arXiv:2303.01140v2 [cs.DB] 3 Jun 2024

基于嵌入和图神经网络的知识图谱基数估计

预印本

Tim Schwabe*慕尼黑工业大学 TUM计算、信息与技术学院，
德国

Maribel Acosta慕尼黑工业大学 TUM计算、信息与技术学院，
德国

2024年6月4日

摘要

知识图谱（KG）的基数估计对于查询优化至关重要，但由于典型知识图谱中数据的半结构化特性和复杂相关性，这仍是一项具有挑战性的任务。在本研究中，我们提出了*GNCE*，一种新颖的方法，利用知识图谱嵌入和图神经网络（GNN），准确预测知识图谱上合取查询的基数。*GNCE* 首先为知识图谱中的所有实体创建具有语义意义的嵌入，然后利用图神经网络学习查询的表示，以估计查询的基数。我们通过 q-误差在多个知识图谱上评估*GNCE* ，结果表明，与基于采样、摘要和（机器）学习的最先进方法相比，它的估计准确度更高，同时执行时间更短、参数更少。此外，我们还表明，*GNCE* 在真实世界查询上同样表现出色，并且能够归纳泛化到未见实体，因此适用于动态查询处理场景。我们提出的方法有望显著改善查询优化，以及其他依赖于准确估计合取查询基数的相关应用。

关键词 基数估计 · 知识图谱 · 机器学习

1 引言

知识图谱（KGs）支持使用资源描述框架（RDF）等半结构化数据模型来表示互联数据，Cygniak 等 [2014]。在知识图谱中，陈述以图的形式表示，其中带标签的节点通常对应实体，而有向的带标签边则表示节点之间的连接。如今，知识图谱正推动问答、推荐系统和语义数据库等多种应用的发展，Noy 等 [2019]。

与其他数据库技术类似，知识图谱中编码的数据可以使用 SPARQL 这类结构化查询语言进行查询Prud’hommeaux 和 Seaborne [2008]。具体而言，针对知识图谱的合取查询也可以表示为图，其中节点和边可以是常数（即在知识图谱中匹配的值），也可以是变量（即占位符）。给定合取查询的答案通过图模式匹配过程计算得出，其中查询基数对应于知识图谱中与查询图同态的子图数量。为了高效执行知识图谱上的（合取）查询，查询引擎会采用优化技术，并依赖对子查询的基数（或选择率）进行估算。这里的基数定义为执行查询所产生的答案数量。优化器依据这些估算来确定子查询的执行顺序

*部分工作在波鸿鲁尔大学完成 m.

以尽量减少中间结果的数量，从而加快查询执行速度。知识图谱查询优化面临的挑战之一是合取查询通常包含许多连接；在此情境下，当查询图的子图通过同一个变量相连时，就会发生连接。例如，每对连接到同一实体的边都可以视为一次连接。另一项挑战来自知识图谱的半结构化特性，这可能导致数据分布偏斜，以及实体与谓词之间存在不规则的相关性。这些情况难以通过传统统计摘要准确捕捉Neumann 和 Moerkotte [2011], Stefanoni 等人[2018] ，尤其是针对不同类别的图查询。

为克服统计摘要的局限性，近期提出了基于机器学习（ML）的方法来解决查询基数估计问题 Davitkova 等 [2022], Zhao 等人 [2021]。这些模型依靠神经网络（NN）或图神经网络（GNN）来学习知识图谱中的底层数据分布。为此，模型以监督方式使用抽样得到的查询图及其对应的基数进行训练。尽管研究表明，与其他摘要相比，这些方法能够更有效地捕捉任意相关性，但它们仍存在若干缺点。首先，在学习过程中，这些方法没有利用知识图谱中实体和谓词各自的特征；也就是说，初始表示是由缺乏含义或相关信息的向量构成的。其次，其中一些方法在归纳情形下表现不佳，即当查询中出现模型未见过的实体时，或者在极端情况下，查询仅由未见实体构成时。这种归纳设置会出现在用于训练的查询样本未涵盖知识图谱中所有实体的情况下。第三，最先进模型的训练需要优化大量参数，这意味着要获得准确的预测，就需要大量训练样本。这严重制约了最先进的机器学习方法扩展至大型知识图谱的能力。

