQE（Bai 等人，2023年）将合取查询图线性化为由实体常量组成的关系路径序列 $\mathcal{C}$ 和关系标记。路径中的标记顺序以及路径之间的交集通过位置编码标记。目标节点（出现在许多路径中）由 [MASK] 标记标识（可选地，kgTransformer 也会用 [MASK]标注存在量化变量）。SQE 不会显式建模变量，而是依赖辅助的 括号 标记来分隔计算图的分支。将该序列输入 Transformer 编码器后，最终的查询嵌入就是目标节点的聚合表示。BiQE 仅支持合取查询，而 kgTransformer 会将包含并集的查询转换为析取范式（DNF），并对得分分布进行后处理（我们将在第6.1节进一步阐述查询重写和范式）。SQE 会将所有运算符标记显式纳入线性化序列，因此支持否定。

最后，MLPMix（Amayuelas 等人，2022年）按顺序执行查询中的各项操作，其中投影、交集和否定运算符分别建模为可学习的多层感知机。并集查询会被转换为析取范式，以便通过投影和交集运算符进行回答，最后再将分数作为并运算符进行后处理。类似地，Query2Particles（Bai 等人，2022a）将每个查询表示为



投影（盒体）　交集（盒体）　并集（圆锥体）　否定（圆锥体）

图15：几何处理器及其归纳偏置。

嵌入空间中的一组向量，并将投影、交集和否定运算符建模为对粒子集合进行注意力计算，再接一个多层感知机。并集则通过拼接查询粒子来表示。

表7：神经处理器。大多数方法仅实现关系投影和交集运算符。模型的上半部分按顺序执行查询，下半部分则编码完整查询图$\mathcal{G}_q$。

模型	嵌入空间	关系投影	交集	并集	否定	距离
GQE	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\mathbf{q}+\mathbf{r}$	$\texttt{DeepSet}(\{\mathbf{q_i}\})$	-	-	$\|\mathbf{q}-\mathbf{e}\|$
GQE+哈希	$\mathbf{q},\ \mathbf{e} \in \{\pm 1\}^d,\ \mathbf{r} \in \mathbb{R}^d$	$\texttt{sgn}(\mathbf{q}+\mathbf{r})$	$\texttt{sgn}(\texttt{DeepSet}(\{\mathbf{q_i}\}))$	-	-	$-cos(\mathbf{q},\mathbf{e})$
CGA	$\mathbf{q},\mathbf{e},\mathbf{r} \in \mathbb{R}^d$	$\mathbf{W}_r\mathbf{q}$	$\texttt{SelfAttn}(\{\mathbf{q_i}\})$	-	-	$-cos(\mathbf{q},\mathbf{e})$
RotatE-m	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{C}^d$	$\mathbf{q}\circ\mathbf{r}$	$\texttt{DeepSet}(\{\mathbf{q_i}\})$	-	-	$\|\mathbf{q}-\mathbf{r}\|$
DisMult-m	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\texttt{L2Norm}(\mathbf{q}\circ\mathbf{r})$	$\texttt{L2Norm}(\texttt{DeepSet}(\{\mathbf{q_i}\}))$	-	-	$-\langle\mathbf{q},\mathbf{e}\rangle$
ComplEx-m	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{C}^d$	$\texttt{L2Norm}(\mathbf{q}\circ\mathbf{r})$	$\texttt{L2Norm}(\texttt{DeepSet}(\{\mathbf{q_i}\}))$	-	-	$-\text{Re}(\langle\mathbf{q},\mathbf{e}\rangle)$
Query2Particles	$\mathbf{q},\ \mathbf{e} \in \mathbb{R}^{d\times K},\ \mathbf{r} \in \mathbb{R}^d$	$f(\mathbf{q},\ \mathbf{r})$ f 是神经门控	$\texttt{MLP}(\texttt{Attn}([\mathbf{q_1},\mathbf{q_2}]))$	$[\mathbf{q}^1,\ \ldots,\ \mathbf{q}^N]$	$\texttt{MLP}(\texttt{Attn}(\mathbf{q}))$	$-\text{最大值}_k\langle\mathbf{q}^k,\ \mathbf{e}\rangle$
SignalE	$\mathbf{q},\mathbf{e},\mathbf{r}=[\mathbf{z},\mathbf{v}]$ $\mathbf{z} \in \mathbb{C}^d,\ \mathbf{v}=\texttt{IDFT}(\mathbf{z}) \in \mathbb{R}^d$	$\mathbf{z}_q \circ \mathbf{z}_r$	$\texttt{SelfAttn}(\{\mathbf{z}^{q_i}\})$	DNF	$(1-\mathbf{z}_q^{\text{amp}},$ $\mathbf{z}_q^{\text{phase}})$	$\beta\|\mathbf{v}_q-\mathbf{v}_e\|_2+$ $k\mathbf{z}^q-\mathbf{z}_e k_2$
MLPMix	$\mathbf{q},\mathbf{e},\mathbf{r} \in \mathbb{R}^d$	$\texttt{MLP}_{\texttt{mix}}(\mathbf{q},\ \mathbf{r})$	$\texttt{MLP}_{\texttt{mix}}(\mathbf{q_1},\mathbf{q_2})$	DNF	$\texttt{MLP}(\mathbf{q})$	$\|\mathbf{q}-\mathbf{e}\|$
MPQE	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\mathbf{W}_r\mathbf{q}$	$\texttt{RGCN}(\mathbf{q},\ \mathcal{G}_q)$	-	-	$cos(\mathbf{q},\mathbf{e})$
StarQE	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$f(\mathbf{q}\ g(\mathbf{r},\mathbf{h}^{\text{quals}}))$	$\texttt{StarE}(\mathbf{q},\mathcal{G}_q)$	-	-	$\texttt{dot}(\text{查询},\ \text{边})$
GNNQ	$\mathbf{q},\ \mathbf{r} \in \mathbb{R}^d$	超图 $\texttt{RGCN}(\mathcal{G}')$		-	-	BCE
LMPNN	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{C}^d$	$\rho(\mathbf{q},\ \mathbf{r},\ \text{dir},\ \text{neg})$	$\texttt{GIN}(\mathbf{q},\mathcal{G}_q)$	DNF	$\rho(\mathbf{q},\ \mathbf{r},\ \text{dir},\ \text{neg})$	$\text{余弦}(\mathbf{q},\ \mathbf{e})$
BiQE	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\text{Transformer}(\mathbf{q},\ \text{线性化的}\ \mathcal{G}_q)$		-	-	CE
kgTransformer	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\text{Transformer}(\mathbf{q},\ \text{线性化的}\ \mathcal{G}_q)$		DNF	-	$\texttt{dot}(\mathbf{q},\ \mathbf{e})$
SQE	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} \in \mathbb{R}^d$	$\text{LSTM / Transformer}(\mathbf{q},\ \text{线性化}\ \mathcal{G}_q,\ \text{使用运算符标记})$				$\texttt{dot}(\text{查询},\ \text{实体})$
层	$\mathbf{q},\ \mathbf{e} \in \mathbb{R}^d,\ \mathcal{F}_r \in \mathbb{R}^{d\times d}$	$f(\mathbf{q},\ \mathcal{G}_q\ \text{作为余链})$		-	-	$k\mathcal{F}_r\mathbf{q}\ -\mathcal{F}_r\mathbf{e}k$

神经符号处理器。 不同于纯神经模型和符号模型，我们将神经符号处理器定义为满足以下条件的处理器：(1) 显式设计逻辑模块（或神经逻辑算子），以模拟真实的逻辑/集合操作，或依赖各种模糊逻辑公式，为查询执行过程提供概率视角；(2) 在潜在空间中执行关系遍历。神经符号处理器与前两类处理器的关键区别在于，神经符号处理器通过强归纳偏置显式建模逻辑操作，使处理/执行过程与符号操作更好地对齐（例如，通过对嵌入空间施加限制），并且更具可解释性。我们进一步将这些方法划分为以下类别。

几何处理器。 几何处理器设计具有不同几何直觉的实体/查询嵌入空间，并进一步定制神经符号算子，直接模拟具有相似属性的逻辑对应物（如图15所示）。我们根据几何模型的嵌入空间特征以及逻辑运算符的归纳偏置，在表8中汇总其特征。

Query2Box（Ren 等人，2020年）将查询 $\mathbf{q}$ 嵌入为欧几里得空间中的超矩形（高维盒）。为此，实体 $\mathbf{e}$ 和关系 $\mathbf{r}$ 均被嵌入为欧几里得空间中的点，其中每个

表8：几何神经符号处理器。 $\odot$ 表示逐元素乘法， $\oplus_c$ 表示莫比乌斯加法， $\odot_c$ 表示莫比乌斯标量积。 DS 表示 DeepSets 神经网络。NewLook 仅将否定部分实现为差分算子。

模型	嵌入空间	关系投影	交集	并集	否定	距离
Query2Box Query2Onto	$\mathbf{q},\ \mathbf{r} \in \mathbb{R}^{2d},\ \ \mathbf{e} \in \mathbb{R}^{d}$	$\mathbf{q}+\mathbf{r}$	$\mathbf{q}_c = \texttt{Attn}(\{\mathbf{q}_c^i\})$ $\mathbf{q}_o = \min(\{\mathbf{q}_o^i\}) \odot \sigma(\texttt{DS}(\{\mathbf{q}_o^i\}))$	DNF	-	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$
Query2Geom	$\mathbf{q},\ \mathbf{r} \in \mathbb{R}^{2d},\ \ \mathbf{e} \in \mathbb{R}^{d}$	$\mathbf{q}+\mathbf{r}$	$\mathbf{q}_c = \frac{1}{2}((\mathbf{q}_c^i + \mathbf{q}_o^i) + (\mathbf{q}_c^i - \mathbf{q}_o^i))$ $\mathbf{q}_o = \mathbf{q}_c - (\mathbf{q}_c^i - \mathbf{q}_o^i)$	DNF	-	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$
RotatE-Box	$\mathbf{q},\mathbf{r} \in \mathbb{C}^{2d},\mathbf{e} \in \mathbb{C}^{d}$	$(\mathbf{q}_c \circ \mathbf{r}_c, \mathbf{q}_o + \mathbf{r}_o)$	-	析取范式 / $\texttt{DS}(\{\mathbf{q}^i\})$	-	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$
NewLook	$\mathbf{q},\mathbf{r} \in \mathbb{R}^{2d},\mathbf{e} \in \mathbb{R}^{d}$ $\mathbf{x} \in \mathcal{T}^{\lvert\mathcal{R}\rvert\times\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	$\texttt{MLP}[\texttt{MLP}(\mathbf{q}_c + \mathbf{r}_c) \Vert$ $\texttt{MLP}(\mathbf{r}_o) \Vert \mathbf{x}_t]$	$\mathbf{q}_c = \texttt{Attn}\ (\{\mathbf{q}_c^i,\ \mathbf{x}^i\})$ $\mathbf{q}_o = \min(\{\mathbf{q}_o^i\}) \odot \sigma(\texttt{DS}(\{\mathbf{q}_o^i\}))$	DNF	$\texttt{Attn}(\{\mathbf{q}_c^i\})$ $\texttt{Attn}(\{\mathbf{q}_o^i, \mathbf{x}^i\})$	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$
HypE	$\mathbf{q},\ \mathbf{r} \in \mathbb{R}^{2d},\ \ \mathbf{e} \in \mathbb{R}^{d}$	$\mathbf{q} \oplus_c \mathbf{r}$	$\mathbf{q}_c = \texttt{Attn}(\{\mathbf{q}_c^i\})$ $\mathbf{q}_o = \min(\{\mathbf{q}_o^i\}) \odot_c \sigma(\texttt{DS}(\{\mathbf{q}_o^i\}))$	DNF	-	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$
ConE	$\mathbf{q},\ \mathbf{r} = (\boldsymbol{\theta}_{\text{ax}},\ \boldsymbol{\theta}_{\text{ap}})$ $\mathbf{e} = (\boldsymbol{\theta}_{\text{ax}}, \mathbf{0})$ $\boldsymbol{\theta}_{\text{ax}} \in [-\pi, \pi)^d$ $\boldsymbol{\theta}_{\text{ap}} \in [0, 2\pi]^d$	$g(\texttt{MLP}(\mathbf{q}+\mathbf{r}))$ g 门控 $\boldsymbol{\theta}_{\text{ax}}$ 和 $\boldsymbol{\theta}_{\text{a p}}$	$\boldsymbol{\theta}_{\text{ax}} = \texttt{SemanticAvg}(\mathbf{q}_1, \ldots, \mathbf{q}_n)$ $\boldsymbol{\theta}_{\text{ap}} = \texttt{CardMin}(\mathbf{q}_1, \ldots, \mathbf{q}_n)$	DNF/DM	$\boldsymbol{\theta}_{\text{ax}} = \boldsymbol{\theta}_{\text{ax}} \pm \pi$ $\boldsymbol{\theta}_{\text{ap}} = 2\pi - \boldsymbol{\theta}_{\text{ap}}$	$\mathrm{d}_{\text{out}} + \alpha \mathrm{d}_{\text{in}}$

关系都有一个额外的可学习偏移向量 $\mathbf{r}_o$ （实体的偏移量为零）。投影算子被建模为对查询和关系的中心与偏移量进行逐元素求和 $\mathbf{q}+\mathbf{r}$，即初始盒体是通过将原始锚定节点嵌入 $\mathbf{e}$（偏移量为零）与关系嵌入 $\mathbf{r}$ 及关系偏移量 $\mathbf{r}_o$进行投影而得到的。相应地，研究者设计了一种基于注意力的神经交集算子，用于模拟欧几里得空间中查询盒体的集合交集。该算子具有封闭性和置换不变性，并且符合交集集合的规模小于所有输入集合规模这一直觉。并运算符通过析取范式实现，即将操作数盒体的结果拼接作为最后一步。多项工作对 Query2Box 进行了扩展：Query2Onto（Andresel 等，2021）尝试通过实例化蕴含三元组，并利用盒嵌入的包含关系强制建立层级关系，来建模复杂的本体公理；RotatE-Box（Adlakha 等，2021）设计了一个额外的基于旋转的克莱尼正闭包（+）算子，用于表示关系路径（我们将在第6.1节详细介绍克莱尼正闭包算子）；NewLook（Liu 等，2021）在算子中加入了从邻接张量[6] 进行符号查找的功能，使用多层感知机修改投影，并将差分算子建模为对中心与偏移量的注意力（请注意，差分算子是 2in 否定查询的一种特殊情况，我们将在第6.1节详细介绍）；Query2Geom（Sardina 等，2023年）则以简单的非参数闭式几何交集取代了基于注意力的交集算子。

HypE（Choudhary 等人，2021b）扩展了 Query2Box 的思想，将查询嵌入为庞加莱超球中的双曲面（两对平行的弧对齐 horocycle），以更好地捕获层级信息。对于双曲面嵌入，他们设计了类似的基于注意力的神经交集算子，旨在缩小与 DeepSets 之间的差距。

ConE（Zhang 等人，2021b）则将查询嵌入到一组单位圆的表面上。每个查询都表示为一个圆锥截面，其优势在于，在大多数情况下，圆锥体的交集仍然是圆锥体；此外，由于角空间受 2π 限制，圆锥体的否定/补集也仍然是圆锥体。基于这一见解，他们设计了几何神经交集和否定算子。

总而言之，基于几何的处理器通常采用强几何先验进行设计，以便更好地模拟或满足逻辑/集合操作的性质。

概率处理器。 概率处理器并非采用几何嵌入空间，而是旨在概率空间中对查询以及逻辑/集合运算进行建模。这类方法与基于几何的方法相似，因为它们同样需要一个概率空间，并可在其上设计与真实算子高度一致的神经符号逻辑/集合算子。 图 16展示了在概率空间中实现逻辑运算符的一些示例。聚合后的特征见于表 9。

[6]实现错误导致了严重的测试集泄漏以及不正确的结果报告。

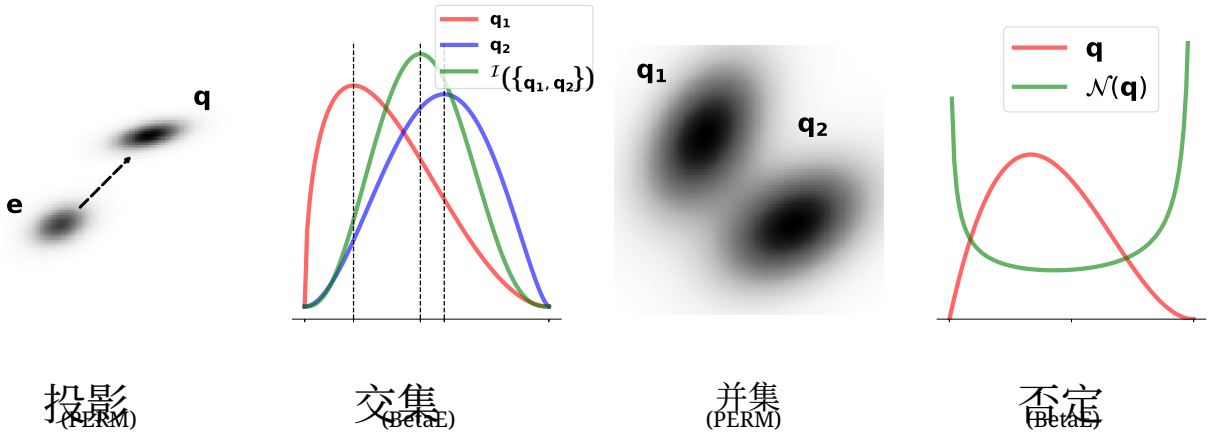


图 16：概率处理器及其归纳偏置。

BetaE（Ren 与 Leskovec，2020年）基于Beta分布，将实体/查询嵌入表示为具有可学习参数的高维Beta分布。其优点在于，可以针对两个Beta嵌入设计参数化的神经交集算子，其输出仍是Beta嵌入，但具有更集中的概率密度函数。通过简单地对参数取倒数以翻转密度，即可设计神经否定算子。PERM（Choudhary 等人，2021a）考察高斯分布空间，并将查询嵌入表示为多元高斯分布。由于高斯概率密度函数（PDF）的乘积仍是高斯概率密度函数，神经交集算子相应地计算交集集合的高斯嵌入参数。NMP-QEM（Long 等，2022年）进一步发展了这一思想，将查询表示为高斯混合，其中逻辑运算符通过对分布参数使用多层感知机或注意力进行建模。LinE（Huang 等，2022b）将Beta分布转换为离散序列的值。它采用与BetaE类似的神经否定算子，即取倒数，同时通过逐元素取最小值/最大值来设计新的神经交集/并运算符。GammaE（Yang 等，2022a）以伽马分布替代Beta分布，作为实体和查询嵌入空间。通过利用伽马分布混合上的操作对逻辑运算符进行参数化，并集和否定变为封闭运算，无需析取范式或德摩根变换，并且可以沿查询计算图顺序执行。总体而言，概率处理器与几何处理器相似，因为它们都受到用于嵌入的概率和几何的某些性质启发，并对神经逻辑算子进行了定制。

表 9：概率神经符号处理器。

模型	嵌入空间	关系投影	交集	并集	否定	距离
BetaE	$\mathbf{q}.\mathbf{e} \in \mathbb{R}^{2d}$	$\texttt{MLP}_r(\mathbf{q})$	$\mathbf{q} = [(\sum w_i\alpha_i, \sum w_i\beta_i)]$	DNF/DM	$\frac{1}{\mathbf{q}}$	$\texttt{KL}(\mathbf{e};\mathbf{q})$
PERM	$\mathbf{q},\ \mathbf{e},\ \mathbf{r} = N(\boldsymbol{\mu}, \Sigma)$	$N(\boldsymbol{\mu}_q + \boldsymbol{\mu}_r, (\Sigma_q^{-1} + \Sigma_r^{-1})^{-1})$	$\Sigma_q^{-1} = \Sigma_{q_1}^{-1} + \Sigma_{q_2}^{-1}$ $\boldsymbol{\mu}_q = \Sigma_q(\Sigma_{q_2}^{-1}\boldsymbol{\mu}_1 + \Sigma_{q_1}^{-1}\boldsymbol{\mu}_2)$	$\texttt{Attn}(\{\mathbf{q}_i\})$	-	$(\boldsymbol{\mu}_q - \boldsymbol{\mu}_e)^T\Sigma_q^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_e)$
LinE	$\mathbf{q},\ \mathbf{e} \in \mathbb{R}^{k\times h}$	$\texttt{MLP}_r(\mathbf{q})$	最小值$\{q_1, q_2\}$	$\max\{q_1, q_2\}$	$[\frac{1}{p_1}\cdots, \frac{1}{p_k^q}]^h$	$\|\mathbf{q} - \mathbf{e}\|_2^2$
GammaE	$\mathbf{q},\ \mathbf{e} \in \mathbb{R}^{2d}$	$\texttt{MLP}_r(\mathbf{q})$	$\mathbf{q} = [(\sum w_i\alpha_i, \sum w_i\beta_i)]$	$\texttt{Attn}(\{\mathbf{q}_i\})$	$[\frac{1}{\alpha}, \beta]$	$\texttt{KL}(\mathbf{e};\mathbf{q})$
NMP-QEM	$\mathbf{e} \in \mathbb{R}^d$ $\mathbf{q} = \sum_{i=1}^K \omega_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$	$\texttt{MLP}_r(\mathbf{q})$	$\texttt{Attn}(\{\mathbf{q}_i\})$	DNF	$\texttt{MLP}(\mathbf{q})$	$k\mathbf{e} - \sum_{i=1}^K \omega_i^q \boldsymbol{\mu}_i^q k$

模糊逻辑处理器。 与上述方法不同，模糊逻辑处理器直接利用现有的模糊逻辑理论对所有逻辑操作进行建模（Klement 等人，2013年；van Krieken 等人，2022年），其中交集可以通过t-范数表示，并集则通过相应的t-余范数（第2.6节）表示。这样一来，模糊逻辑处理器无需像前两种处理器那样手动设计或学习神经逻辑算子，而是直接使用已建立的模糊算子，这些算子通常表示为对向量执行的可微、逐元素代数算子（图17）。虽然交集、并集和否定属于非参数化操作，但投影算子仍可以通过神经网络进行参数化。模糊逻辑处理器的聚合特性见表10。总体而言，模糊处理器旨在结合执行

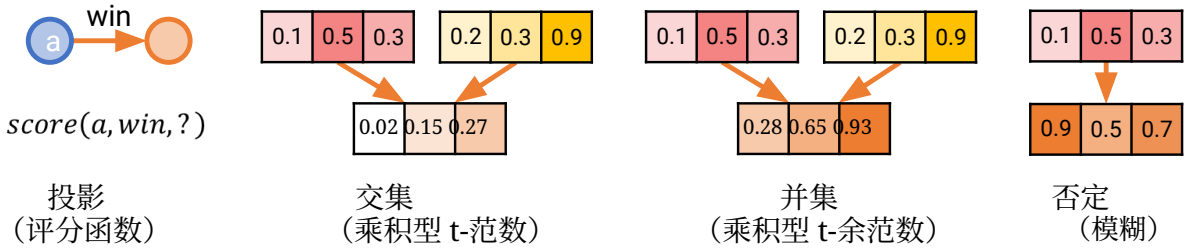


图 17：模糊逻辑处理器与模糊逻辑算子。

在嵌入空间（向量）中执行的操作与实体空间（符号）。所述方法在设计这种组合方式上各不相同。

早期的模糊处理器之一是 EmQL（Sun 等，2020），它将计数最小草图（Cormode 和 Muthukrishnan，2005）融入实体和关系嵌入中。在该方法中，投影、交集和并集同时在嵌入空间和符号草图空间中执行，例如，交集被建模为逐元素乘法，而并集则是两个草图的逐元素求和。CQD（Arakelyan 等，2021）使用预训练神经链接预测器为查询中的每个原子公式评分，并利用 t-范数和 t-余范数直接在嵌入空间中计算查询的最终分数。CQD 不会训练任何神经逻辑算子，只需使用单跳链接预训练实体和关系嵌入，使投影算子等价于所选评分函数（例如 ComplEx（Lacroix 等人，2018年））的 top-k 结果。此后，这一思路沿着多个方向得到扩展：查询树优化（QTO）（Bai 等，2022b）在关系投影步骤中加入了对所有可能三元组分数的物化张量 $\mathbf{M} \in [0,1]^{|\mathcal{R}|\times|\mathcal{E}|\times|\mathcal{E}|}$ 的查找；CQD$^{\mathcal{A}}$（Arakelyan 等，2023年）和 Var2Vec（Wang 等，2023a）则学习实体-关系拼接 W[q, r]的额外线性变换。

LogicE（Luus 等人，2021年）为每个实体设计逻辑嵌入，其中包含一系列下界 – 上界对，值域为 $[0,1]$, ，可将其解释为下界与上界之间的均匀分布。LogicE 使用连续 t-范数对上下界执行否定和合取操作。FuzzQE（Chen 等人，2022b）和 TAR（Tang 等人，2022年将查询嵌入到高维模糊空间中 $[0,1]^d$，并同样使用 Gödel t- 范数和 Łukasiewicz t- 范数来建模析取、合取和否定。FuzzQE 将关系投影建模为关系特定 MLP，而 TAR 使用几何平移（逐元素求和）。FLEX（Lin 等人，2022b）和 TFLEX（Lin 等人，2022a）将查询嵌入为特征与逻辑嵌入的混合表示。对于逻辑部分，两种方法都使用向量逻辑中的真实逻辑操作（Mizraji，2008年）。TFLEX 添加了一个时间模块，根据时间嵌入对逻辑运算符进行条件化。

GNN-QE（Zhu 等人，2022年）使用图神经网络 NBFNet（Zhu 等人，2021年）对每个关系投影步骤中所有实体的似然进行建模。随后，它采用乘积逻辑，直接对关系投影后获得的模糊集上的集合运算（交集、并集和否定）进行建模。GNN-QE 采用节点标记技术，将起始节点初始化为关系向量（其他节点则初始化为零）。这使 GNN-QE 具备归纳式能力，并且无需依赖可训练实体嵌入。另一方面，ENeSy（Xu 等，2022年）同时为查询、实体和关系保留向量表示与符号表示（其中，符号关系被编码为$\mathbf{M}_r$ 稀疏邻接矩阵）。逻辑运算符首先在神经空间中执行，例如，关系投影采用 RotatE 组合函数（Sun 等人，2018年），然后与符号表示交织在一起。符号空间中的逻辑运算符采用乘积逻辑的广义版本及相应的 t-（余）范数。

总而言之，模糊逻辑处理器直接依赖既有的模糊逻辑形式体系来执行查询中的所有逻辑操作，从而避免在（可能）无界的嵌入空间中手动设计和训练神经算子。模糊逻辑空间是连续的，但被限制在 $[0,1]$ 之内——这既是此类处理器的优势，也是其弱点。对于封闭逻辑算子而言，有界空间很有益，因为它们的输出值仍属于同一有界空间。另一方面，大多数已知的 t-范数（及相应的 t-余范数）仍会导致梯度消失，只有乘积逻辑范数是稳定的（van Krieken 等人，2022年；Badreddine 等人，2022年）。另一个需要注意的问题是，如何设计一种有效且

表10：模糊神经符号处理器。

模型	嵌入空间	关系投影	交集	并集	否定	距离
EmQL	$\mathbf{e},\ \mathbf{r}\in\mathbb{R}^d,\ \mathbf{q}\in\mathbb{R}^{3d}$ $\mathbf{b}$ 计数最小草图	$\texttt{MIPS}(\mathbf{q},[\mathbf{r},\ \mathbf{e}_h,\ \mathbf{e}_t])$	$(\mathbf{q_1}+\mathbf{q_2})/2$ $\mathbf{b}_1\odot\mathbf{b}_2$	$(\mathbf{q_1}+\mathbf{q_2})/2$ $\mathbf{b}_1+\mathbf{b}_2$	-	$\texttt{dot}(\mathbf{q},\ \mathbf{e})$
CQD	$\mathbf{q},\mathbf{e},\mathbf{r}\in\mathbb{C}^d$	最小值$_{q_2}\ d(\mathbf{q}_1,\mathbf{r},\ q_2)$ 或 $\text{top}_k(d(\mathbf{q}_1,\mathbf{r},\mathbf{e_k}))$	乘积：$\mathbf{q_1}\cdot\mathbf{q_2}$ Gödel：最小值$(\mathbf{q}_1,\mathbf{q_2})$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$ 最大值$(\mathbf{q}_1,\mathbf{q_2})$	-	$\texttt{dist}$(查询，边)
CQD$^{\mathcal{A}}$	$\mathbf{q},\ \mathbf{e},\ \mathbf{r}\in\mathbb{C}^d$ $\mathbf{W}\in\mathbb{R}^{2\times 2d}$	$\theta=\mathbf{W}[\mathbf{q}_1,\mathbf{r}]$ 顶部$_k[\rho_\theta(d(\mathbf{q}_1,\mathbf{r},\mathbf{e_k}))]$	乘积：$\mathbf{q_1}\cdot\mathbf{q_2}$ Gödel：最小值$(\mathbf{q}_1,\mathbf{q_2})$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$ 最大值$(\mathbf{q}_1,\mathbf{q_2})$	$1-\mathbf{q}$ $(1+\cos(\pi\mathbf{q}))/2$	$\texttt{dist}$(查询，实体)
Var2Vec	$\mathbf{q},\mathbf{e},\mathbf{r}\in\mathbb{C}^d$ $\mathbf{W}\in\mathbb{R}^{d\times 2d}$	$\mathbf{q}_2=\mathbf{W}[\mathbf{q}_1,\mathbf{r}]$ $d(\mathbf{q}_1,\mathbf{r},\ \mathbf{q}_2)$	乘积：$\mathbf{q_1}\cdot\mathbf{q_2}$ Gödel：最小值$(\mathbf{q}_1,\mathbf{q_2})$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$ 最大值$(\mathbf{q}_1,\mathbf{q_2})$	$1-\mathbf{q}$	$\texttt{dist}$(查询，实体)
QTO	$\mathbf{q},\ \mathbf{e},\ \mathbf{r}\in\mathbb{C}^d$ $\mathbf{M}\in[0,1]^{\lvert\mathcal{R}\rvert\times\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	$d(\mathbf{q}_1,\mathbf{r},\mathbf{q}_2)$ $\text{row}_{q1}(\mathbf{M}_r)$	$\mathbf{q_1}\cdot\mathbf{q_2}$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$	$d(\mathbf{q}_1,\mathbf{r},\mathbf{q}_2)$ 行$_{q1}(\mathbf{1}-\mathbf{M}_r)$	$\texttt{dist}(\mathbf{q},\mathbf{e})$
LogicE	$\mathbf{q},\ \mathbf{e}=([l_i,u_i],$ $l_i,u_i\in[0,1])_{i=1}^d$ $\mathbf{r}\in\mathbb{R}^d$	σ(最大值(0，最大值(0， $[\mathbf{r},\mathbf{q}]\mathbf{F}_1)\mathbf{F}_2)\mathbf{F}_3)$	$([\top(l_i^{(1)},\ldots,l_i^{(n)}),$ $\top(u_i^{(1)},\ldots,u_i^{(n)}))$ $i=1\ldots d$	DM	$([1-l_i,$ $1-u_i])$ $i=1\ldots d$	$\lVert\mathbf{q}-\mathbf{e}\rVert_1$
FuzzQE	$\mathbf{q},\ \mathbf{e}\in[0,1]^d$	$\sigma(\texttt{MLP}_r(\mathbf{q}))$	乘积：$\mathbf{q_1}\cdot\mathbf{q_2}$ Gödel：最小值$(\mathbf{q}_1,\mathbf{q_2})$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$ 最大值$(\mathbf{q}_1,\mathbf{q_2})$	$1-\mathbf{q}$ -	$\texttt{dot}$(查询，实体)
TAR	$\mathbf{q},\mathbf{e},\mathbf{r}\in\mathbb{R}^d$	$\mathbf{q}+\mathbf{r}$	$\texttt{Attn}(\mathbf{q}_1,\mathbf{q}_2)$	最大值$(\mathbf{q}_1,\mathbf{q}_2)$	$1-\mathbf{q}$	$\lVert\mathbf{q}-\mathbf{e}\rVert$
GNN-QE	$\mathbf{q},\ \mathbf{r}\in\mathbb{R}^{\lvert\mathcal{E}\rvert}$	$\sigma(\texttt{GNN}(\mathbf{q},\mathcal{G}))$	$\mathbf{q_1}\cdot\mathbf{q_2}$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$	$1-\mathbf{q}$	$\texttt{BCE}(\mathbf{q})$
FLEX	$\mathbf{q}=(\theta_f,\theta_l)$ $\theta_f\in[-L,L]^d$ $\mathbf{e}=(\theta_f,\mathbf{0})$ $\theta_l\in[0,1]^d$	$g(\texttt{MLP}([\theta_f+\theta_{f,r};$ $\theta_l+\theta_{l,\ r}]))$ g 门控 θ_f，$\theta_{e\circ}$	$\theta_f=\sum_i\theta_{q_i,\ f}$ $\theta_l=\prod_i\theta_{q_i,l}$ $\sum_i\alpha\mathbf{q}^{e}_{i,f}\bigotimes_i(\{\mathbf{q}^{e}_{i,l}\})$, $\sum_i\beta\mathbf{q}^{t_{i,f}}\bigotimes_i(\{\mathbf{q}^{t}_{i,l}\})$	$\theta_f=\sum_i\mathbf{a}_i\theta_{q_i,f}$ $\theta_l=\sum_i\theta_{q_il}-\sum_{1\le i<j\le n}\theta_{q_il}\theta_{q_jl}+\cdots+(-1)^{n-1}\prod_i\theta_{q_i,\ l}$ $\sum_i\alpha\mathbf{q}^{e}_{i,f}\bigoplus_i(\{\mathbf{q}^{e}_{i,l}\})$, $\sum_i\beta\mathbf{q}^{t_{i,f}}\bigotimes_i(\{\mathbf{q}^{t}_{i,l}\})$	$\theta_f=L\cdot\tanh($ $\texttt{MLP}([\theta_f;\ \theta_l]))$ $\theta_l=1-\theta_l$	$k\theta^e_f-\theta^q_f\ k_1+$ θ^q_l
TFLEX	$\mathbf{q},\mathbf{r}=(\mathbf{q}^e_f,\mathbf{q}^e_l,\mathbf{q}^t_f,\mathbf{q}^t_l)$ $\mathbf{q}^e_f,\mathbf{q}^t_f,\mathbf{e}^e_f\in\mathbb{R}^d$ $\mathbf{e}=(\mathbf{e}^e_f,\mathbf{0},\mathbf{0},\mathbf{0})$ $\mathbf{q}^e_l,\ \mathbf{q}^t_l\in[0,1]^d$	实体：$g(\texttt{MLP}($ $\mathbf{q}+\mathbf{r}+\mathbf{t}))$ 时间：$g(\texttt{MLP}($ $\mathbf{q_1}+\mathbf{r}+\mathbf{q_2}))$	$\sum_i\alpha\mathbf{q}^{e}_{i,f}\bigotimes_i(\{\mathbf{q}^{e}_{i,l}\})$, $\sum_i\beta\mathbf{q}^{t_{i,f}}\bigotimes_i(\{\mathbf{q}^{t}_{i,l}\})$ $\sum_i\alpha\mathbf{q}^{e}_{i,f}\bigoplus_i(\{\mathbf{q}^{e}_{i,l}\})$, $\sum_i\beta\mathbf{q}^{t_{i,f}}\bigotimes_i(\{\mathbf{q}^{t}_{i,l}\})$	$f^e_{\text{not}}(\mathbf{q}^e_f)$, $\ominus(\mathbf{q}^e_l),\ \mathbf{q}^t_f,\ \mathbf{q}^t_l$ $\mathbf{q}^e_f,\ \mathbf{q}^e_l,$ $f^t_{\text{not}}(\mathbf{q}^t_f)\ \ominus(\mathbf{q}^t_l)$	$k\mathbf{e}^e_f-\mathbf{q}^e_f k_1+$ $\mathbf{q}^e_f\rVert_1+\mathbf{q}^e_l$ $k\mathbf{t}^t_f-\mathbf{q}^t_f k_1+$ $\mathbf{q}^t_l$
ENeSy	$\mathbf{q},\ \mathbf{e},\ \mathbf{r}\in\mathbb{C}^d$ $\mathbf{p}_a\mathbf{p}_e\in\{0,1\}^{\lvert\mathcal{E}\rvert}$ $\mathbf{M}_r\in\{0,1\}^{\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	神经：$\mathbf{q}\circ\mathbf{r}$ 符号：$g(\mathbf{p}_q\mathbf{M}_r)^\top$ $g=\mathbf{x}/\text{SUM}(\mathbf{x})$	$g(\mathbf{p_1}\cdot\mathbf{p_2})$	$g(\mathbf{p_1}+\mathbf{p_2}-\mathbf{p_1}\cdot\mathbf{p_2})$	$g(\overset{\alpha}{_\mathcal{E}}-\mathbf{p})$	$\lVert\mathbf{q}-\mathbf{e}\rVert$
NQE	$\mathbf{q},\mathbf{e},\mathbf{r}\in[0,1]^d$	$\sigma(\texttt{Trf}(\mathbf{q},\ \mathcal{G}_q))$	$\mathbf{q_1}\cdot\mathbf{q_2}$	$\mathbf{q_1}+\mathbf{q_2}-\mathbf{q_1}\cdot\mathbf{q_2}$	$1-\mathbf{q}$	$\texttt{dot}(\mathbf{q},\ \mathbf{e})$

可微的交互机制，以连接模糊空间 $[0,1]^d$ 与无界嵌入空间 $\mathbb{R}^d$ （或$\mathbb{C}^d$），关系表示通常由此初始化。也就是说，在处理计算图时，对向量值进行重新缩放和压缩可能会导致梯度含噪且训练不稳定；例如，GNN-QE 就观察到了这一问题，因此不得不关闭除最后一个投影步骤之外所有步骤的梯度。

5.3 解码器

解码的目标是获得最终的答案集合，或对所有实体进行排序。这是处理之后查询回答任务的最后一步。我们将这些方法分为两类：非参数化方法和参数化方法。参数化方法需要通过参数化方法为实体评分（或从处理后的潜变量中预测回归目标），而非参数化方法则可以直接衡量图上查询与实体这一对对象之间的相似度（或距离）。如处理器表中的距离列所示，大多数方法属于非参数化类别，见表 7、表 8、表 9 和表10。例如，几何模型（Ren 与 Leskovec，2020；Andresel 等，2021；Adlakha 等，2021；Choudhary 等，2021b；Zhang 等，2021b）预先定义了查询表示与实体表示之间的距离函数。常用的距离函数包括 L1（Hamilton 等，2018；Ren 等，2022；Gebhart 等，2021；Amayuelas 等，2022；Tang等，2022；Xu 等，2022；Long 等，2022）、L2（Huang 等，2022b；Chen 等，2022b）或其变体（Luus 等，2021；Lin 等，2022b;a）、余弦相似度（Wang 等，2019；Mai 等，2019；Daza 与 Cochez，2020；Wang 等，2023c）、点积（Sun 等，2020；Ren 等，2022；Alivanistos 等，2022；Liu 等，2022；Bai 等，2022a；Luo 等，2023），或者直接对所有实体的似然进行建模，而无需距离函数（Arakelyan 等，2021；Kotnis 等，2021；Zhu 等，2022；Pflueger 等，2022；Bai 等，2022b；Arakelyan 等，2023年；Wang 等，2023a）。概率模型通常采用 KL 散度（Ren 与 Leskovec，2020；Yang 等，2022a）或马氏距离（Choudhary 等，2021a）。

当前方法大多忽视了一个重要方向（与距离函数正交）：如何在拥有数十亿实体的超大型图上高效执行答案实体检索。一种可扩展的

近似最近邻（ANN）搜索算法是必需的。包括 FAISS（Johnson 等人，2019年）或 ScaNN（Guo 等人，2020年）在内的现有框架提供了 ANN 的可扩展实现。然而，ANN 仅限于 L1、L2 和余弦距离，并且主要针对 CPU 进行了优化。如何针对 KL 散度等更复杂的距离函数设计高效、可扩展的 ANN 搜索算法，使我们能够针对采用不同距离函数的不同 CLQA 方法，以更高的效率进行检索（最好使用 GPU），仍然是一个开放的研究问题。

我们推测，在查询求答中，参数化解码器除了用于普通的实体检索外，还将在数值任务中得到更广泛的应用。这类任务可能涉及节点、边和图层级上的数值与类别特征，例如，训练回归器来预测节点属性的数值，比如年龄、长度等。此外，参数化解码器也为归纳式设置下的泛化带来了新的机会，因为评估时可能会遇到未见实体。SE-KGE（Mai 等人，2020年）通过预测查询目标的地理空间坐标，在这一方向上迈出了一步。

5.4 计算复杂度

这里我们分析不同查询推理模型的时间复杂度，并根据不同操作对其进行分类，包括关系投影、交集、并集、否定，以及获得查询表示后的答案检索。我们在表11中列出了对查询图进行逐步编码的方法的渐近复杂度；在表12中列出了同时编码完整查询图（包括投影及其他逻辑操作）的方法（主要基于图神经网络）的渐近复杂度。

表 11：在 $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$ 上各项操作的时间复杂度。对于所有方法，我们将多层感知机的嵌入维度和隐藏维度记为 d，将多层感知机/图神经网络的层数记为 l，将交集操作中的分支数记为 i，将并集操作中的分支数记为 u。

模型	投影	交集	否定	并集	答案检索	定义
GQE GQE+哈希 RotatE-m Distmult-m ComplEx-m	$\mathcal{O}(d)$	$\mathcal{O}(ild^2)$	-	-	$O(\lvert E\rvert d)$	-
CGA	$\mathcal{O}(d^2)$	$\mathcal{O}(id+d^2)$	-	-	$O(\lvert E\rvert d)$	-
Query2Particles	$\mathcal{O}(Kd^2)$	$\mathcal{O}(iKd^2)$	$\mathcal{O}(Kd^2)$	DNF	$O(\lvert E\rvert dK)$	K：#粒子。
SignalE	$\mathcal{O}(d)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(d)$	DNF	$O(\lvert E\rvert d)$	-
MLPMix	$\mathcal{O}(ld^2)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(ld^2)$	DNF	$O(\lvert E\rvert d)$	-
Query2Box Query2Onto	$\mathcal{O}(d)$	$\mathcal{O}(ild^2)$	-	DNF	$O(\lvert E\rvert d)$	-
Query2Geom	$\mathcal{O}(d)$	$\mathcal{O}(d)$	-	DNF	$O(\lvert E\rvert d)$	-
RotatE-Box	$\mathcal{O}(d)$	-	-	析取范式 /$\mathcal{O}(uld^2)$	$O(\lvert E\rvert d)$	-
NewLook	$O(\lvert E\rvert d+ld^2)$	$\mathcal{O}(i(\lvert\mathcal{E}\rvert+ld^2))$	$\mathcal{O}(ld^2)$	DNF	$O(\lvert E\rvert d)$	-
HypE	$\mathcal{O}(d)$	$\mathcal{O}(ild^2)$	-	DNF	$O(\lvert E\rvert d)$	-
ConE/BetaE	$\mathcal{O}(ld^2)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(d)$	DNF	$O(\lvert E\rvert d)$	-
PERM	$\mathcal{O}(d^2)$	$\mathcal{O}(id^3)$	-	DNF	$O(\lvert E\rvert d^2)$	-
LinE	$\mathcal{O}(ld^2)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$O(\lvert E\rvert d)$	-
GammaE	$\mathcal{O}(ld^2)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(d)$	$\mathcal{O}(uld^2)$	$O(\lvert E\rvert d)$	-
NMP-QEM	$\mathcal{O}(Kld^2)$	$\mathcal{O}(iK^2d)$	$\mathcal{O}(Kld^2)$	DNF	$O(\lvert E\rvert dK)$	K：#中心
EmQL	$O(\lvert E\rvert d)$	$\mathcal{O}(d)$	-	$\mathcal{O}(d)$	$O(\lvert E\rvert d)$	-
CQD-CO	$O(\lvert E\rvert d)$	Opt	-	Opt	$O(\lvert E\rvert d)$	-
CQD-Beam	$O(\lvert E\rvert d)$	$O(\lvert E\rvert kd)$	-	$O(\lvert E\rvert kd)$	$O(\lvert E\rvert d)$	-
QTO	$\mathcal{O}$(最大值$_k$ $\lvert T^*(v_k)$ $> 0\rvert\lvert E\rvert$)	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	$T^*(v)$: 最大真值 对于子查询 以节点 v为根节点。
LogicE	$\mathcal{O}(ld^2)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$O(\lvert E\rvert d)$	-
FuzzQE	$\mathcal{O}(d^2)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$O(\lvert E\rvert d)$	-
TAR	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$O(\lvert E\rvert d)$	-
GNN-QE	$O(\lvert E\rvert d^2+\lvert S\rvert d)$	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	$O(\lvert E\rvert)$	-
FLEX	$\mathcal{O}(ld^2)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(ld^2)$	$\mathcal{O}(uld^2+2^ud)$	$O(\lvert E\rvert d)$	-
TFLEX	$\mathcal{O}(ld^2)$	$\mathcal{O}(ild^2)$	$\mathcal{O}(ld^2)$	$\mathcal{O}(uld^2)$	$O(\lvert E\rvert d)$	-

表12：直接编码查询图的方法的查询求答时间复杂度。除表 11中定义的算子外，我们将 n_q 表示查询图 $\mathcal{G}_q = (\mathcal{E}_q, \mathcal{R}_q, \mathcal{S}_q)$的平均度。

模型	投影	交集	否定	并集	答案检索
MPQE / GNNQ	$O(d^2 n_q l\|E_q\|)$		-	-	$O(\|E\|d)$
StarQE	$O(d^2 n_q l\|E_q\| + \|S_q\|\|qp\|d)$		-	-	$O(\|E\|d)$
LMPNN	$O(d^2 n_q l\|E_q\|)$			DNF	$O(\|E\|d)$
BiQE	$\mathcal{O}((\|\mathcal{E}_q\|d^2 + \|\mathcal{E}_q\|^2 d)l)$		-	-	$O(\|E\|d)$
kgTransformer	或$((d^2 n_q \|存在_q\| + \|存在_q\| n_q^2 d)l)$			-	$O(\|E\|d)$
SQE	$\mathcal{O}((\|\mathcal{E}_q\|d^2 + \|\mathcal{E}_q\|^2 d)l)$				$O(\|E\|d)$

6 个查询

从查询的角度对这些方法进行划分，是第三种划分方式。在查询类别下，我们有三个子类别：查询算子、查询模式和投影变量。对于查询算子，不同方法具有不同的算子表达能力，即模型能够处理的查询算子集合，包括存在量化（∃）、合取（∧）、析取（∨）、否定（¬）、克莱尼正闭包（+）、过滤器以及各种聚合运算符。对于查询模式，我们指（优化后）查询计划的结构/模式，范围从路径和树到任意有向无环图（DAG）及循环模式。至于投影变量（投影变量 指必须绑定到特定图元素（如实体或关系）的目标变量），查询中的目标变量数量可能不同（零个或多个）。我们关注此类投影的复杂度，因为绑定两个或更多变量涉及关系代数（Codd，1970年），并可能导致所有检索到的答案的笛卡尔积。

6.1 查询算子

不同的查询处理器在方法能够处理的算子方面具有不同的表达能力。在所有相关工作中，我们汇总了 表13，根据所支持的算子对所有方法进行了分类。

表13：查询回答处理器及其支持的查询算子。支持并集的处理器 ∨ 也支持投影与交集（∧）。支持否定的模型（¬）也支持并集、投影和交集。

投影与交集（∧）	并集（∨）	否定（¬）	克莱尼 +	过滤器与聚合
GQE (Hamilton 等人，2018年) 哈希 GQE (Wang 等人，2019年) CGA (Mai 等人，2019年) TractOR (Friedman & Broeck，2020年) MPQE (Daza 和 Cochez，2022年) BiQE (Kotnis 等人，2021年) 层 (Gebhart 等人，2021年) StarQE (Alivanistos 等人，2022年) RotatE-m、DistMult-m、ComplEx-m (Ren 等人，2022年) GNNQ (Pflueger 等人，2022年)	Query2Box (Ren 等人，2020年) EmQL (Sun 等人，2020年) Query2Onto (Andresel 等人，2021年) HypE (Choudhary 等人，2021b) NewLook (Liu 等人，2021年) PERM (Choudhary 等人，2021a) CQD (Arakelyan 等人，2021年) kgTransformer (Liu 等人，2022年) Query2Geom (Sardina 等人，2023年)	BetaE (Ren 与 Leskovec，2020年) ConE (Zhang 等人，2021b) LogicE (Luus 等人，2021年) MLPMix (Amayuelas 等人,2022年) Query2Particles (Bai 等人，2022a) LinE (Huang 等，2022b) GammaE (Yang 等,2022a) NMP-QEM (Long 等，2022年) FuzzQE (Chen 等人，2022年) TAR (Tang 等人，2022年) GNN-QE (Zhu 等人，2022年) FLEX (Lin 等人，2022b) TFLEX (Lin 等人，2022a) ENeSy (Xu 等，2022年) QTO (Bai 等人，2022b) SignalE (Wang 等人，2022a) LMPNN (Wang 等人，2023c) NQE (Luo 等，2023年) Var2Vec (Wang 等人，2023a) CQD^A (Arakelyan 等人，2023年) SQE (Bai 等人，2021年)	RotatE-Box (Adlakha 等人,2021年)	None

我们从最简单的合取查询开始，这类查询仅涉及存在量化（∃）和合取（∧），并逐步增加所支持算子的复杂度。

存在量化（∃）。当 ∃ 出现在查询中时，这意味着至少存在一个存在量化变量。例如，给定查询“图灵奖获奖者在哪些大学工作？”及其逻辑形式 $q = U_? . \exists V$: win(TuringAward, V) ∧ university($V, U_?$)，其中 V 是存在量化变量。查询处理器通过使用关系投影算子来建模存在量化。映射到 SPARQL 等查询语言时，关系投影等同于包含一个变量的三元组模式，例如 `{TuringAward win ?v}` （图18）。通常，正如第5.2节所述，查询嵌入方法以自底向上的方式嵌入查询：从锚点（叶节点）的嵌入开始，逐步沿查询树向上遍历至根节点。通过这种方式，查询嵌入方法（例如几何方法或概率方法）会显式地为存在量化变量生成嵌入。这些变量的嵌入/表示由关系投影函数计算，该函数通过浅层向量运算（Hamilton 等，2018；Ren 等，2020；Choudhary 等，2021b；Arakelyan 等，2021；Ren 等，2022）或深度神经网络（Ren 与 Leskovec，2020；Zhang 等，2021b；Bai 等，2022a；Amayuelas 等，2022）实现。另一类基于图神经网络和 Transformer 的方法，则直接为存在量化变量分配可学习的初始嵌入。随后，这些嵌入会在查询计划上经过若干消息传递层更新（Daza 与 Cochez，2020；Alivanistos 等，2022；Pflueger 等，2022；Wang 等，2023c），或在序列化查询计划上经过若干注意力层更新（Kotnis 等，2021；Liu 等，2022）。

现有神经查询处理器的主要缺点是，假设查询中至少有一个锚实体，求答过程从该实体开始，并可执行关系投影。如何支持不含锚实体的查询仍是一个尚待解决的挑战，也是未来的研究方向，例如， $q_1 = U_?.\exists v_1, v_2$： 获胜($v_1$， v_2) ∧ 大学(v_2， $U_?$)；以及关系为变量的查询，例如，$q_2 = r_?$： $r_?$(TuringAward， Bengio)，这类查询可以表述为关系预测任务。

全称量化（∀）。全称量化变量 $\forall x.P(x)$表示逻辑公式 $P(x)$对所有可能的 x均成立。通常，在不包含类型和复杂公理的仅事实 ABox 图中（见第 4.3 节），全称量词不会出现，因为它会意味着某个实体与所有其他实体相连。例如， $\forall V$.win(TuringAward, V)在没有其他约束的情况下，意味着所有实体都通过 win 关系与 TuringAward 相连（这在实践中并不会发生）。然而，全称量词与类层次结构（一元关系）搭配使用时更有用，例如，∃?paper ∀$r \in$ Researcher： Researcher(r) ∧authored(r, ?paper)表示， authored 关系投影只会应用于类别为 Researcher的实体。

目前，现有的 CLQA 方法既不明确支持全称量词，数据集也不包含带有 ∀量词的查询。不过，利用基本恒等式 $\forall x.P(x) \equiv \neg(\exists x.\neg P(x))$，任何支持存在量化 ∃ 和否定 ¬的方法都可以对全称量词进行建模。默认情况下，我们在神经图数据库中采用封闭世界假设。关于全称量化在开放世界假设（OWA）下所涉及的影响，则不在本文的讨论范围内。

合取（∧）。我们将在以下第 6.2 节中详细阐述这些查询模式之间的差异，并强调我们关注的是更复杂的交集查询，这些查询超越了更简单的类路径查询。
如第5.2节所述，查询处理器采用不同的合取参数化方式，将其实现为置换不变集合函数。一类神经处理器（Hamilton 等，2018；Wang 等，2019；Ren 等，2022）通常采用 DeepSet 架构（Zaheer 等，2017）：首先分别投影集合中的每个元素，然后通过置换不变函数（例如求和、求均值）将表示池化，最后接入多层感知机。或者，自注意力（Mai 等，2019；Bai 等，2022a）也可以替代 DeepSet，通过注意力操作为集合元素赋予不同权重。另一类神经处理器通过使用图神经网络处理完整查询图（Daza 与 Cochez，2020；Alivanistos 等，2022；Pflueger 等，2022；Wang 等，2023c），或使用 Transformer 处理线性化查询序列（Kotnis 等，2021；Liu 等，2022），将投影与交集结合起来。几何处理器（Ren 等，2020；Liu 等，2021；Choudhary 等，2021b；Zhang 等，2021b）将合取实现为相应几何对象（盒体、双曲面或圆锥体）的质心和偏移量的基于注意力的平均。概率处理器（Ren 与 Leskovec，2020；Choudhary 等，2021a；Huang

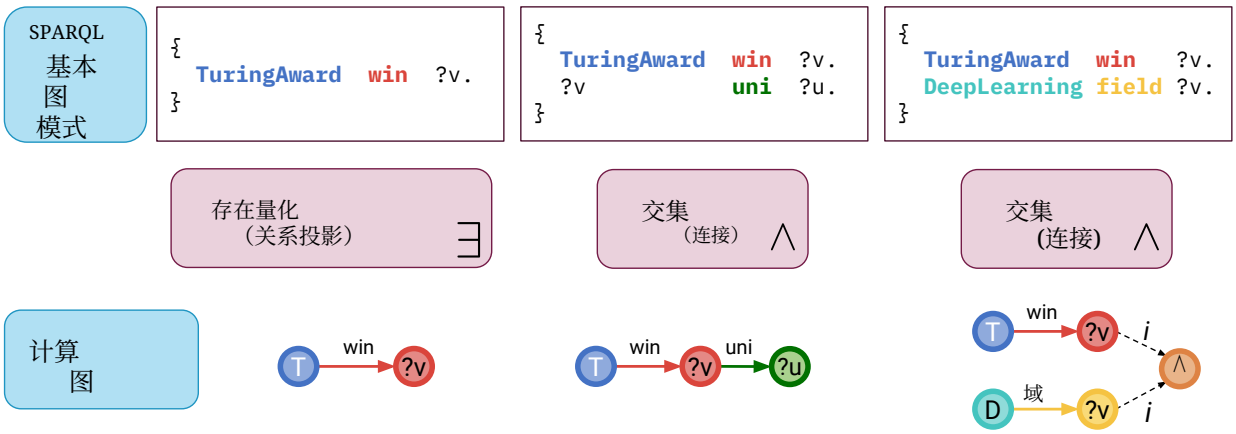


图 18：查询算子（关系投影与交集）、对应的 SPARQL 基本图模式（BGP）及其计算图。关系投影（左）对应一个三元组模式。

等人，2022b；Yang 等人，2022a）将交集实现为代表查询和变量的参数化分布的加权和。

模糊逻辑处理器（Arakelyan 等，2021；Luus 等人，2021；Chen 等人，2022b；Zhu 等人，2022）通常采用 t-范数（连续 $[0,1]$ 空间中合取的广义形式）及其对应的 t-余范数来建模并集（第2.6节）。通常，由于缺乏权威研究，模糊逻辑的选择属于超参数。我们认为，这类研究是模糊处理器未来工作的重要方向。用于建模合取的、更为新颖的神经符号方法包括计数最小草图的逐元素乘积（Sun 等，2020），或在特征逻辑中采用加权和（Lin 等人，2022b；a）。最后，一些处理器（Tang等，2022；Xu 等，2022）同时在嵌入空间和符号空间中，使用神经算子和模糊算子执行合取。

我们注意到，某些直接嵌入查询（Hamilton 等，2018年；Wang 等，2019年；Mai 等，2019年；Friedman 与 Broeck，2020年；Ren 等，2022年），或通过在查询图上使用 GNN/Transformer编码器嵌入查询（Daza 和 Cochez，2020年；Kotnis 等，2021年；Gebhart 等，2021年；Alivanistos 等，2022年；Das 等，2022年；Pflueger 等，2022年）的神经查询处理器，仅支持投影和交集。也就是说，它们扩展到更复杂逻辑运算符的过程并非易事，可能需要改变对实体、变量和查询进行建模时所依据的基本假设。在某些情况下，当将查询重写为析取范式时，可能可以支持并集（下文将对此进行讨论）。

析取（$\vee$）。查询处理器以多种方式实现析取算子。然而，析取的建模一直以来都非常困难，因为这要求在向量空间中对图上实体的任意幂集进行建模[7] 。在深入探讨析取的不同建模方式之前，我们首先请读者参阅 Query2Box 中的定理 1 （ Ren 等人，2020 年）。该定理证明，距离函数的函数类的 VC 维数需要大致等于图中的实体数量。

该定理表明，要使用现有框架准确建模任意EPFO查询，距离函数的复杂度（以 VC 维数衡量）必须至少与知识图谱实体的数量一样大。这意味着，如果使用基于超平面、欧几里得球或轴对齐矩形的常见距离函数，[8] 其参数维度就需要达到 $\Theta(M)$，而对于我们关注的真实知识图谱，这一数值为 $\Theta(|\mathcal{E}|)$。换言之，逻辑查询嵌入的维度需要达到$\Theta(|\mathcal{E}|)$，这并非低维表示；因此，它无法扩展到大型知识图谱，也无法在存在未观测到的知识图谱边时实现良好的泛化。

[7]这里最严谨的描述应当是图中“同构”实体集合的幂集。所谓“同构”实体，是指这样一个实体集，其中每个元素[8]关于这些函数类的详细 VC 维度，请参见Vapnik (2013年)。至关重要的是，它们的 VC 维度都与参数数量呈线性关系 d。

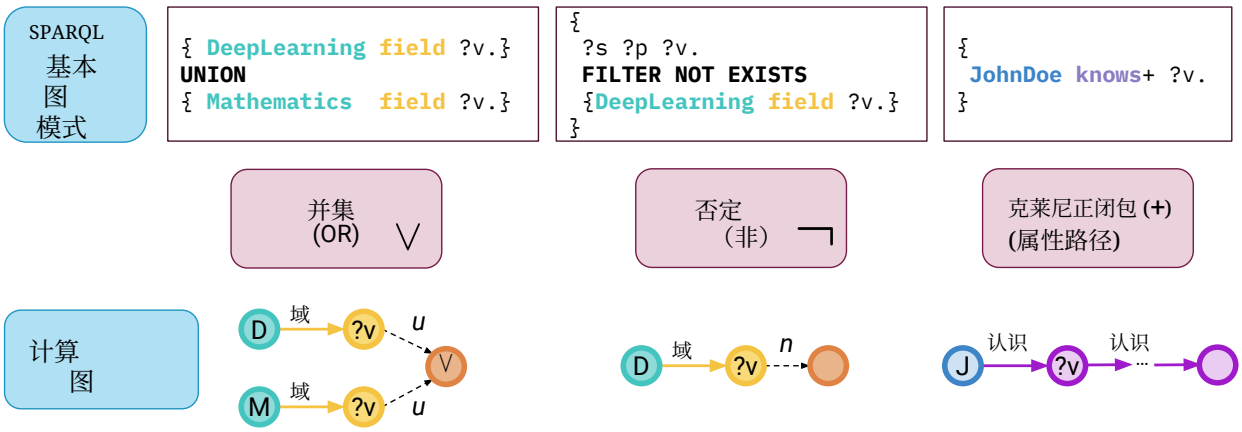


图19：查询算子（并集、否定、克莱尼正闭包）、对应的 SPARQL 基本图模式（BGP）及其计算图。

Query2Box（Ren 等，2020）提出的第一个思想是：给定一个模型已定义查询表示与实体表示之间的距离函数，那么查询就可以转换（或重写）为其等价的析取范式（DNF），即合取查询的析取。例如，我们可以安全地将查询 $(A \vee B) \wedge (C \vee D)$ 转换为 $((A \wedge C) \vee (A \wedge D) \vee (B \wedge C) \vee (B \wedge D))$，其中 A、B、C、D 是原子公式。这样一来，我们只需在最后一步处理析取 $\vee$。对于已定义查询表示与实体表示之间距离函数 d(q, e)的模型（例如几何处理器（Ren 等，2020；Andresel 等，2021；Adlakha 等，2021；Choudhary 等，2021b；Zhang 等，2021b；Sardina 等，2023）以及一些神经处理器（Liu 等，2022；Amayuelas 等，2022；Wang 等，2023c）），使用析取范式处理析取的思路是：（1）将析取范式中的每个原子公式/合取查询嵌入为向量$\mathbf{q}_i$；（2）计算每个原子公式/合取查询的表示/嵌入与实体 $d(\mathbf{q}_i, \mathbf{e})$之间的距离；（3）取距离的最小值$\min_i(d(\mathbf{q}_i, \mathbf{e}))$。其直觉在于，由于析取建模的是并操作，只要节点接近某个原子公式/合取查询，它就应当接近整个查询。理论上，许多已定义距离函数、原本仅支持交集和投影的神经处理器，都可以借助析取范式扩展为支持并集。这种建模的一个显著缺点是：在最坏情况下，将查询转换为其析取范式的开销会随析取数量呈指数级增长。

另一类主要是概率处理器（Choudhary 等人，2021a；Yang 等，2022a）提出了一种神经析取算子，通过对输入集合采用置换不变注意力来实现，并基于闭包假设，认为注意力加权后的并集结果仍处于与其输入相同的概率空间中。这些模型采用一种更为黑盒的框架来处理析取操作，其依据的强闭包假设并非在所有情况下都成立。

第三种对析取进行建模的方法基于德摩根定律（Ren 与 Leskovec，2020年）。根据德摩根定律（DM），析取等价于对构成该析取的语句先取否定、再对其合取取否定，即，$A \vee B = \neg(\neg A \wedge \neg B)$。对于能够处理否定运算符的方法（详见下一段），它们通过三次否定操作和一次合取操作来对析取建模。概率（Ren 与 Leskovec，2020年）、几何（Zhang 等人，2021b）和模糊（Luus 等人，2021年）处理器均明确测试了 DM 转换。

最后，大多数模糊逻辑处理器（Arakelyan 等，2021；Chen 等，2022b；Tang 等，2022；Zhu 等人，2022；Bai 等，2022b；Arakelyan 等，2023年；Wang 等，2023a）采用 t-余范数，即连续 $[0,1]$ 空间中析取的广义形式（第2.6节）。更为特殊的神经符号析取形式包括对 Count-Min 草图逐元素求和（Sun 等，2020）、特征逻辑运算（Lin 等人，2022b;a），以及借助模糊算子在嵌入空间和符号空间中同时执行并集（Tang 等，2022；Xu 等，2022）。

否定（¬）。对于否定操作，目标是对补集进行建模，即查询 $q = V_?: \neg r(v, V_?)$ 的答案 A_q，恰好是查询 $q' = V_?: r(v, V_?)$ 的答案 $A_{q'}$ 的补集：$A_q = V/A_{q'}$。相应地，SPARQL 中的否定可以通过 `FILTER NOT EXISTS` 或`MINUS` 子句实现。例如（图19），包含否定 ¬域(DeepLearning, V)的逻辑公式等价于 SPARQL BGP `{?s ?p ?v. FILTER NOT EXISTS {DeepLearning field ?v}}`，其中 `{?s ?p ?v}` 对所有事实构成的全集 (1) 建模，再由三元组模式进行过滤。

在设计神经查询处理器中的否定运算符时，如何建模全集(1)及其补集是关键问题。例如，任意实数 $\mathbb{R}$ 或复数 $\mathbb{C}$ 空间都是无界的，因此1 未定义。正因如此，许多神经处理器不支持否定运算符。不过，也存在几种处理否定的方法。

第一类工作（Bai 等人，2022a；Amayuelas 等人，2022年；Long 等，2022年）设计了一种完全基于神经网络的多层感知机否定算子，作用于查询表示，从而完全避开全集。同样，也可以将否定运算符的标记纳入线性化查询表示（Bai 等人，2023年），并使用 Transformer 或循环网络对其进行编码。基于 GNN 的处理器（Wang 等人，2023c）则采用了有别于纯神经算子的做法：在查询计算图上进行消息传递时，将否定边视为一种新的边类型。

第二行针对不同的嵌入空间进行了定制，旨在模拟嵌入空间中全集与补集的计算，例如，使用几何锥（Zhang 等人，2021b），参数为角度 θ，使得该空间（因此，1）被限制在 2π，且补集为直线 $2\pi - \theta$。概率方法（Ren 与 Leskovec，2020年；Huang 等，2022b；Yang 等，2022a）则自然地将否定表示为分布参数的逆。

第三，模糊逻辑处理器在同一个实值逻辑空间中显式地对全集 1 和补集进行建模。例如，LogicE（Luus 等人，2021年）和 FuzzQE（Chen 等人，2022b）将查询嵌入空间限制在值域 $[0,1]^d$，其中每个查询 $\mathbf{q} \in [0,1]^d$ 都是一个向量。这样，全集 1 就用一个全为 1 的向量表示（位于嵌入空间 $\mathbf{1}^d$中），而否定就是简单的 $\mathbf{1} - \mathbf{q}$。TAR（Tang 等人，2022年）和 GNN-QE（Zhu 等人，2022年）在模糊集上运行，其中每个实体都有一个对应的标量 $\mathbf{q} \in [0,1]$，其位于有界值域内。因此，全集 1 仍然可以用一个全为 1 的向量表示（位于实体空间 $\mathbf{1}^{|\mathcal{E}|}$中），而否定就是 $\mathbf{1} - \mathbf{q}$。ENeSy（Xu 等，2022年）将全集定义为实体空间上的均匀分布，其中每个元素的权重为 $|\mathbf{E}|^{\alpha}$（α 是一个超参数）。CQD$^{\mathcal{A}}$（Arakelyan 等人，2023年）采用严格余弦模糊否定 $1 + \cos(\pi \mathbf{q})$）对标量分数$\mathbf{q}$。更为复杂的处理器（Lin 等人，2022b；a）采用特征逻辑对否定进行建模。我们还注意到，Liu 等人（2021年）引入的差算子实际上是一种常见的交集否定（2in）查询模式，所有标准基准测试（第7节）中都使用了这种模式。

克莱尼正闭包（+）和属性路径。克莱尼正闭包是一种算子，可组合地递归应用于任何表示指定模式出现一次或多次的正则表达式（RegEx）。正则表达式与 SPARQL 中的属性路径有直接联系。我们在定义 2.12 中定义了一个非常基本的正则图查询；此处我们将其进一步泛化为属性路径。为了更正式地定义属性路径，给定一个关系集合 R 和算子 $\{+, *, ?, !, \hat{}, /, |\}$，可通过递归文法 $p ::= r \mid p^+ \mid p^* \mid p? \mid !p \mid \hat{}p \mid p_1/p_2 \mid “p_1|p_2”$ 得到属性路径 p。其中，r 是 R 中的任意元素；$+$ 表示克莱尼正闭包，即出现一次或多次；$*$ 表示 Kleene 星号，即出现零次或多次；? 表示出现零次或一次；! 表示对关系或路径取否定；$\hat{}p$ 沿相反方向遍历类型为 p 的边；p_1/p_2 表示关系序列（对应关系投影）；$p_1|p_2$ 表示 p_1 或 p_2 的替代路径（对应并操作）。例如（图19），带有克莱尼正闭包的表达式 knows(JohnDoe, V)$^+$ 可表示为 SPARQL 属性路径 `{JohnDoe knows+ ?v.}`。

由于组合性和递归特性，神经查询处理器难以对属性路径进行建模。据我们所知，RotatE-Box（Adlakha 等人，2021年）是唯一能够处理克莱尼正闭包算子的几何处理器，其实现的算子子集为 $\{+, /, |\}$。RotatE-Box 提供了两种处理克莱尼正闭包的方法。第一种方法是为每个关系 $\mathbf{r}^+$ 嵌入 $r \in \mathcal{R}$，注意这与关系 r的常规关系嵌入相互独立且彼此分开；另一种方法是使用可训练矩阵将关系嵌入 $\mathbf{r}$ 变换为 $\mathbf{r}^+$ 嵌入。请注意，这两种方法确实

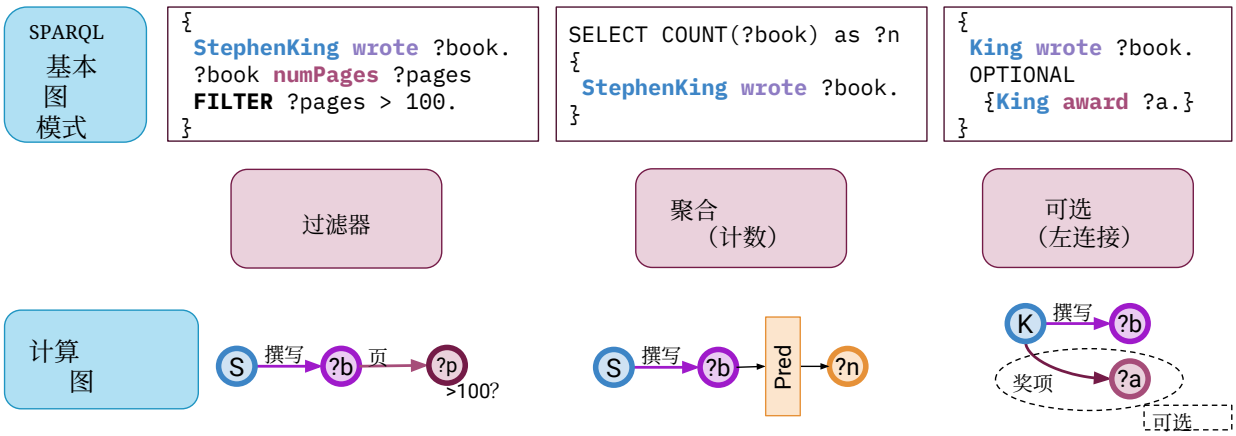


图20：查询算子（Filter、 Count 聚合、 Optional）、对应的 SPARQL 基本图模式（BGP）及其计算图。

不支持对路径应用克莱尼正闭包。RotatE-Box 也实现了关系投影（即在复数空间中的旋转）和并集（使用 DeepSets 或析取范式），但不支持交集运算符。

我们假设，更好地支持属性路径词汇可能是未来神经查询处理器的主要关注点之一。特别是对于克莱尼正闭包，一些尚未解决的问题包括支持幂等性（$(r^+)^+ = r^+$）和集合的无限并集（$r^+ = r|(r/r)|(r/r/r)\ldots$）。

过滤器。 过滤器是一种可插入 SPARQL 查询的操作。它以任意布尔类型的表达式作为输入，并旨在根据布尔值过滤结果，即，只有经表达式求值为 True的结果才会被返回。因此，布尔表达式可被视为查询答案应当遵循的条件。对于过滤运算符，我们可以对值/字面量/属性进行过滤，例如， Filter($V_{\text{date}} \geq$ "$2000-01-01$"&&$V_{\text{date}} \leq$ "$2000-12-31$") 表示我们希望过滤掉不在 2000 年的日期； Filter(LANG(V_{book}) = "en") 表示我们希望过滤掉并非用英语写成的书籍，Filter(?页数 $>$ 100) 表示返回页数超过 100 页的书籍（如图 20所示）。据我们所知，目前尚不存在声称能够处理过滤器的推理模型，这为未来在该方向上的工作留下了空间。

我们设想了几种在神经查询引擎中支持过滤的可能方式：（1）由Thorne 等人（2021a；b）在自然语言引擎中采用的最简单方案，是在识别出候选节点集合并可通过查找提取其属性后，将过滤推迟到后处理阶段。（2）过滤通常意味着要对字面量值和数值节点属性进行推理，即支持连续值的处理器（如第4.1节所述）或许能够在潜在空间中执行过滤，例如，在预测第5.3节中的?页数 $>$ 100时附加一个参数回归器解码器。

聚合。聚合是 SPARQL 查询中的一组算子，包括 COUNT （返回元素数量）、 MIN、 MAX、 SUM、 AVG （返回所有元素的最小值/最大值/总和/平均值）、SAMPLE （从集合中返回任意一个样本）。例如（图 20），给定一个三元组模式 {StephenKing wrote?book.}，子句 COUNT （?book） as ?n）返回由 StephenKing撰写的图书总数。

大多数聚合运算符都需要对数值/字面量集合进行推理。这类符号运算长期以来一直被认为是神经模型面临的挑战（Hendrycks 等人，2021年）。如何为数值/字面量设计更好的表示仍然是一个悬而未决的问题。一些神经查询处理器（Ren 等人，2020年；Ren 与 Leskovec，2020年；Zhang 等人，2021b；Zhu 等人，2022年），然而，能够估计答案集基数（包括预测的困难答案），这与 COUNT 对目标投影变量（假设其为实体而非字面量）进行聚合直接对应。例如，GNN-QE（Zhu 等人，2022年）返回一个模糊集，即为每个实体返回一个标量似然值；经过阈值处理后，真实答案数量的平均绝对百分比误差（MAPE）较低

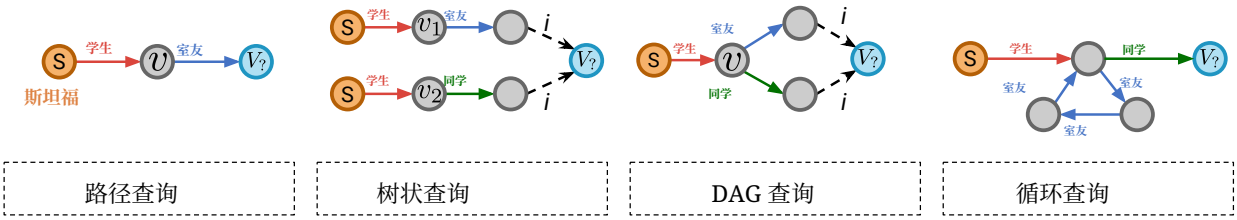


图 21：查询模式：路径、树状模式、有向无环图和循环查询。有向无环图查询从中间变量分出两个分支，循环查询则包含一个 3-环。现有的神经查询处理器支持路径和树状模式。

真实答案。因此，答案基数估计可作为神经查询处理器的副产品获得，而无需专门的预测器。或者，当模型无法预测确切计数时，斯皮尔曼等级相关系数可作为替代指标，用于评估模型预测与确切计数之间的相关性。斯皮尔曼等级相关系数和真实答案数量的平均绝对百分比误差是评估神经查询处理器性能的常用指标，我们将在第7.5节中详细阐述这些指标。

可选项与解答修饰符。 SPARQL 提供了许多尚未纳入神经查询引擎的功能，可用于扩展其表达能力。其中一些常见功能包括 `OPTIONAL` 子句，它本质上是一个 `LEFT JOIN` 算子。例如（图 20），给定一个返回图书的三元组模式 `{King wrote ?book.}`，可选子句 `{King wrote ?book. OPTIONAL {King award ?a.}}` 会将 `King`获得的任何现有奖项信息补充到答案集中。重要的是，即使可选子句没有任何绑定，查询仍会返回 `?book`的值。在查询的计算图中，可选子句对应于关系投影的可选分支。对于在不完整图上运行的神经查询处理器，一个特殊挑战是：图中缺少被查询的边，并不意味着不存在绑定——该边可能只是缺失，并可能在查询处理过程中被预测出来。

解答修饰符，例如，`GROUP BY`、`ORDER BY`、`LIMIT`，会对投影（返回）的结果进行进一步后处理，并且当查询中投影多个变量时尤为重要。到目前为止，所有现有的神经查询处理器都仅针对一个返回变量。我们将在第6.3节中详细讨论这一问题。

关于不完整性的一般说明。最后，我们想强调，执行所有上述算子（**投影、交集、并集、否定、属性路径、过滤器、聚合、可选项**和**修饰符**）的神经查询引擎都假设底层图是不完整的，查询可能存在一些需要预测的缺失答案。因此，所有算子都应纳入通过图遍历可达的困难答案，以及简单答案，后者可通过符号图数据库中的图遍历获得。在不完整性的背景下，使用这些算子评估查询性能仍是一项开放性挑战（我们将在第7.5节中详细讨论），例如，当包含一个可选子句时，可能无法确定不存在真实答案（即使是预测出的答案实际上也是错误的），还是模型无法预测这些答案。

6.2 查询模式

这里，我们介绍实践任务中常用的几种查询模式，并按照复杂度递增的顺序对它们进行排列。首先是文献中多年来广为人知的链式路径查询，随后介绍树结构查询（现代复杂逻辑查询问答系统主要支持的模式）。然后，我们概述有向无环图模式和循环模式；目前，任何神经查询问答系统都不支持这两类模式，它们代表了未来研究的一个重要方向。

路径查询。 如第2.4节所述，先前的研究从路径查询（也称为多跳查询）入手，其目标只是超越单跳查询，例如 $q = V_?.r(v,\ V_?)$，其中 $r \in \mathcal{R}$ $v \in \mathcal{V}$和 $V_?$ 表示答案变量。如图4 和 图21所示，其中不涉及分支交集或并集等逻辑运算符。因此，为了回答这类查询，我们只需找到或推断出

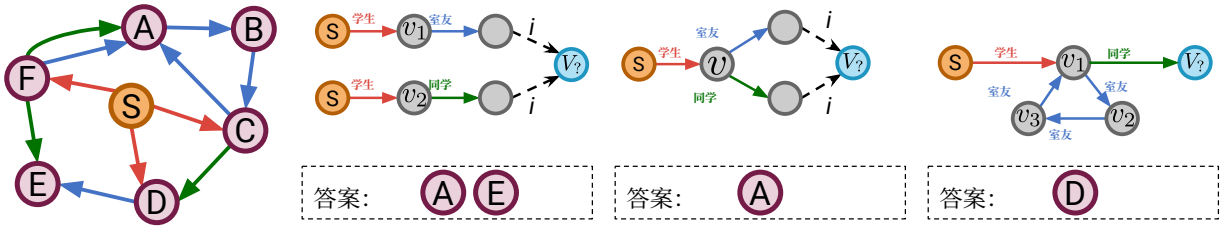


图 22：玩具图中树状、有向无环图和循环查询模式的示例答案。请注意树状查询与有向无环图查询的答案集之间的差异——在有向无环图查询中，变量 v 必须从同一节点引出两条边。

与实体 v 通过关系 r相连的邻居。路径查询是单跳查询的一种自然扩展。形式上，我们将路径查询表示如下。 $q_{\text{path}} = V_?.\exists V_1, \ldots, V_{k-1}: r_1(v, V_1) \wedge r_2(V_1, V_2) \wedge \cdots \wedge r_k(V_{k-1}, V_?)$，其中$r_i \in \mathrm{R}\ \forall i \in [1, k]$, $v \in \mathrm{V}$， V_i均为存在量词化变量。若一个 k跳路径查询包含k 个原子公式，我们就称其为 k跳路径查询。 k跳路径查询的查询计划是一个从锚实体出发、长度为 k 的链。例如（图21），一个2跳路径查询为 $V_?.\exists v: \text{student}(\texttt{Stanford}, v) \wedge \text{roommate}(v, V_?)$，其中 `Stanford` 是起始锚定节点， v 是第一个投影 student 的尾变量，同时也是第二个投影 roommate 的头变量，从而形成一条链。

如定义所示，为了处理路径查询，有必要开发一种能够处理存在量化 $\exists$ 和合取 $\wedge$ 算子的方法。若干查询推理方法（Guu 等人，2015年；Das 等人，2017年）旨在使用序列模型回答路径查询，模型采用可链式连接的知识图谱嵌入（例如，TransE），如 Guu 等人 (2015年) 的方法，或采用 LSTM，如Das 等人 (2017年) 的方法。这些方法开启了早期尝试之一，即利用嵌入和神经方法回答多跳路径查询。我们肯定该领域的相关努力，但也强调这些方法在查询表达能力方面的局限性。因此，本研究将重点放在能够处理树状及更复杂模式、表达能力更强的查询回答方法上。

树结构查询。路径查询只有一个锚实体和一个答案变量。这类查询的表达能力有限，与现实世界知识图谱（如维基数据）的日志（Malyshev 等，2018年）中所体现的现实世界查询复杂度相去甚远。提高表达能力和复杂度的一种直接扩展方式，是支持树结构（树状）查询。树状查询可能包含多个锚实体，不同的分支（来自不同锚点）会在最终的单个答案节点处合并，从而形成树结构查询计划。这种合并可以通过交集、并集或否定算子实现。例如，如图1所示，“深度学习领域的图灵奖获奖者在哪些大学任职？”这一查询的查询计划不是一条路径，而是一棵树。作为替代方案，图21中的示例描述了一个查询$q = V_?, \exists v_1, v_2:$ $\text{student}(\texttt{Stanford}, v_1) \wedge \text{roommate}(v_1, V_?) \wedge \text{student}(\texttt{Stanford}, v_2) \wedge \text{classmate}(v_2, V_?)$，它由两个长度为2跳的路径查询分支组成，并在末端通过交集运算符连接。

树状查询给此前只能处理路径（多跳）查询的模型带来了更多挑战，因为序列模型不再适用于包含逻辑运算符的树状执行计划。鉴于这些挑战，神经与神经符号查询处理器（详见第 5 节）旨在执行更复杂的查询模式。这些处理器设计了神经集合/逻辑算子，并对该树执行自底向上遍历，直至单个根节点。

任意有向无环图。基于树结构查询，我们可以进一步将查询模式的复杂度提升至任意有向无环图（DAG）。这两类查询的关键区别在于，对于 DAG 结构查询，查询计划中的一个变量节点（代表一组实体）可能会分裂并被引导至不同的推理路径，而查询计划中的分支/推理路径数量始终从锚节点到答案节点逐渐减少。我们在图21和图22中展示了一个例子。考虑上一段中的树状查询 $q_1 = V_?, \exists v_1, v_2: \text{student}(\texttt{Stanford}, v_1) \wedge \text{roommate}(v_1, V_?) \wedge \text{student}(\texttt{Stanford}, v_2) \wedge \text{classmate}(v_2, V_?)$，以及 DAG 查询$q_2 = V_?, \exists v: \text{student}(\texttt{Stanford}, v) \wedge \text{roommate}(v, V_?) \wedge \text{classmate}(v, V_?)$。这两个查询都在寻找 $V_?$ 谁

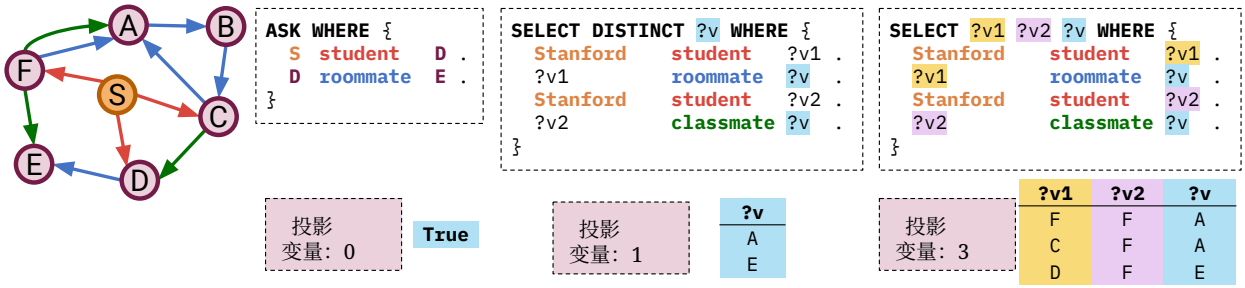


图23：来自图22的树形查询的投影变量。当前神经查询处理器支持单变量 DISTINCT 模式（中间），而查询也可能有零个返回变量，类似于子图匹配布尔 ASK 查询（左侧），或者有多个投影变量（右侧），这意味着需要返回中间答案并形成输出元组。

与 Stanford 学生既是室友又是同学。不过，这两个查询的答案集不同（如图22所示）。也就是说，DAG 查询V_{q_2} ={A}的答案是树状查询 V_{q_1} ={A,E}答案的子集，因为 q_2 必须与中间变量 v中的同一位斯坦福学生既是室友又是同学。另一方面，树状查询 q_1 的两个分支彼此独立，因此中间变量 v_1 和 v_2 不必是同一实体；所以，该查询有更多有效的中间答案，也有更多正确答案。据我们所知，目前仍不存在能够忠实处理任意 DAG 查询的神经查询处理器。尽管 BiQE（Kotnis 等，2021）声称支持 DAG 查询，但其挖掘数据集由树状查询构成。尽管如此，我们推测，能够考虑整个查询图结构 G_q 的消息传递处理器或 Transformer 架构，或许能够处理 DAG 查询；我们将这一问题留待未来研究。

循环查询。循环查询比 DAG 结构查询更复杂。查询中的环意味着遍历查询计划时并不存在特定的顺序。图21和图22展示了一个循环查询示例：$q = V_?.\exists v_1.\exists v_2.\exists v_3 : \text{student}(\texttt{Stanford}, v_1) \wedge \text{roommate}(v_1, v_2) \wedge \text{roommate}(v_2, v_3) \wedge \text{roommate}(v_3, v_1) \wedge \text{classmate}(v_1, V_?)$。在 q中，三个变量构成一个三角循环：$\text{roommate}(v_1, v_2) \wedge \text{roommate}(v_2, v_3) \wedge \text{roommate}(v_3, V_?)$。给定图22中的图，该环从节点 C出发并最终回到该节点，因此，只有在从 C 执行同学关系投影并以 D、 V_q ={D}结束后，才能得到唯一正确的答案。

因此，假设查询计划具有特定遍历方式或节点排序的推理方法和查询处理器，无法如实地回答循环查询。如何有效地对具有循环结构的查询进行建模，仍是一个悬而未决的问题。此外，在处理带有正则表达式和属性路径（第6.1节）的查询时，循环结构也经常出现。我们认为，支持环可能是神经查询引擎全面支持属性路径的必要条件。

6.3 投影变量

所谓投影变量，是指必须绑定到特定图元素（如实体、关系或字面量）的目标查询变量。例如，图 1 $q = V_?.\exists v$ 中的查询：$\text{win}(\texttt{TuringAward}, v) \wedge \text{field}(\texttt{DeepLearning}, v) \wedge \text{university}(v, V_?)$，包含一个投影变量 $V_?$，它可以绑定到图中的三个答案节点： $V_? = \{\texttt{UofT}, \texttt{UdeM}, \texttt{NYU}\}$。在 SPARQL 文献（Hawke 等，2013）中， SELECT查询规定了要将哪些存在量化变量投影为最终答案。投影变量与答案的配对构成绑定，作为 SELECT 查询的结果。通常，查询可能包含零个、一个或多个投影变量，我们也据此进行分类。此类查询及其可能的答案示例见图 23。目前，大多数神经查询处理器都专注于仅包含一个答案变量的查询场景——即计算图的叶节点，如图 23（中）所示。

零投影变量。 零投影变量的查询不返回任何绑定，而是探测图中是否存在某个特定子图或关系模式，此时答案为布尔值True 或 False。在 SPARQL 中，与零变量查询等价的是 ASK 子句。零变量查询中的所有实体和关系都可能用常量实例化，例如， q = 学生(S, D) ∧ 室友(D, E)如图 23（左）所示，等价于 SPARQL 查询 ASK WHERE {S student D. D roommate E.}。该查询探测图中是否包含由这些常量诱导出的特定子图（路径）。这样的路径确实存在，因此答案为 q ={True}。

或者，零变量查询也可能包含从未被投影的存在量化变量（但所有主语、谓语或宾语均为变量的情况除外）。例如，查询 $q_1 = \exists v_1, v_2, v_3 : 学生(v_1, v_2) \wedge 室友(v_2, v_3)$ 用于探查是否存在构成关系路径 $v_1 \xrightarrow{学生} v_2 \xrightarrow{室友} v_3$ 的节点。一般情况下，查询 $q_2 = \exists p, s, o : p(s, o)$ 则询问图中是否至少包含一条边。

我们注意到，在本文主要考虑的不完整图和缺失边这一设定下，零变量查询仍然不易求答。尤其是，神经子图匹配 (Rex 等人，2020年；Huang 等，2022a) 这一子领域涉及不完整图。我们推测，这类方法或许有助于神经查询处理器支持零变量查询的求答。

单投影变量。 具有一个投影变量的查询会返回一个（可能存在多个）存在量化变量的绑定。在 SPARQL 中，投影变量在 SELECT 子句中指定，例如，SELECT DISTINCT ?v 在图 23 （中）所示。尽管 SPARQL 允许投影查询中任意部分的变量，但第 5 节介绍的大多数神经查询引擎都遵循 GQE（Hamilton 等人，2018年）的任务设定，并且只允许将叶节点作为投影目标变量，即查询计算图中的**叶节点**。图图 23 （中）展示了这一限制，其中目标变量? v 是查询图的叶节点，并有两个绑定 v = {A, E}。

值得注意的是，现有的神经查询处理器旨在为输入查询返回一组唯一的答案，也就是说，这对应于 SPARQL 中的 SELECT DISTINCT 子句。相比之下，默认的 SELECT会返回多重集，其中可能包含重复项。例如，若去掉图 23 （中）中的 DISTINCT，同一个查询会得到绑定 v = {A, A, E}，因为存在两个匹配的图模式以 A结尾。实现非DISTINCT 查询求答仍然是一项尚待解决的挑战。

大多数神经查询处理器都具有中间变量的概念，并对其在实体空间中的分布进行建模。例如，几何处理器在定义了距离函数后（Ren 等人，2020年；Choudhary 等人，2021b）可以找到作为中间变量的最近实体。类似地，基于模糊集运行的模糊逻辑处理器（Tang 等人，2022年；Zhu 等人，2022年）在每个执行步骤后，都会维护一个覆盖所有实体的标量分布。最后，基于图神经网络（Daza 和 Cochez，2020年；Alivanistos 等人，2022年）和基于 Transformer 的（Liu 等人，2022年）处理器会明确地将中间变量作为节点纳入查询图（或作为词元纳入查询序列），因此能够将其表示解码到实体空间。所有这些方法的主要缺点在于：叶节点确定后，缺乏对中间变量集合进行过滤的机制之后。也就是说，为了过滤并投影出仅能导向最终答案的中间变量，需要某种形式的反向传播 。朝这一方向迈出的第一步是 QTO（Bai 等人，2022b）：它在到达答案叶节点后执行剪枝反向传播。

多个投影变量。查询最一般、最复杂的情形是包含多个投影变量，如图 23（右）所示。在 SPARQL 中，所有投影变量都在 SELECT 子句中指定（也可以通过 SELECT * 投影查询中的所有变量）。在逻辑形式中，查询包含多个目标变量 q =?v_1、?v_2、?v : student(Stanford, ?v_1) ∧ roommate(?v_1, ?v) ∧student(Stanford, ?v_2) ∧ classmate(?v_2, ?v)，其输出绑定以元组形式组织。例如，一个可能的答案元组是 {?v_1 : F, ?v_2 : F, ?v : A}，表示满足该查询模式的特定节点（变量绑定）。

如前一段关于单变量查询的讨论所示，一些神经查询处理器具备跟踪中间变量的能力。然而，它们都无法构造包含变量绑定的答案元组，而如何将多个投影变量纳入此类处理器仍然是一个尚未解决的挑战。此外，需要考虑的一些常见问题包括：（1）处理

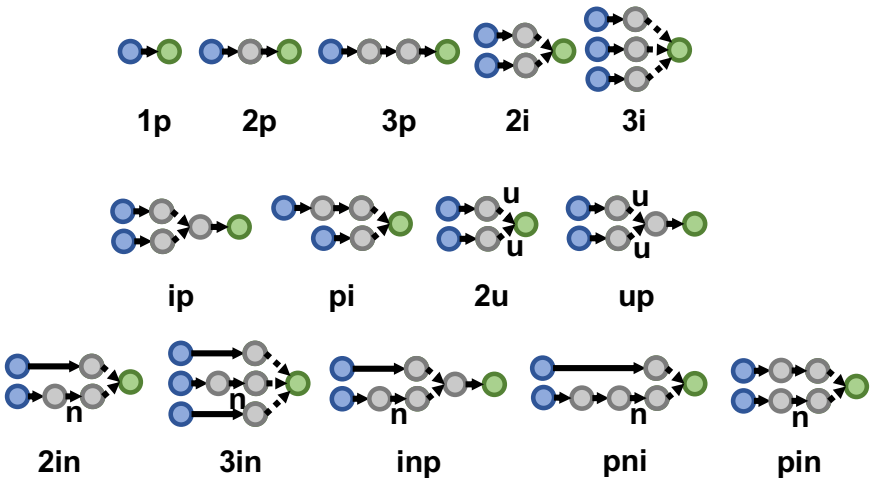


图24：标准查询模式及其名称，其中p 表示投影，i 表示交集，u 表示并集，n 表示否定。在一个模式中，蓝色节点表示非变量实体，灰色节点表示变量节点，绿色节点表示答案节点。在典型的训练协议中，模型使用10种模式（第一行和第三行）进行训练，并在所有模式上进行评估。在最难的泛化情形中，模型仅使用$1p$ 查询进行训练。一些数据集还会通过添加限定符或时间戳等特征来进一步修改这些模式。

经常出现的未绑定变量，例如在 OPTIONAL 第6.1节涵盖的查询中，答案元组可能针对某些变量包含空值（∅ 或 NULL）；（2）复杂度不断增加的问题，因为答案集的规模可能取决于投影变量的数量，并且有可能达到多项式规模。

7 数据集和指标

7.1 评估设置

已有多个数据集被提出，用于评估查询推理模型。这里我们介绍 CLQA任务的通用设置。给定一个知识图谱 G = (E, R, S)，标准做法是将 G 划分为训练图 G_{train}、验证图 G_{val} 和测试图 G_{test} （模拟第2节中的未观测完整图 ˆG）。标准实验协议是仅使用训练图 G_{train}训练查询推理模型，并评估该模型在验证图 G_{val} 和测试图 G_{test}上回答查询的能力。给定一个查询 q，将该查询在训练图、验证图和测试图上的答案分别记为 $[\![q]\!]_{\text{train}}$、 $[\![q]\!]_{\text{val}}$ 和 $[\![q]\!]_{\text{test}}$。评估期间，查询可能存在缺失答案，例如，验证查询 q 可能有一些不在 $[\![q]\!]_{\text{train}}$中的答案 $[\![q]\!]_{\text{val}}$ ，测试查询 q 可能有一些不在 $[\![q]\!]_{\text{val}}$中的答案$[\![q]\!]_{\text{test}}$ 。CLQA任务的总体目标是找到这些缺失答案。典型训练查询、训练协议、推理和评估指标的详情分别在第7.2节、第7.3节、第 7.4 节和第7.5节中介绍。

7.2 查询类型

许多数据集使用的标准图查询集合包含 14 种类型：$1p/2p/3p/2i/3i/ip/pi/2u/up/2in/3in/inp/pni/pin$ 其中p 表示关系投影，i 表示交集，u 表示并集，n 表示否定，数字表示投影查询的跳数，或由逻辑运算符合并的分支数。图24 展示了常见的查询模式。例如，$3p$ 是由三个连续的关系投影构成的链式查询，$2i$ 是两个关系投影的交集，$3in$ 是三个关系投影的交集，其中一个分支包含否定，up 是两个关系投影的并集，后接另一个投影。最初的 GQE 由Hamilton 等人(2018年)提出

提出了 7 种包含投影与交集的查询模式 $1p/2p/3p/2i/3i/ip/pi$，Query2Box（Ren 等人，2020年）增加了并集查询 $2u/up$，而 BetaE（Ren 与 Leskovec，2020年）则新增了五种包含否定的类型。

后续工作从多个方面修改了标准查询类型集合，例如，超关系查询（Alivanistos 等人，2022年；Luo 等，2023年）在关系投影上加入实体-关系限定符，或在边上加入时间算子（Lin 等人，2022a）。新的查询模式包括使用关系正则表达式（属性路径）的查询（Adlakha 等人，2021年）、更多树状查询（Kotnis 等人，2021年），以及投影、交集和并集的更多组合（Wang 等人，2021年；Pflueger 等人，2022年）。我们在表 15中总结了现有查询回答数据集及其属性，涵盖支持的查询算子、推理设置，以及时间戳、类层次结构或复杂本体公理等附加特征。

通常，查询数据集从不同的知识图谱中采样，以研究模型在不同图分布下的性能。例如，BetaE 数据集包含来自较稠密的 Freebase（Bollacker 等人，2008年）的查询集合，其平均节点度为 18；也包含来自较稀疏的 WordNet（Miller，1998年）和 NELL（Mitchell 等人，2015年）的查询集合，其平均节点度为 2。超关系数据集 WD50K（Alivanistos 等，2022）和 WD50K-NFOL（Luo 等，2023）采样自维基数据（Vrandecic 和 Krötzsch，2014年），其中限定符十分常见。带有类层次结构的 TAR 数据集（Tang 等人，2022）采样自 YAGO 4（Pellissier Tanon 等人，2020年）和 DBpedia（Lehmann 等人，2015年），这些数据集中的类层次结构经过了精心整理。带有本体公理的 Q2B Onto 数据集（Andresel 等人，2021）采样自 LUBM（Guo 等人，2005年）和 NELL。时间型 TFLEX 数据集（Lin 等，2022a）采样自 ICEWS（Boschee 等人，2015年）和 GDELT（Leetaru 和 Schrodt，2013年），这些数据集保留了事件信息。InductiveQE 数据集（Galkin 等，2022b）采样自 Freebase 和维基数据，而归纳式 GNNQ 数据集（Pflueger 等，2022）采样自 WatDiv基准（Aluç 等人，2014年）和 Freebase。

7.3 训练

查询推理方法在给定的$\mathcal{G}_{train}$ 上进行训练，采用不同的目标函数/损失和不同的数据集。遵循标准协议，这些方法在 10 种查询模式$1p/2p/3p/2i/3i/2in/3in/inp/pni/pin$ 上训练，并在全部 14 种模式上评估，其中包括对未见过的$ip/pi/2u/up$ 模式的泛化。也就是说，该训练协议假定，在原子逻辑算子上训练的模型能够通过组合泛化到使用多个算子的模式，例如同时使用交集和投影的ip 和pi 查询。

我们总结了不同的训练目标函数，见表 14。大多数在图上学习查询和实体表示的方法都会优化对比损失，即，最小化查询表示与其正答案表示之间的距离 q，同时最大化查询表示与负答案表示之间的距离 e。不同的目标函数包括：(1) 最大间隔损失（表 14的第一列），其目标是使负答案的距离至少比正答案的距离大一个间隔 γ。这类损失通常具有如下方程的形式。

$$\ell = \max(0, \gamma - \texttt{dist}(\mathbf{q}, \mathbf{e}) + \texttt{dist}(\mathbf{q}, \mathbf{e}'));$$

（2）对数 Sigmoid 损失（表 14中的第二列），其目标类似，即提高负样本的距离，反之亦然。损失通常还包含一个间隔项；满足间隔条件后，梯度会逐渐减小。

$$\ell = -\log \sigma(\gamma - \texttt{dist}(\mathbf{q}, \mathbf{e})) - \sum \frac{1}{k} \log \sigma(\texttt{dist}(\mathbf{q}, \mathbf{e}') - \gamma),$$

其中 k 表示负答案的数量。其他方法（表 14中的第三列）直接对图中所有节点的 logit 向量建模，因此可能会优化交叉熵损失，而非对比损失。此外，CQD 的两种变体（Arakelyan 等人，2021年）或 QTO（Bai 等人，2022b）等方法只优化链接预测损失，因为它们不学习查询的表示。

几乎所有数据集，包括 GQE（Hamilton 等，2018）、Q2B（Ren 等，2020）、BetaE（Ren 与 Leskovec，2020）、RegEx（Adlakha 等，2021）、BiQE（Kotnis 等，2021）、Query2Onto（Andresel 等，2021）、TAR（Tang 等，2022）、StarQE（Alivanistos 等，2022）、GNNQ（Pflueger 等，2022）、TeMP（Hu 等，2022）、TFLEX（Lin等，2022a）、InductiveQE（Galkin 等，2022b）、SQE（Bai 等，2023年），都提供

表 14：按损失分类的复杂查询回答方法。

最大间隔	对数 Sigmoid（Sun 等人，2018年）	交叉熵
GQE (Hamilton et al., 2018年)，带哈希的 GQE (Wang 等人，2019年)，CGA (Mai 等人，2019年)，MPQE (Daza & Cochez，2020年)，HyPE (Choudhary et al., 2021b)，Shv (Gebhart 等人，2021年)	Query2Box (Ren 等人，2020年)，BetaE (Ren 与 Leskovec，2020年)，RotatE-Box (Adlakha 等人,2021年)，ConE (Zhang 等人，2021b)，NewLook (Liu 等人，2021年)，Q2B Onto (Andresel 等人，2021年)，PERM (Choudhary 等人，2021a)，LogicE (Luus 等人，2021年)，MLPMix (Amayuelas 等人，2022年)，FuzzQE (Chen 等人，2022b)，FLEX (Lin 等人，2022b)，TFLEX (Lin 等人，2022a)，SMORE (Ren 等人，2022a)，LinE (Huang 等，2022b)，GammaE (Yang 等，2022a)，NMP-QEM (Long 等，2022年)，ENeSy (Xu 等，2022年)，RoMA (Xi 等人，2022年)，SignalE (Wang 等人，2022a)，Query2Geom (Sardina 等人，2023年)	BiQE (Kotnis et al., 2021年)，NodePiece-QE (Galkin et al., 2022b)，GNN-QE (Zhu et al., 2022年)，GNNQ (Pflueger 等 al., 2022年)，KGTrans (Liu 等人，2022年)，Query2Particles (Bai 等人，2022a)，EmQL (Sun 等人，2020年)、LMPNN (Wang 等人，2023c)，StarQE (阿里-vanistos 等人，2022年)，NQE (Luo 等人，2023年)，CQD$^{\mathcal{A}}$ (Arakelyan 等人，2023年)，SQE (Bai 等人，2023年)

表 15：现有逻辑查询回答数据集按照支持的**查询算子**、**推理**场景和**领域**属性进行分类。

		查询算子					**推理**		**领域**					
来源	数据集	Conjunctive	Union	Negation	Kleene Plus	Filter + Agg	Transductive	Inductive	Discrete	+ Timestamps	+ Continuous	Types	Rules	Qualifiers
Hamilton 等人 (2018年)	GQE 数据集	✓					✓		✓					
Ren 等人 (2020年)	Q2B 数据集	✓	✓				✓		✓					
Ren 与 Leskovec (2020年)	BetaE 数据集	✓	✓	✓			✓		✓					
Adlakha 等人(2021年)	正则表达式查询	✓	✓		✓		✓		✓					
Kotnis 等人(2021年)	有向无环图查询	✓					✓		✓					
Wang 等人 （2021年）	EFO-1 查询	✓	✓	✓			✓		✓					
Andresel 等人 （2021年）	LUBM/NELL（类型）	✓	✓				✓		✓				✓	
Tang 等人 （2022年）	TAR 数据集	✓	✓				✓		✓			✓		
Ren 等人 (2022年)	SMORE 数据集	✓	✓				✓		✓					
Alivanistos 等人(2022年)	WD50K 数据集	✓					✓		✓					✓
Hu 等人(2022年)	TeMP 数据集	✓	✓					✓	✓			✓		
Pflueger 等人 （2022年）	GNNQ 数据集	✓						✓	✓					
Lin 等人 （2022a）	TFLEX 数据集	✓	✓	✓			✓		✓	✓				
Galkin 等人 （2022b）	InductiveQE 数据集	✓	✓	✓				✓	✓					
Luo 等(2023年)	WD50K-NFOL 数据集	✓	✓	✓			✓		✓					✓
Bai 等人(2023年)	SQE 数据集	✓	✓	✓			✓		✓					

从$\mathcal{G}_{train}$中采样得到的一组具有给定结构的训练查询$\mathcal{G}_{train}$，其优势在于训练期间，方法无需在线采样查询。然而，这通常意味着$\mathcal{G}_{train}$ 中只有部分信息得到利用，因为G上存在数量呈指数级增长的多跳查询，而数据集不可能预先离线生成所有查询。SMORE（Ren 等人，2022）提出了一种双向在线查询采样器，使方法无需离线预生成训练集，即可高效地直接进行在线采样。另一种情况是，对于没有参数化方式来处理逻辑操作的方法，例如，CQD（Arakelyan 等人，2021），训练整个系统只需要单跳边。

7.4 推理

我们将 推理 理解为对经过训练的查询回答模型将要部署并评估的测试场景进行测试。遵循相关文献，我们区分 传导式 和归纳式 推理（图 25）。在

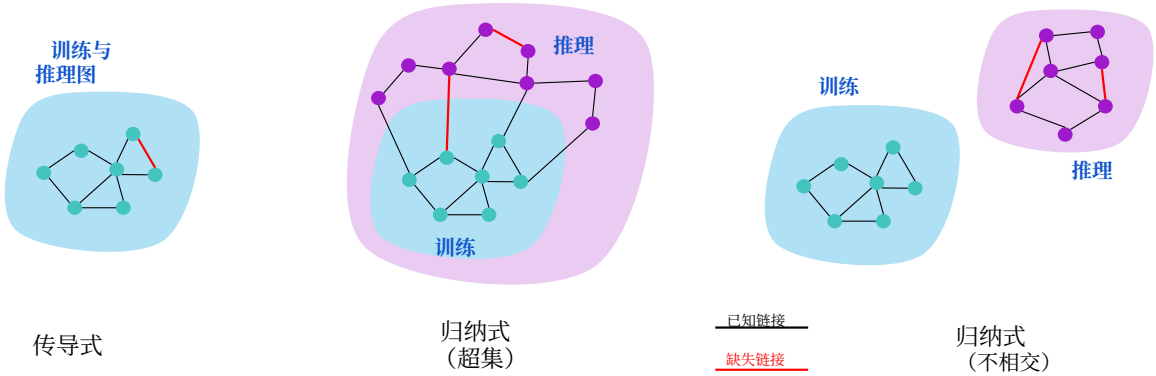


图25：推理场景。在传导式情况下，训练图与推理图相同，并共享相同的节点（$\mathcal{E}^{inf} = \mathcal{E}_{train}$）。归纳式情况可分为超集情况，此时推理图扩展了训练图（$\mathcal{E}_{train} \subseteq \mathcal{E}^{inf}$，缺失链接涵盖已见节点和未见节点）；以及不相交情况，此时推理图彼此断开（$\mathcal{E}_{train} \cap \mathcal{E}_{inf} = \emptyset$，缺失链接位于未见节点之间）。

传导式情形中，推理在具有与训练时相同节点集和关系类型、但边不同的图上进行。当推理图中的节点数量或关系类型与训练时不同，或出现其他此类情形时，则将其视为归纳式。推理场景在查询回答模型的设计中发挥着重要作用，即传导式模型得益于固定的实体集，可以学习浅层实体嵌入矩阵；而归纳式模型则必须依赖底层图中可用的其他 不变性，以便泛化到未见过的实体/关系类型。我们在 第5.2节中讨论了许多传导式和归纳式模型。下面，我们从 推理的角度对现有数据集进行分类。现有复杂逻辑查询回答数据集的概览呈现在 表 15 中，考察了其支持的查询算子、推理场景、图领域，以及类型或限定符等其他特征。

传导式推理。形式化地说，给定训练图$\mathcal{G}_{train} = (\mathcal{E}_{train}, \mathcal{R}_{train}, \mathcal{S}_{train})$，传导式推理图$\mathcal{G}_{inf}$[9] 包含相同的实体集和关系类型，即$\mathcal{E}_{train} = \mathcal{E}^{inf}$ 且$\mathcal{R}_{train} = \mathcal{R}^{inf}$；而$\mathcal{G}_{train}$ 上的边集是推理图$\mathcal{G}^{inf}$上边集的子集，即，$\mathcal{S}_{train} \subset \mathcal{S}^{inf}$。在此设置下，查询回答在训练期间已见的相同节点和边上进行。从实体集的角度来看，预测模式是已见节点到已见节点– 在已知实体之间预测缺失链接。

传统上，知识图谱链接预测更加侧重于传导式任务。因此，在复杂逻辑查询回答中，现有数据集中的大多数数据集（表 16）都遵循传导式场景。从 GQE 数据集（Hamilton 等，2018）、Query2Box 数据集（Ren 等，2020）和 BetaE 数据集（Ren 与 Leskovec，2020）中的简单基于三元组的图和固定查询模式开始，这些数据集后来成为查询回答方法事实上的标准基准测试；较新的数据集则纳入了正则表达式查询（Adlakha 等，2021）、更广泛的查询模式集合（Kotnis 等，2021；Wang 等，2021；Bai 等，2023年）、实体类型信息（Tang等，2022）、本体公理（Andresel 等，2021）、带有限定符的超关系查询（Alivanistos 等，2022；Luo 等，2023）、时序查询（Lin等，2022a），或规模达到 1 亿个节点的大型图（Ren 等，2022）。

归纳式推理。形式上，给定训练图 $G_{train} = (E_{train}, R_{train}, S_{train})$，归纳式推理图 $G^{inf} = (E^{inf}, R^{inf}, S^{inf})$ 与训练图在实体集、关系集或两者上均有所不同。这种差异的性质说明了归纳式推理的几种子类型。首先，推理时的关系集可能共享，也可能不共享，即 $R_{inf} \subseteq R_{train}$ 或 $|R^{inf} \setminus R_{train}| > 0$。大多数关于知识图谱归纳式链接预测的研究（Teru等，2020；Zhu 等，2021；Galkin等，2022a）都假设关系集是共享的，而推理时出现新关系的设定仍极具挑战性（Huang等，2022a；Gao 等，2023；Chen等，2023）。

另一方面，推理图可能是在添加新节点和边之后成为训练图的超集，$\mathcal{E}_{train} \subseteq \mathcal{E}^{inf}$，也可能是一个包含全新实体、作为非连通分量的不相交图，

[9]下文使用 $\mathcal{G}^{inf}$ 指代推理过程中使用的图，可以是 $\mathcal{G}_{val}$ 或 $\mathcal{G}_{test}$，且不损失泛化能力。

$\mathcal{E}_{inf} \cap \mathcal{E}_{train} = \emptyset$ 如图 25所示。从节点集的角度来看，超集归纳推理情况可能同时包含未见实体到已见实体的和未见实体到未见实体的缺失链接，而在不相交推理图中，只有未见实体到未见实体的链接会自然出现。

在 CLQA 中，归纳推理仍是一个新兴领域，因为它会直接影响可能的变量空间 $\mathcal{V}$、常量集合 $\mathcal{C}$以及答案 A，这些现在可能包含训练时未见过的实体。最近有几项工作开始探索归纳式查询回答（表 16）。InductiveQE 数据集（Galkin 等人，2022b）关注归纳超集情形：训练图最多可扩展至包含 500% 的新未见节点。测试查询从未见过的常量开始，因此求答需要对已见和未见节点进行推理。同样，在扩展后的推理图上求答时，训练查询也可能有许多新的正确答案。GNNQ 数据集（Pflueger 等人，2022年）关注不相交归纳推理情形，其中常量、变量和答案都属于一个新的实体集。TeMP 数据集（Hu 等人，2022年）同样关注不相交归纳推理，但还提供了一个额外的类层次结构，可用作可学习的 不变量。也就是说，训练时和推理时的类集合保持不变。

如第 3 节所述，归纳式推理对于神经图数据库至关重要，它使模型能够在无需重新训练的情况下运行于可更新图上。我们推测，归纳式数据集和模型可能会成为未来工作的主要贡献领域。

表 16：按推理分类的复杂查询回答数据集。

传导式	归纳式
GQE 数据集 (Ren 等人，2020年)	InductiveQE 数据集 (Galkin 等人，2022b)
Query2Box 数据集 (Ren 等人，2020年)	TeMP 数据集（类型感知） (Hu 等人，2022年)
BetaE 数据集 (Ren 与 Leskovec，2020年)	GNNQ 数据集（Pflueger 等人，2022年）
RegEx 数据集 (Adlakha 等人，2021年)	
BiQE 数据集 (Kotnis 等人，2021年)	
Query2Onto 数据集 (Andresel 等人，2021年)	
EFO-1 数据集 (Wang 等人，2021年)	
TAR 数据集 (Tang 等人，2022年)	
SMORE 数据集 (Ren 等人，2022年)	
StarQE 数据集 (Alivanistos 等人，2022年)	
TFLEX 数据集 (Lin 等人，2022a)	
WD50K-NFOL 数据集 (Luo 等，2023年)	
SQE 数据集 (Bai 等人，2023年)	

7.5 指标

人们提出了多种指标，用于评估查询推理模型的性能。这些指标大致可分为**泛化**、**蕴含**和**查询表示质量**指标。

泛化指标。 由于查询推理模型旨在对大规模不完整图进行推理，因此大多数指标都用于评估模型发现缺失答案的泛化能力，即，$[\![q]\!]_{\texttt{test}} \backslash [\![q]\!]_{\texttt{val}}$ 对于给定的测试查询 q。作为该领域最早的工作之一，GQE（Hamil- ton 等人，2018年）提出了 ROC-AUC 和平均百分位排名（APR）。其思想是，对于给定的测试查询 q，GQE 会为其所有缺失答案 $e \in [\![q]\!]_{\texttt{test}} \backslash [\![q]\!]_{\texttt{val}}$ 以及负例 $e' \notin [\![q]\!]_{\texttt{test}}$计算分数。模型的性能由 ROC-AUC 分数和 APR 衡量，其中，模型会将一个缺失答案与至多 1000 个随机采样的、具有相同实体类型的负例进行排名比较。除 GQE 外，GQE+哈希（Wang 等人，2019年）、CGA（Mai 等人，2019年）和 TractOR（Friedman &Broeck，2020年）也使用相同的评估指标。

然而，上述指标无法反映现实场景，因为在现实中，负例的数量往往比缺失答案多出几个数量级。Query2Box（Ren 等人，2020年）没有采用 ROC-AUC 或 APR，而是提出了基于排名的指标，例如平均倒数排名（MRR）和 hits@k。给定一个测试查询 q，对于每个缺失答案 $e \in [\![q]\!]_{\texttt{test}} \backslash [\![q]\!]_{\texttt{val}}$，我们将其与所有其他负例 $e' \notin [\![q]\!]_{\texttt{test}}$进行排序。给定排名结果 r，MRR 的计算方法为 $\frac{1}{r}$，hits@k 的计算方法为 $1[r \leq k]$。这些指标一直是该任务中最常用的指标。

请注意，最终排名仅针对困难答案计算，这些答案需要预测至少一个缺失链接。对于可通过边遍历到达的简单答案，通常会舍弃其排名。

表示质量指标。 除了评估模型寻找缺失答案的能力外，另一个方面是评估所有模型所学习的查询表示的质量。BetaE（Ren 与 Leskovec，2020年）提出评估所学习的查询表示能否建模查询答案集的基数，并将此视为查询表示质量的替代指标。对于具有“体积”概念的模型（例如，Beta嵌入的微分熵），目标是衡量查询的“体积”（根据查询表示计算）与答案集基数之间的斯皮尔曼等级相关系数和皮尔逊相关系数。BetaE 还提出使用 ROC-AUC 评估模型对无答案查询的建模能力。

蕴含指标。 另一种评估协议关注模型能否发现已有答案，例如，$[\![q]\!]_{\texttt{val}}$ 对于测试查询而言，这不需要推断缺失链接，而是侧重于记忆图结构（简单答案，按常见术语而言）。在 EmQL（Sun 等人，2020年）中，这被称为忠实性（或蕴含）。对于数据库查询任务而言，查询回答模型首先应找回图中已有的简单答案（通过边遍历即可到达），然后再利用链接预测推断出困难答案，以扩充答案集。因此，一个自然的指标是模型将简单答案排在困难答案之前的能力——InductiveQE（Galkin 等人，2022b）对此进行了研究，并提出使用 ROC-AUC 作为该任务的主要指标。

不过，我们认为，现有指标可能无法完全捕捉神经查询回答的本质，因此可能需要新的指标。例如，一些尚未得到充分探索但可能有用的指标包括：(1) 研究合理答案（介于简单答案和困难答案之间），这些答案可以由符号推理器利用更高层次的图模式（本体）推导出来（Andresel 等人，2021年）。计算合理答案时需要注意的是，符号推理器的处理时间可能是无限的，因此必须通过时间或表达能力加以限制，才能在有限时间内完成推理。因此，合理答案集可能仍然不完整；(2) 根据开放世界假设（OWA）进行评估。该假设认为，图中的未知三元组不一定是错误的（而标准的封闭世界假设在链接预测中被广泛采用，后者则认为未知内容为假）。实际上，OWA意味着，即使测试集也可能是不完整的，并且测试集判定为错误的一些高排名预测，实际上可能在（可能无法观测到的）完整图中是正确的。Yang 等（2022b）对链接预测进行OWA评估的初步实验表明，常见数据集上的排名指标（例如，MRR）趋于饱和，是因为神经链接预测器能够预测测试集中遗漏的、来自完整图的答案。例如，测试集上饱和的MRR为0.4，可能对应于真实完整图上的MRR为0.9。在传导式和归纳式设置下研究查询回答任务中的OWA评估，是未来研究的一个重要方向。

8 应用

除图数据库的应用场景外，复杂查询回答框架还应用于各种图条件机器学习任务。例如，SE-KGE（Mai 等人，2020年）将 GQE 应用于根据数值 $\{x,\ y\}$ 坐标进行条件化的地理空间查询。坐标编码器将数值表示与实体嵌入融合，因此预测任务仍是实体排序。

在基于案例的推理中，CBR-SUBG（Das 等人，2022年）是一种基于子图提取和编码、用于知识图谱问答的方法。作为副产品，CBR-SUBG 能够回答带有投影和交集的合取查询。然而，由于采用了非标准评估协议和定制合成数据集，其性能无法与复杂查询回答模型直接比较。类似地，Wang 等人（2023b）将类似 Query2Box 的模型与预训练语言模型结合，以提升问答性能。LEGO（Ren 等人，2021年）也将复杂查询回答模型应用于知识图谱问答。其思路是同时将自然语言问题解析为查询步骤，并使用复杂查询回答模型在潜在空间中执行该步骤。

LogiRec（Wu 等人，2022年）将产品推荐建模为复杂逻辑查询，其中源产品是根节点，多个产品的组合形成交集，而与源产品不相似的产品则

会通过否定进行过滤。LogiRec 采用 BetaE 作为查询引擎。同样，Syed 等人（2022年）基于 Query2Box 设计了一种可解释推荐系统。给定一个逻辑查询，他们首先使用 Query2Box 生成候选集，然后利用神经协同过滤对候选项进行重新排序（He 等人，2017年）。

9 总结与未来机遇

我们总结了复杂逻辑查询回答（CLQA）任务的相关文献。该任务是我们提出的神经查询引擎和神经图数据库（NGDB）概念的核心。神经图数据库将神经数据库与图数据库的理念相结合，为信息（包括实体、关系和事实）提供独特的潜在空间。神经图数据库利用图表示学习，在潜在空间中完成查询规划、执行和结果检索。这种设计选择为存储和查询提供了统一、更高效且更稳健的接口。我们设想，神经图数据库将成为新一代数据库，专为海量不完整图结构数据上的预测性查询而设计。

我们还提出了一套深入而详尽的方法分类体系，用于应对预想中的神经查询引擎的特定方面。展望未来，要充分释放神经图数据库的潜力，仍有很大空间；为此，需要解决神经查询引擎和神经存储领域中的开放挑战。针对神经查询引擎，并依据第 3.2 节中的分类体系，我们将挑战归纳为三个主要方面：图、建模和查询。

沿着**图**这一分支：

- **模态：**支持更多图模态：从经典的仅含三元组的图，到超关系图、超图，以及结合图、文本、图像等内容的多模态数据源。
- **推理域：**支持对时间和连续数据进行逻辑推理和神经查询回答（例如文本和数值数据）——字面量构成了图的重要部分，也涉及针对字面量的相关查询。
- **背景语义：**支持复杂公理和形式语义，以编码（潜在）实体类别及其层级结构之间的高阶关系，例如，实现基于描述逻辑和 OWL 片段的神经推理。

在**建模**这一分支：

- **编码器：**归纳式编码器支持在推理时处理未见过的关系——这对于（1）神经数据库的**可更新性**至关重要，无需重新训练；（2）采用**预训练-微调**策略，将查询回答泛化到具有自定义关系模式的自定义图。
- **处理器：** 能够有效且高效地执行类似于 SPARQL 和 Cypher 算子的复杂查询算子的表达能力强大的处理器网络。提高神经处理器的样本效率，对于权衡训练时间与质量至关重要，即在缩短训练时间的同时保持较高的预测质量。
- **解码器：** 迄今为止，所有神经查询回答解码器都仅在离散节点上运行。将答案值域扩展至连续输出，对于回答现实世界中的查询至关重要。

  复杂度： 处理器网络的主要计算瓶颈在于嵌入空间的维度（对于纯神经模型）和/或节点数量（对于神经符号模型）。因此，神经逻辑算子和检索方法的新型高效算法，是将神经图数据库扩展至数十亿节点和数万亿条边的关键。

关于**查询**：

- **算子：**将更复杂的查询算子神经化，使其表达能力与声明式图查询语言相匹配，例如支持克莱尼正闭包与克莱尼星闭包、属性路径和过滤器。

- **模式：**求答超越树状查询的更复杂模式，包括有向无环图和循环图。

- **投影变量：**允许投影多个变量，而不仅仅是最终叶节点实体；也就是说，允许返回中间变量、关系，以及以元组（绑定）形式组织的多个变量。
- **表达能力：**回答简单 EPFO 和 EFO 片段之外的查询，将表达能力扩展至带否定的合取查询并集（UCQ 和 $\text{UCQ}_{\text{否定}}$），并致力于达到数据库语言的表达能力。

在**数据集**与**评估**：

- 需要更大且更多样化的**基准数据集**，以涵盖更多图模态、更丰富的查询语义、更多查询算子和查询模式。

- 现有评估协议似乎有所局限（仅关注推断困难答案），因此需要更为规范的**评估框架**和**指标**，以涵盖查询回答工作流的各个方面。

关于神经图存储和神经图数据库总体而言，我们确定了以下挑战：

- 需要一种**可扩展的检索**机制，使神经推理能够扩展到拥有数十亿节点的图。检索与查询处理器及其建模先验紧密相关（如第 3.1 节所示）。现有的可扩展近似最近邻库只能处理基本的 L1、L2 和余弦距离，这限制了神经查询引擎中可用处理器的种类。

- 目前，第 7 节中列出的所有复杂查询数据集都提供了硬编码的查询执行计划，而这些计划可能并非神经处理器的最优方案。因此，需要一种**神经查询规划器**，将输入查询转换为最优执行序列，同时考虑预测任务、查询复杂度、神经处理器的类型以及存储层的配置（存储层也可以是联邦式的）。

- 由于编码器具有归纳性和无需重新训练即可更新的特性，因此有必要缓解**持续学习**(Thrun，1995；Ring，1998年）、**灾难性遗忘**(McCloskey &Cohen，1989年）以及**规模泛化**方面的问题，即在远大于训练图的图上进行推理时所面临的问题（Yehudai 等人，2021年；Buffelli 等人，2022年）。

最后，在基础模型时代，许多研究工作展示了这些通常拥有数千亿参数的大语言模型中隐藏的各种能力，更重要的是，展示了如何释放这些能力（Wei 等人，2022年；Wang 等人，2022b；Zhou 等人，2022a）。我们设想，一个重要的未来方向是设计自然语言接口，从而能够沿着我们前文提到的方向，更好地发挥这些大语言模型在 CLQA任务中的推理能力（Drozdov 等，2022年）。此外，神经图数据库还为在各个阶段改进基础模型提供了令人振奋的未来机遇，尤其是在推理阶段。由于并非所有下游任务都需要调用完整的十亿参数模型，因此，研究如何利用神经图数据库加速或压缩基础模型，同时保留这些超大模型所必需的涌现行为，具有很大的前景。

致谢

感谢 Pavel Klinov（Stardog）、Matthias Fey（Kumo.AI）、李倩（斯坦福）和 Keshav Santhanam（斯坦福）对本文草稿提出宝贵意见。

参考文献

Vaibhav Adlakha、Parth Shah、Srikanta J. Bedathur 和 Mausam。关于不完整知识库的正则表达式查询。载于第*3*届自动知识库构建会议（*AKBC*），2021年。

Mehdi Ali、Max Berrendorf、Charles Tapley Hoyt、Laurent Vermue、Mikhail Galkin、Sahand Sharifzadeh、Asja Fischer、Volker Tresp 和 Jens Lehmann。拨开黑暗：统一框架下对知识图谱嵌入模型的大规模评估。*IEEE*模式分析与机器智能汇刊，2021年。doi：10.1109/TPAMI.2021.3124805。

Dimitrios Alivanistos、Max Berrendorf、Michael Cochez 和 Mikhail Galkin。超关系知识图谱上的查询嵌入。发表于国际学习表征会议，2022年。

Uri Alon、Frank Xu、Junxian He、Sudipta Sengupta、Dan Roth 和 Graham Neubig。利用自动机增强检索进行神经符号语言建模。发表于国际机器学习会议，第468–485页。PMLR，2022年。

Güneş Aluç、Olaf Hartig、M Tamer Özsu 和 Khuzaima Daudjee。RDF数据管理系统的多样化压力测试。载于 第*13*届国际语义网会议，第197–212页。施普普林格，2014年。

Alfonso Amayuelas、Shuai Zhang、Xi Susie Rao 和 Ce Zhang。知识图谱上的逻辑推理神经方法。载于 国际学习表征会议，2022年。

Medina Andresel、Csaba Domokos、Daria Stepanova 和Trung-Kien Tran。一种用于本体介导的查询回答的神经符号方法。*arXiv*预印本 *arXiv:2106.14052*，2021年。

Erik Arakelyan、Daniel Daza、Pasquale Minervini 和 Michael Cochez。使用神经链接预测器进行复杂查询回答。载于 国际学习表征会议，2021年。

Erik Arakelyan、Pasquale Minervini 和 Isabelle Augenstein。为复杂查询回答调整神经链接预测器。*arXiv*预印本 *arXiv:2301.12313*，2023年。

Alessandro Artale、Diego Calvanese、Roman Kontchakov 和 Michael Zakharyaschev。DL-Lite族与关系。《人工智能》研究期刊，36:1–69，2009年。

Akari Asai、Kazuma Hashimoto、Hannaneh Hajishirzi、Richard Socher 和 Caiming Xiong。学习通过维基百科图谱检索用于问答的推理路径。发表于国际学习表征会议，2020年。URL https://openreview.net/forum?id=SJgVHkrYDH。

Franz Baader、Diego Calvanese、Deborah McGuinness、Peter Patel-Schneider、Daniele Nardi 等。《描述逻辑手册：理论、实现与应用》。剑桥大学出版社，2003年。

Samy Badreddine、Artur d'Avila Garcez、Luciano Serafini 和 Michael Spranger。逻辑张量网络。《人工智能》，303:103649，2022年。

Jiaxin Bai、Zihao Wang、Hongming Zhang 和 Yangqiu Song。Query2Particles：利用粒子嵌入进行知识图谱推理。载于计算语言学协会：*NAACL 2022* 年会论文集。计算语言学协会，2022a。

Jiaxin Bai、Tianshi Zheng 和 Yangqiu Song。面向知识图谱复杂查询回答的序列查询编码。*arXiv* 预印本 *arXiv:2302.13114*，2023年。

Yushi Bai、Xin Lv、Juanzi Li 和 Lei Hou。通过查询计算树优化回答知识图谱上的复杂逻辑查询。*arXiv* 预印本 *arXiv:2212.09567*，2022b。

Pablo Barceló、Egor V. Kostylev、Mikael Monet、Jorge Pérez、Juan Reutter 和 Juan Pablo Silva。图神经网络的逻辑表达能力。载于国际学习表征会议，2020年。URL https://openreview.net/forum?id=r1lZ7AEKvB。

Pablo Barceló、Mikhail Galkin、Christopher Morris 和 Miguel Romero Orth。Weisfeiler和Leman走向关系化。载于第一届图学习会议，2022年。URL https://openreview.net/forum?id=wY_IYhh6pqj。

Peter W Battaglia、Jessica B Hamrick、Victor Bapst、Alvaro Sanchez-Gonzalez、Vinicius Zambaldi、Mateusz Malinowski、Andrea Tacchetti、David Raposo、Adam Santoro、Ryan Faulkner 等。关系归纳偏置、深度学习与图网络。*arXiv* 预印本 *arXiv:1806.01261*，2018年。

Maciej Besta、Patrick Iff、Florian Scheidl、Kazuki Osawa、Nikoli Dryden、Michal Podstawski、Tiancheng Chen 和 Torsten Hoefler。神经图数据库。载于图学习会议，2022年。URL https://openreview.net/forum?id=pOsMj8oH2O。

Luca Beurer-Kellner、Marc Fischer 和 Martin Vechev。提示即编程：面向大语言模型的查询语言。*arXiv* 预印本 *arXiv:2212.06094*，2022年。

Kurt Bollacker、Colin Evans、Praveen Paritosh、Tim Sturge 和 Jamie Taylor。《Freebase：一个协作创建的图数据库，用于构建人类知识》。载于*2008*年 *ACM SIGMOD* 数据管理国际会议论文集，第 1247–1250 页，2008年。

Rishi Bommasani、Drew A. Hudson、Ehsan Adeli、Russ Altman、Simran Arora、Sydney von Arx、Michael S. Bernstein、Jeannette Bohg、Antoine Bosselut、Emma Brunskill、Erik Brynjolfsson、S. Buch、Dallas Card、Rodrigo Castellon、Niladri S. Chatterji、Annie S. Chen、Kathleen A. Creel、Jared Davis、Dora Demszky、Chris Donahue、Moussa Doumbouya、Esin Durmus、Stefano Ermon、John Etchemendy、Kawin Ethayarajh、Li Fei-Fei、Chelsea Finn、Trevor Gale、Lauren E. Gillespie、Karan Goel、Noah D. Goodman、Shelby Gross- man、Neel Guha、Tatsunori Hashimoto、Peter Henderson、John Hewitt、Daniel E. Ho、Jenny Hong、Kyle Hsu、Jing Huang、Thomas F. Icard、Saahil Jain、Dan Jurafsky、Pratyusha Kalluri、Siddharth Karamcheti、Geoff Keeling、Fereshte Khani、O. Khattab、Pang Wei Koh、Mark S. Krass、Ranjay Krishna、Rohith Kudi- tipudi、Ananya Kumar、Faisal Ladhak、Mina Lee、Tony Lee、Jure Leskovec、Isabelle Levent、Xiang Lisa Li、Xuechen Li、Tengyu Ma、Ali Malik、Christopher D. Manning、Suvir P. Mirchandani、Eric Mitchell、Zanele Munyikwa、Suraj Nair、Avanika Narayan、Deepak Narayanan、Benjamin Newman、Allen Nie、Juan Carlos Niebles、Hamed Nilforoshan、J. F. Nyarko、Giray Ogut、Laurel Orr、Isabel Papadimitriou、Joon Sung Park、Chris Piech、Eva Portelance、Christopher Potts、Aditi Raghunathan、Robert Reich、Hongyu Ren、Frieda Rong、Yusuf H. Roohani、Camilo Ruiz、Jack Ryan、Christopher R'e、Dorsa Sadigh、Shiori Sagawa、Keshav Santhanam、Andy Shih、Krishna Parasuram Srinivasan、Alex Tamkin、Rohan Taori、Armin W. Thomas、Florian Tramèr、Rose E. Wang、William Wang、Bohan Wu、Jiajun Wu、Yuhuai Wu、Sang Michael Xie、Michihiro Yasunaga、Jiaxuan You、Matei A. Zaharia、Michael Zhang、Tianyi Zhang、Xikun Zhang、Yuhui Zhang、Lucia Zheng、Kaitlyn Zhou 和 Percy Liang。《基础模型的机遇与风险》。*arXiv*，2021年。URL https://crfm.stanford.edu/assets/report.pdf。

Antoine Bordes、Nicolas Usunier、Alberto Garcia-Duran、JasonWeston 和 OksanaYakhnenko。《用于多关系数据建模的平移嵌入》。神经信息处理系统进展，第 26 卷，2013年。

Elizabeth Boschee、Jennifer Lautenschlager、Sean O'Brien、Steve Shellman、James Starz 和 Michael Ward。《ICEWS编码事件数据》。*Harvard Dataverse*，2015年。URL https://doi.org/10.7910/DVN/28075。

Dan Brickley、Ramanathan V Guha 和 Brian McBride。《RDF模式1.1》。*W3C* 建议，25：2004–2014，2014年。

Nathan Bronson、Zach Amsden、George Cabrera、Prasad Chakka、Peter Dimov、Hui Ding、Jack Ferris、Anthony Giardullo、Sachin Kulkarni、Harry Li 等。{TAO}：Facebook 的社交图分布式数据存储。载于*2013 {USENIX}* 年度技术会议（*{USENIX}{ATC} 13)*，第 49–60 页，2013年。

Michael M Bronstein、Joan Bruna、Taco Cohen 和 Petar Veličković。《几何深度学习：网格、群、图、测地线与规范》。*arXiv* 预印本 *arXiv:2104.13478*，2021年。

Davide Buffelli、Pietro Liò 和 Fabio Vandin。SizeShiftReg：一种用于提升图神经网络规模泛化能力的正则化方法。载于神经信息处理系统进展，2022年。

Alison Callahan、José Cruz-Toledo、Peter Ansell 和 Michel Dumontier。Bio2RDF 第二版：提升生命科学关联数据的覆盖范围、互操作性和溯源能力。载于 Philipp Cimiano、Oscar Corcho、Valentina Presutti、Laura Hollink 和 Sebastian Rudolph（编），语义网：语义与大数据，第 200–212 页。Springer Berlin Heidelberg，2013年。Ines Chami、Sami Abu-El-Haija、Bryan Perozzi、Christopher Ré 和 Kevin Murphy。图上的机器学习：一种模型及全面的分类体系。机器学习研究杂志，23(89)：1–64，2022年。URL http://jmlr.org/papers/v23/20-852.html。Payal Chandak、Kexin Huang 和 Marinka Zitnik。构建知识图谱以实现精准医学。*Scientific Data*，10(1)：67，2023年。URL https://doi.org/10.1038/s41597-023-01960-3。Mingyang Chen、Wen Zhang、Yuxia Geng、Zezhong Xu、Jeff Z Pan 和 Huajun Chen。泛化到未见元素：知识图谱知识外推综述。*arXiv*预印本 *arXiv:2302.01859*，2023年。Wenhu Chen、Xueguang Ma、Xinyi Wang 和 William W Cohen。思维程序提示：将计算与推理分离，以完成数值推理任务。*arXiv*预印本 *arXiv:2211.12588*，2022a。Xinyun Chen、Chen Liang、Adams Wei Yu、Denny Zhou、Dawn Song 和 Quoc V. Le。神经符号阅读器：将分布式表示与符号表示可扩展地结合，用于阅读理解。载于国际学习表征会议，2020年。URL https://openreview.net/forum?id=ryxjnREFwH。Xuelu Chen、Ziniu Hu 和 Yizhou Sun。基于模糊逻辑的知识图谱逻辑查询问答。载于第三十六届*AAAI*人工智能会议，*AAAI*，第 3939–3948 页。AAAI出版社，2022b。Nurendra Choudhary、Nikhil Rao、Sumeet Katariya、Karthik Subbian 和 Chandan Reddy。用于知识图谱推理的概率实体表示模型。载于神经信息处理系统进展，2021a。Nurendra Choudhary、Nikhil Rao、Sumeet Katariya、Karthik Subbian 和 Chandan K Reddy。基于知识图谱逻辑查询的自监督双曲面表示。载于*2021*年万维网大会论文集，第 1373–1384 页，2021b。Michael Cochez。语义智能体编程语言：应用与形式化。硕士论文，于韦斯屈莱大学，2012年。Edgar F Codd。大型共享数据库的数据关系模型。*ACM*通讯，13 (6)：377–387，1970年。Graham Cormode 和 Shan Muthukrishnan。一种改进的数据流摘要：计数最小草图及其应用。算法杂志，55(1)：58–75，2005年。Rajarshi Das、Arvind Neelakantan、David Belanger 和 Andrew McCallum。使用循环神经网络对实体、关系和文本进行链式推理。载于计算语言学协会欧洲分会第*15*届会议论文集：第一卷，长篇论文，第 132–141 页，西班牙瓦伦西亚，2017年。计算语言学协会。URL https://aclanthology.org/E17-1013。Rajarshi Das、Ameya Godbole、Ankita Naik、Elliot Tower、Manzil Zaheer、Hannaneh Hajishirzi、Robin Jia 和 Andrew McCallum。通过对子图进行基于案例的推理来回答知识库问题。载于国际机器学习会议，*ICML 2022*，机器学习研究论文集。PMLR，2022年。

Daniel Daza 和 Michael Cochez。消息传递查询嵌入。《*ICML 2020*图表示学习及其拓展研讨会论文集》，2020年。

Lauren Nicole Delong、Ramon Fernández Mir、Matthew Whyte、Zonglin Ji 和 Jacques D. Fleuriot。用于图结构推理的神经符号人工智能：综述。*arXiv*预印本 *arXiv:2302.07200*，2023年。Xin Luna Dong。构建产品知识图谱的挑战与创新。载于第*24*届*ACM SIGKDD*国际知识发现与数据挖掘会议论文集，第2869–2869页，2018年。Andrew Drozdov、Nathanael Schärli、Ekin Akyürek、Nathan Scales、Xinying Song、Xinyun Chen、Olivier Bousquet 和 Denny Zhou。大语言模型的组合语义解析。*arXiv*预印本 *arXiv:2209.15003*，2022年。Cynthia Dwork、Frank McSherry、Kobbi Nissim 和 Adam Smith。在隐私数据分析中校准噪声与敏感度。载于理论密码学会议，第265–284页。施普林格，2006年。

Kemele M Endris、Mikhail Galkin、Ioanna Lytra、Mohamed Nadjib Mami、Maria-Esther Vidal 和 Sören Auer。通过连接RDF分子模板查询互联数据。载于大规模数据与知识中心系统汇刊 *XXXIX*，第 1–42 页。施普林格，2018年。

Javier D Fernández、Miguel A Martínez-Prieto、Claudio Gutiérrez、Axel Polleres 和 Mario Arias。用于发布和交换的二进制RDF表示（HDT）。*Journal of Web Semantics*，19：22–41，2013年。Emma Flint。宣布推出 Alexa实体（测试版）：借助便捷的 Alexa 知识访问功能，打造更智能、更具吸引力的技能，2021年。URL https://developer.amazon.com/en-US/blogs/alexa/alexa-skills-kit/2021/02/alexa-entities-beta。Tal Friedman 和 Guy Broeck。向量空间的符号查询：概率数据库与关系嵌入相遇。载于人工智能不确定性会议，第 1268–1277 页。PMLR，2020年。

Luis Galárraga、Christina Teflioudi、Katja Hose 和 Fabian M. Suchanek。Amie：本体知识库中基于不完整证据的关联规则挖掘。载于国际万维网会议（*WWW*）论文集，2013年。

Mikhail Galkin、Priyansh Trivedi、Gaurav Maheshwari、Ricardo Usbeck 和 Jens Lehmann。超关系知识图谱的消息传递。载于*2020*年自然语言处理实证方法会议（*EMNLP 2020*）论文集，第7346–7359页。计算语言学协会，2020年。

Mikhail Galkin、Etienne Denis、Jiapeng Wu 和 William L. Hamilton。Nodepiece：大型知识图谱的组合式参数高效表示。载于国际学习表征会议，2022a。URL https://openreview.net/forum?id=xMJWUKJnFSw。

Mikhail Galkin、Zhaocheng Zhu、Hongyu Ren 和 Jian Tang。知识图谱中的归纳逻辑查询回答。载于《神经信息处理系统进展》，2022b。Jianfei Gao、Yangze Zhou 和 Bruno Ribeiro。知识图谱补全的双重置换等变性。《*arXiv* 预印本 *arXiv:2302.01313*》，2023年。Thomas Gebhart、Jakob Hansen 和 Paul Schrater。知识层：知识图谱嵌入的层论框架。《*arXiv* 预印本 *arXiv:2110.03789*》，2021年。Justin Gilmer、Samuel S. Schoenholz、Patrick F. Riley、Oriol Vinyals 和 George E. Dahl。用于量子化学的神经消息传递。载于《*ICML*》，第 1263–1272 页，2017年。Aditya Grover 和 Jure Leskovec。node2vec：面向网络的可扩展特征学习。载于《第 *22* 届 *ACM SIGKDD* 知识发现与数据挖掘国际会议论文集》，第 855–864 页，2016年。Ruiqi Guo、Philip Sun、Erik Lindgren、Quan Geng、David Simcha、Felix Chern 和 Sanjiv Kumar。利用各向异性向量量化加速大规模推理。载于《国际机器学习会议》，2020年。URL https://arxiv.org/abs/1908.10396。

Yuanbo Guo、Zhengxiang Pan 和 Jeff Heflin。LUBM：OWL 知识库系统的基准。*Journalof Web Semantics*，3(2-3):158–182，2005年。Nitish Gupta、Kevin Lin、Dan Roth、Sameer Singh 和 Matt Gardner。用于文本推理的神经模块网络。载于国际学习表征会议，2020年。URL https://openreview.net/forum?id=SygWvAVFPr。Kelvin Guu、John Miller 和 Percy Liang。在向量空间中遍历知识图谱。载于*2015*年自然语言处理实证方法会议论文集，第 318–327 页。计算语言学协会，2015年。doi：10.18653/v1/D15-1038。URL https://aclanthology.org/D15-1038。Kelvin Guu、Kenton Lee、Zora Tung、Panupong Pasupat 和 Mingwei Chang。检索增强语言模型预训练。载于国际机器学习会议，第 3929–3938 页。PMLR，2020年。Ferras Hamad、Issac Liu 和 Xian Xing Zhang。Uber Eats 的食品发现：构建查询理解引擎，2018年。URL
https://www.uber.com/blog/uber-eats-query-understanding/。

Will Hamilton、Payal Bajaj、Marinka Zitnik、Dan Jurafsky 和 Jure Leskovec。知识图谱上的逻辑查询嵌入。载于《神经信息处理系统进展》，第 31 卷，2018年。

William L Hamilton。《图表示学习》。Morgan & Claypool Publishers，2020年。

Jakob Hansen 和 Robert Ghrist。迈向胞腔层的谱理论。《应用与计算拓扑学期刊》，3：315–358，2019年。

Olaf Hartig、Pierre-Antoine Champin、Gregg Kellogg 和 Andy Seaborne。RDF-star 和 SPARQL-star。《社区组报告草案》，2022年。

Sandro Hawke、Ivan Herman、Bijan Parsia、Axel Polleres、，和 Andy Seaborne。SPARQL 1.1 蕴涵机制。《*W3C* 建议》，2013年。

Xiangnan He、Lizi Liao、Hanwang Zhang、Liqiang Nie、Xia Hu 和 Tat-Seng Chua。神经协同过滤。载于第*26*届万维网国际会议论文集，第173–182页，2017年。Dan Hendrycks、Collin Burns、Saurav Kadavath、Akul Arora、Steven Basart、Eric Tang、Dawn Song 和 Jacob Steinhardt。使用 MATH 数据集衡量数学问题求解能力。*NeurIPS*，2021年。Vinh Thinh Ho、Daria Stepanova、Dragan Milchevski、Jannik Strötgen 和 Gerhard Weikum。利用数量事实增强知识库。载于*WWW '22*：*2022* 年 *ACM Web Conference*，第893–901页。ACM，2022年。Aidan Hogan、Eva Blomqvist、Michael Cochez、Claudia d'Amato、Gerard de Melo、Claudio Gutierrez、Sabrina Kirrane、José Emilio Labra Gayo、Roberto Navigli、Sebastian Neumaier 等。知识图谱。*ACMComputing Surveys (CSUR)*，54(4):1–37，2021年。Zhiwei Hu、Víctor Gutiérrez-Basulto、Zhiliang Xiang、Xiaoli Li、Ru Li 和 Jeff Z Pan。用于知识图谱多跳推理的类型感知嵌入。*arXiv*预印本 *arXiv:2205.00782*，2022年。Qian Huang、Hongyu Ren 和 Jure Leskovec。通过连接子图预训练进行少样本关系推理。载于 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho（编），神经信息处理系统进展，2022a年。URL https://openreview.net/forum?id=LvW71lgly25。Xingyue Huang、Miguel Romero Orth、İsmail İlkan Ceylan 和 Pablo Barceló。通过关系 Weisfeiler-Leman 进行链接预测的理论。*arXiv*预印本 *arXiv:2302.02209*，2023年。Zijian Huang、Meng-Fen Chiang 和 Wang-Chien Lee。LinE：层次知识图谱上的逻辑查询推理。载于 Aidong Zhang 和 Huzefa Rangwala（编），*KDD '22*：第*28*届*ACM SIGKDD*知识发现与数据挖掘会议，第615–625页。ACM，2022b年。

Jena D. Hwang、Chandra Bhagavatula、Ronan Le Bras、Jeff Da、Keisuke Sakaguchi、Antoine Bosselut 和 Yejin Choi。（comet-）ATOMIC 2020：关于符号与神经常识知识图谱。*AAAI*人工智能会议论文集，第6384–6392页，2021年。Ihab F Ilyas、Theodoros Rekatsinas、Vishnu Konda、Jeffrey Pound、Xiaoguang Qi 和 Mohamed Soliman。Saga：用于大规模持续构建和提供知识的平台。载于*2022*年国际数据管理会议论文集，第2259–2272页，2022年。Jeff Johnson、Matthijs Douze 和 Hervé Jégou。利用 GPU 进行十亿规模的相似性搜索。*IEEE Transactions on Big Data*，7(3):535–547，2019年。Thomas N. Kipf 和 Max Welling。利用图卷积网络进行半监督分类。载于*ICLR*，2017年。Erich Peter Klement、Radko Mesiar 和 Endre Pap。三角范数，第8卷。施普林格科学与商业媒体，2013年。Alexander Kolesnikov、André Susano Pinto、Lucas Beyer、Xiaohua Zhai、Jeremiah J. Harmsen 和 Neil Houlsby。UVim：一种利用学习得到的引导代码进行视觉建模的统一方法。载于神经信息处理系统进展，2022年。URL https://openreview.net/forum?id=lxsL16YeE2w。Bhushan Kotnis、Carolin Lawrence 和 Mathias Niepert。利用双向序列编码器求答知识图谱中的复杂查询。载于*AAAI*人工智能会议论文集，第4968–4977页。AAAI出版社，2021年。Timothée Lacroix、Nicolas Usunier 和 Guillaume Obozinski。用于知识库补全的规范张量分解。载于国际机器学习会议，第2863–2872页。PMLR，2018年。Kalev Leetaru 和 Philip A Schrodt。GDELT：1979–2012年的全球事件、地点和情感数据。载于*ISA*年度大会，第2卷，第1–49页。Citeseer，2013年。Jens Lehmann、Robert Isele、Max Jakob、Anja Jentzsch、Dimitris Kontokostas、Pablo N Mendes、Sebastian Hellmann、Mohamed Morsey、Patrick Van Kleef、Sören Auer 和 Christian Bizer。DBpedia——从维基百科中提取的大规模多语言知识库。语义网，6(2):167–195，2015年。Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich Küttler、Mike Lewis、Wen-tau Yih、Tim Rocktäschel 等。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，33:9459–9474，2020年。Tian Li、Anit Kumar Sahu、Ameet Talwalkar 和 Virginia Smith。联邦学习：挑战、方法与未来方向。*IEEE Signal Processing Magazine*，37(3):50–60，2020年。Xi Victoria Lin、Richard Socher 和 Caiming Xiong。通过奖励塑形进行知识图谱多跳推理。载于自然语言处理经验方法（*EMNLP*），2018年。Xueyuan Lin、Chengjin Xu、Fenglong Su、Gengxian Zhou、Tianyi Hu、Ningyuan Li、Mingzhi Sun、Haoran Luo 等。TFLEX：用于时序知识图谱复杂推理的时序特征逻辑嵌入框架。*arXiv*预印本 *arXiv:2205.14307*，2022a年。Xueyuan Lin、Gengxian Zhou、Tianyi Hu、Li Ningyuan、Mingzhi Sun、Haoran Luo 等。FLEX：用于复杂知识图谱推理的特征逻辑嵌入框架。*arXiv*预印本 *arXiv:2205.11039*，2022b年。Lihui Liu、Boxin Du、Heng Ji、ChengXiang Zhai 和 Hanghang Tong。知识图谱上的神经逻辑查询求答。载于第*27*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，第1087–1097页，2021年。Xiao Liu、Shiyu Zhao、Kai Su、Yukuo Cen、Jiezhong Qiu、Mengdi Zhang、Wei Wu、Yuxiao Dong 和 Jie Tang。掩码与推理：针对复杂逻辑查询预训练知识图谱 Transformer。载于第*28*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，2022年。

Xiao Long、Liansheng Zhuang、Aodi Li、Shafei Wang 和 Houqiang Li。用于在知识图谱上求答一阶逻辑查询的基于神经网络的混合概率查询嵌入。载于 *2022*年自然语言处理实证方法会议论文集，*EMNLP 2022*，第 3001–3013 页。计算语言学协会，2022年。

Haoran Luo、Haihong E、Yuhao Yang、Gengxian Zhou、Yikai Guo、Tianyu Yao、Zichen Tang、Xueyuan Lin 和 Kaiyang Wan。NQE：用于超关系知识图谱上复杂查询回答的 N元查询嵌入。载于 *AAAI 2023*，2023年。

Zhezheng Luo、Jiayuan Mao、Joshua B. Tenenbaum 和 Leslie Pack Kaelbling。关于超图上关系神经网络的表达能力与学习，2022年。URL https://openreview.net/forum?id=HRF6T1SsyDn。

Francois Luus、Prithviraj Sen、Pavan Kapanipathi、Ryan Riegel、Ndivhuwo Makondo、Thabang Lebese 和 Alexander Gray。用于复杂查询回答的逻辑嵌入。*arXiv* 预印本 *arXiv:2103.00418*，2021年。

Gengchen Mai、Krzysztof Janowicz、Bo Yan、Rui Zhu、Ling Cai 和 Ni Lao。用于在不完整知识图谱上求答逻辑查询的上下文图注意力。载于第*10*届知识捕获国际会议论文集，第171–178页，2019年。

Gengchen Mai、Krzysztof Janowicz、Ling Cai、Rui Zhu、Blake Regalia、Bo Yan、Meilin Shi 和 Ni Lao。Sekge：一种用于地理问答和空间语义提升的位置感知知识图谱嵌入模型。*GIS*汇刊，24(3)：623–655，2020年。

Stanislav Malyshev、Markus Krötzsch、Larry González、Julius Gonsior 和 Adrian Bielefeldt。从维基数据中获取最大价值：维基百科知识图谱中的语义技术应用。载于第*17*届国际语义网会议论文集（*ISWC'18*），第11137卷计算机科学讲义，第376–394页。施普林格，2018年。

Michael McCloskey 和 Neal J Cohen。在<style id='1'>学习与动机心理学，第 24 卷，第 109–165 页。爱思唯尔，1989年。

Grégoire Mialon、Roberto Dessì、Maria Lomeli、Christoforos Nalmpantis、Ram Pasunuru、Roberta Raileanu、Baptiste Rozière、Timo Schick、Jane Dwivedi-Yu、Asli Celikyilmaz 等。增强型语言模型：一项综述。*arXiv* 预印本 *arXiv:2302.07842*，2023年。

George A Miller。*WordNet*：电子词汇数据库。麻省理工学院出版社，1998年。

Mike Mintz、Steven Bills、Rion Snow 和 Dan Jurafsky。无需标注数据的远程监督关系抽取。载于 *ACL* 第 *47* 届年会与 *AFNLP* 第 *4* 届国际自然语言处理联合会议论文集，第 1003–1011 页，2009年。

T. Mitchell、W. Cohen、E. Hruschka、P. Talukdar、J. Betteridge、A. Carlson、B. Dalvi、M. Gardner、B. Kisiel、J. Krishnamurthy、N. Lao、K. Mazaitis、T. Mohamed、N. Nakashole、E. Platanios、A. Ritter、M. Samadi、B. Settles、R. Wang、D. Wijaya、A. Gupta、X. Chen、A. Saparov、M. Greaves 和 J. Welling。永无止境的学习。载于第二十九届*AAAI*人工智能会议论文集（*AAAI-15*），2015年。

Eduardo Mizraji。向量逻辑：逻辑门的基础自然代数表示。《逻辑与计算期刊》，18(1)：97–121，2008年。Boris Motik、Peter F Patel-Schneider、Bijan Parsia、Conrad Bock、Achille Fokoue、Peter Haase、Rinke Hoekstra、Ian Horrocks、Alan Ruttenberg、Uli Sattler 等。OWL 2 Web 本体语言：结构规范与函数式语法。*W3C* 推荐标准，27(65)：159，2009年。

Thomas Neumann 和 Gerhard Weikum。RDF-3X：用于 RDF 的 RISC风格引擎。超大型数据库会议论文集，第 647–659 页，2008年。ISSN 2150-8097。URL https://doi.org/10.14778/1453856.1453927。

Maxwell Nye、ArmandoSolar-Lezama、Josh Tenenbaum 和 Brenden M Lake。通过神经程序合成学习组合规则。神经信息处理系统进展，33：10832–10842，2020年。

Thomas Pellissier Tanon、Gerhard Weikum 和 Fabian Suchanek。Yago 4：一个合理的知识库。载于欧洲语义网会议，第583–596页。施普林格，2020年。Bryan Perozzi、Rami Al-Rfou 和 Steven Skiena。Deepwalk：社会表征的在线学习。载于第*20*届*ACM SIGKDD*国际知识发现与数据挖掘会议，第701–710页，2014年。Maximilian Pflueger、David Tena Cucala 和 Egor Kostylev。GNNQ：一种针对不完整知识图谱查询回答的神经符号方法。载于国际语义网会议，2022年。Tahereh Pourhabibi、Kok-Leong Ong、Booi H Kam 和 Yee Ling Boo。欺诈检测：基于图的异常检测方法的系统性文献综述。*决策支持系统*，133：113303，2020年。Shuofei Qiao、Yixin Ou、Ningyu Zhang、Xiang Chen、Yunzhi Yao、Shumin Deng、Chuanqi Tan、Fei Huang 和 Huajun Chen。通过语言模型提示进行推理：综述。*arXiv*预印本 *arXiv:2212.09597*，2022年。Meng Qu、Junkun Chen、Louis-Pascal Xhonneux、Yoshua Bengio 和 Jian Tang。RNNLogic：学习用于知识图谱推理的逻辑规则。载于国际学习表征会议（*ICLR*），2021年。Hongyu Ren 和 Jure Leskovec。用于知识图谱多跳逻辑推理的Beta嵌入。载于*神经信息处理系统进展*，第33卷，2020年。Hongyu Ren、Weihua Hu 和 Jure Leskovec。Query2box：使用盒嵌入在向量空间中进行知识图谱推理。载于国际学习表征会议，2020年。Hongyu Ren、Hanjun Dai、Bo Dai、Xinyun Chen、Michihiro Yasunaga、Haitian Sun、Dale Schuurmans、Jure Leskovec 和 Denny Zhou。LEGO：用于知识图谱多跳问答的潜在执行引导推理。载于国际机器学习会议，第8959–8970页。PMLR，2021年。Hongyu Ren、Hanjun Dai、Bo Dai、Xinyun Chen、Denny Zhou、Jure Leskovec 和 Dale Schuurmans。SMORE：大规模知识图谱补全与多跳推理。第*28*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，2022年。Rex、Ying、Zhaoyu Lou、Jiaxuan You、Chengtao Wen、Arquimedes Canedo 和 Jure Leskovec。神经子图匹配，2020年。Mark B Ring。Child：迈向持续学习的第一步。*学会学习*，第261–292页，1998年。Tomer Sagi、Matteo Lissandrini、Torben Bach Pedersen 和 Katja Hose。RDF数据表示的设计空间。*VLDB*期刊，31(2)：347–373，2022年。Jeffrey Sardina、Callie Sardina、John D Kelleher 和 Declan O’Sullivan。盒嵌入系统中注意力机制的分析。载于《人工智能与认知科学：第*30*届爱尔兰会议，*AICS 2022*》，第68–80页。施普林格，2023年。Franco Scarselli、Marco Gori、Ah Chung Tsoi、Markus Hagenbuchner 和 Gabriele Monfardini。图神经网络模型。*IEEE*神经网络汇刊，20(1)：61–80，2008年。Michael Schlichtkrull、Thomas N Kipf、Peter Bloem、Rianne Van Den Berg、Ivan Titov 和 Max Welling。使用图卷积网络对关系数据建模。载于欧洲语义网会议，第593–607页。施普林格，2018年。Robyn Speer、Joshua Chin 和 Catherine Havasi。ConceptNet 5.5：一个开放的多语言常识图。载于*AAAI*人工智能会议论文集，第31卷，2017年。Haitian Sun、Andrew Arnold、Tania Bedrax Weiss、Fernando Pereira 和 William W Cohen。知识库查询的忠实嵌入。载于*神经信息处理系统进展*，2020年。

Zhiqing Sun、Zhi-Hong Deng、Jian-Yun Nie 和 Jian Tang。RotatE：复数空间中的关系旋转知识图谱嵌入。载于《国际学习表征会议》，2018年。Zhiqing Sun、Zhi-Hong Deng、Jian-Yun Nie 和 Jian Tang。RotatE：复数空间中的关系旋转知识图谱嵌入。载于《国际学习表征会议》，2019年。URL https://openreview.net/forum?id=HkgEQnRqYQ。Muzamil Hussain Syed、Tran Quoc Bao Huy 和 Sun-Tae Chung。基于领域知识图谱的上下文感知可解释推荐。《大数据与认知计算》，6(1):11，2022年。Zhenwei Tang、Shichao Pei、Xi Peng、Fuzhen Zhuang、Xiangliang Zhang 和 Robert Hoehndorf。TAR：跨TBox和ABox的神经逻辑推理。arXiv预印本 arXiv:2205.14591，2022年。Komal Teru、Etienne Denis 和 Will Hamilton。通过子图推理进行归纳关系预测。载于《国际机器学习会议》，第9448–9457页。PMLR，2020年。James Thorne、Majid Yazdani、Marzieh Saeidi、Fabrizio Silvestri、Sebastian Riedel 和 Alon Halevy。文本上的数据库推理。载于《计算语言学协会第59届年会暨第11届自然语言处理国际联合会议论文集（第一卷：长篇论文）》，第3091–3104页。计算语言学协会，2021a。James Thorne、Majid Yazdani、Marzieh Saeidi、Fabrizio Silvestri、Sebastian Riedel 和 Alon Halevy。从自然语言处理到神经数据库。载于《超大型数据库基金会会议录》，第14卷，第1033–1039页。VLDB基金会，2021b。Sebastian Thrun。移动机器人控制的终身学习视角。载于《智能机器人与系统》，第201–214页。爱思唯尔，1995年。Yuanyuan Tian、Wen Sun、Sui Jun Tong、En Liang Xu、Mir Hamid Pirahesh 和 Wei Zhao。IBM Db2内部的协同图与SQL分析。《超大型数据库基金会会议录》，12(12):1782–1785，2019年。Théo Trouillon、Johannes Welbl、Sebastian Riedel、Éric Gaussier 和 Guillaume Bouchard。用于简单链接预测的复数嵌入。载于《国际机器学习会议》，第2071–2080页。PMLR，2016年。Emile van Krieken、Erman Acar 和 Frank van Harmelen。可微模糊逻辑算子分析。《人工智能》，302:103602，2022年。ISSN 0004-3702。doi：https://doi.org/10.1016/j.artint.2021.103602。URL https://www.sciencedirect.com/science/article/pii/S0004370221001533。V Vapnik。《统计学习理论的本质：施普林格科学与商业媒体》。德国柏林，2013年。Shikhar Vashishth、Soumya Sanyal、Vikram Nitin 和 Partha Talukdar。基于组合的多关系图卷积网络。载于《国际学习表征会议》，2020年。URLhttps://openreview.net/forum?id=BylA_C4tPr。Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。注意力机制即所需。载于《神经信息处理系统进展》，第30卷，2017年。Petar Velickovic、Guillem Cucurull、Arantxa Casanova、Adriana Romero、Pietro Liò 和 Yoshua Bengio。图注意力网络。载于ICLR，2018年。Denny Vrandecic 和 Markus Krötzsch。维基数据：一个免费协作的知识库。《ACM通讯》，57(10):78–85，2014年。Dingmin Wang、Yeyuan Chen 和 Bernardo Cuenca Grau。不完整知识图谱查询回答中逻辑变量的高效嵌入。载于《AAAI人工智能会议论文集》。AAAI出版社，2023a。

Kai Wang、Chunhong Zhang、Jibin Yu 和 Qi Sun。用于知识图谱复杂逻辑推理的信号嵌入。载于《国际知识科学、工程与管理会议》，第255–267页。施普林格，2022a。Meng Wang、Haomin Shen、Sen Wang、Lina Yao、Yinlin Jiang、Guilin Qi 和 Yang Chen。学习哈希，以高效搜索不完整知识图谱。载于《2019年IEEE国际数据挖掘会议（ICDM）》，2019年。Siyuan Wang、Zhongyu Wei、Jiarong Xu 和 Zhihao Fan。统一结构推理与语言模型预训练，以实现复杂推理。arXiv预印本 arXiv:2301.08913，2023b。Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc Le、Ed Chi 和 Denny Zhou。自洽性提升语言模型中的思维链推理。arXiv预印本 arXiv:2203.11171，2022b。Zihao Wang、Hang Yin 和 Yangqiu Song。基准测试知识图谱复杂查询回答的组合泛化能力。载于《第三十五届神经信息处理系统会议数据集与基准赛道（第二轮）》，2021年。Zihao Wang、Yangqiu Song、Ginny Y. Wong 和 Si mon See。基于原子公式单跳推理的逻辑消息传递网络。载于《第十一届国际学习表征会议》，ICLR，2023c。URL https://openreview.net/forum?id=SoyOsp7i_l。Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、brian ichter、Fei Xia、Ed H. Chi、Quoc V Le 和 Denny Zhou。思维链提示可激发大语言模型的推理能力。载于 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho（编），《神经信息处理系统进展》，2022年。URL https://openreview.net/forum?id=_VjQlMeSB_J。Boris Weisfeiler 和 Andrey Leman。将图约化为规范形式及其中出现的代数。《Nauchno-Technicheskaya Informatsia》，2(9):12–16，1968年。G. Ryabov提供的英语译文见 https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf。Robert West、Evgeniy Gabrilovich、Kevin Murphy、Shaohua Sun、Rahul Gupta 和 Dekang Lin。通过基于搜索的问答进行知识库补全。载于《第二十三届国际万维网会议》，WWW '14，第515–526页。ACM，2014年。Longfeng Wu、Yao Zhou 和 Dawei Zhou。通过逻辑推理网络实现高阶互补推荐。arXiv预印本 arXiv:2212.04966，2022年。Zhaohan Xi、Ren Pang、Changjiang Li、Tianyu Du、Shouling Ji、Fenglong Ma 和 Ting Wang。多视图知识图谱推理。arXiv预印本 arXiv:2209.13702，2022年。Wenhan Xiong、Thi-Lan-Giao Hoang 和 William Yang Wang。DeepPath：一种用于知识图谱推理的强化学习方法。载于《自然语言处理实证方法会议（EMNLP）》，2017年。Keyulu Xu、Weihua Hu、Jure Leskovec 和 Stefanie Jegelka。图神经网络的能力有多强？载于ICLR，2019a。Keyulu Xu、Weihua Hu、Jure Leskovec 和 Stefanie Jegelka。图神经网络的能力有多强？载于《国际学习表征会议》，2019b。Zezhong Xu、Wen Zhang、Peng Ye、Hui Chen 和 Huajun Chen。用于复杂查询回答的神经符号纠缠框架。载于《神经信息处理系统进展》，2022年。Bishan Yang、Wen-tau Yih、Xiaodong He、Jianfeng Gao 和 Li Deng。为知识库中的学习与推理嵌入实体和关系。载于《国际学习表征会议》，2015年。Dong Yang、Peijun Qing、Yang Li、Haonan Lu 和 Xiaodong Lin。GammaE：用于知识图谱逻辑查询的Gamma嵌入。载于《2022年自然语言处理实证方法会议论文集》，EMNLP 2022，第745–760页。计算语言学协会，2022a。

Haotong Yang、Zhouchen Lin 和 Muhan Zhang。在开放世界假设下重新思考知识图谱评估。发表于 神经信息处理系统进展，2022b 年。URL： https://openreview.net/forum?id=5xiLuNutzJG。

Gilad Yehudai、Ethan Fetaya、Eli A. Meirom、Gal Chechik 和 Haggai Maron。从局部结构到图神经网络中的规模泛化。发表于第 *38* 届国际机器学习会议论文集，2021年。

Manzil Zaheer、Satwik Kottur、Siamak Ravanbakhsh、Barnabas Poczos、Russ R Salakhutdinov 和 Alexan- der J Smola。深度集合。发表于神经信息处理系统进展，第 30 卷，2017年。

Bohui Zhang、Filip Ilievski 和 Pedro Szekely。使用关联开放数据丰富维基数据。*arXiv* 预印本 *arXiv:2207.00143*，2022a。

Muhan Zhang、Pan Li、Yinglong Xia、Kai Wang 和 Long Jin。标注技巧：使用图神经网络进行多节点表示学习的理论。载于神经信息处理系统进展，第 34 卷，第 9061–9073 页。Curran Associates, Inc.，2021a。

Wen Zhang、Jiaoyan Chen、Juan Li、Zezhong Xu、Jeff Z Pan 和 Huajun Chen。使用逻辑与嵌入进行知识图谱推理：综述与展望。*arXiv* 预印本 *arXiv:2202.07412*，2022b。

Zhanqiu Zhang、Jie Wang、Jiajun Chen、Shuiwang Ji 和 Feng Wu。ConE：用于知识图谱上多跳推理的锥嵌入。载于神经信息处理系统进展，2021b。

Denny Zhou、Nathanael Schärli、Le Hou、Jason Wei、Nathan Scales、Xuezhi Wang、Dale Schuurmans、Olivier Bousquet、Quoc Le 和 Ed Chi。由少到多提示能够让大语言模型进行复杂推理。*arXiv* 预印本 *arXiv:2205.10625*，2022a。

Yangze Zhou、Gitta Kutyniok 和 Bruno Ribeiro。消息传递图神经网络在更大测试图上的分布外链接预测泛化能力。发表于 神经信息处理系统进展，2022b。

Zhaocheng Zhu、Zuobai Zhang、Louis-Pascal Xhonneux 和 Jian Tang。神经 Bellman-Ford 网络：用于链接预测的通用图神经网络框架。神经信息处理系统进展，34：29476–29490，2021年。

Zhaocheng Zhu、Mikhail Galkin、Zuobai Zhang 和 Jian Tang。知识图谱逻辑查询的神经符号模型。发表于 国际机器学习会议，*ICML 2022*，《机器学习研究论文集》。PMLR，2022年。