在本文中，我们提出了*GNCE*，即图神经网络基数估计，这是一种旨在缓解当前最先进方法缺陷的解决方案。*GNCE* 可用于预测以 SPARQL 编写的 RDF 图上的合取查询的基数。*GNCE* 基于图神经网络（GNN），因为这是一种表达能力强的模型 Xu 等人 [2019]，能够自然地捕获查询的图结构。此外，我们的方法利用知识图谱嵌入（KGE）领域的最新进展 Rossi 等人 [2021] ，为查询元素提供具有语义意义的特征。KGE 是低维向量，用于表示实体和谓词，并捕获有关知识图谱的潜在语义信息。KGE 为*GNCE*带来两项关键优势：从紧凑且有意义的表示中学习数据相关性，以及泛化到训练查询中未见过的实体。最后，由于查询图的含义和结构大多已由 KGE 和 GNN 覆盖，*GNCE* 可以采用相对简单的架构，所需的可学习参数较少。这使*GNCE* 适用于大型知识图谱，从而克服当前解决方案的第三项局限。总之，本文的贡献如下：

1. 我们提出了一种新颖的监督模型，称为*GNCE*，用于知识图谱上的查询基数估计。该模型有效结合了图神经网络和知识图谱嵌入。2. 我们的解决方案在归纳情形下具有鲁棒性，即模型能够有效处理涉及未见实体的查询。3. 我们构建了一种快速、轻量且参数和数据高效的机器学习架构，即*GNCE* 能够扩展至大型知识图谱。4. 我们使用文献中的基准数据集表明，*GNCE* 的平均表现优于最先进水平。

2 预备知识

2.1 知识图谱

知识图谱（KG）定义为一种有向图，其中节点和边均由标签标识 Hogan 等 [2022]。此外，节点可以被赋予类[2], , 这些类对应于具有某些共同特征的节点群。知识图谱有多种数据模型。本文采用资源描述框架（RDF）的定义 Cyganiak 等 [2014], , 其中每条陈述都建模为一个三元组。为简单起见，RDF 中定义的项（即国际化资源标识符、空白节点[3], 和字面量）在知识图谱中均视为标签，并表示为术语全集 $\mathcal{U}$。

定义 1 （知识图谱）。知识图谱 $\mathcal{G}$ 是一个有向标记图，以三元组集合的形式表示，其中每个三元组 $t = (s, p, o)$ 表示图中节点之间一条带标签的边。在每个三元组中，t, s 是主语，p是谓词，o 是宾语，并且 $s, p, o \in \mathcal{U}$，其中 $\mathcal{U}$是一个集合。三元组的元素称为原子。

[2]在文献中，类有时也被称为标签，但不应将其与标签/标识符混淆。[3]以下定义假设斯科伦化空白节点，即转换为国际化资源标识符。

针对知识图谱的问题可以表述为结构化查询。在本研究中，我们关注合取查询，它也可以定义为一种图，其中原子要么实例化为（已绑定的）项，要么是变量，充当可在知识图谱中匹配的占位符。

定义 2（查询图）。考虑满足 $\mathcal{V} \cap \mathcal{U} = \emptyset$ 的变量集合 $\mathcal{V}$。查询图 Q 是由一组三元组模式表示的有向标记图，其中每个三元组模式 $tp = (s_q, p_q, o_q)$ 表示查询图中节点之间的一条带标签的边，且 $s_q, p_q, o_q \in \mathcal{U} \cup \mathcal{V}$。

随后，通过在知识图谱中匹配相应的查询图来找到查询的答案。这个过程称为模式匹配，即将原子变量映射到图中的元素。查询解是知识图谱中与查询图同态的子图。具体而言，在包含绑定原子的知识图谱中（即知识图谱中没有变量），这等价于直接替换查询中的变量，并判断所得图是否为知识图谱的子图。

定义 3（查询解）。给定知识图谱 $\mathcal{G}$ 和查询图 Q，考虑映射 $\mu: \mathcal{V} \mapsto \mathcal{U}$。我们将 $\mu(Q) \subseteq \mathcal{G}$，则 $\mu(Q)$ 是 $\mathcal{G}$ 的子图，并称为 $\mathcal{G}$ 上 Q 的查询解。

基数是查询的解的数量：

定义 4 (查询基数)。 合取查询 Q 在 $\mathcal{G}$上的基数，记为 $||Q||_{\mathcal{G}}$，是不同解的数量 f 不同的解，如定义 3：

$$||Q||_{\mathcal{G}} = |\{\mu : \mu(Q) \subseteq \mathcal{G} \ \wedge vars(Q) = dom(\mu)\}|$$

$vars(Q)$ 表示在 Q中出现的变量，而 $dom(\mu)$ 表示 μ中的变量集合。

2.2 知识图谱嵌入

知识图谱嵌入（KGE）是知识图谱中原子（实体和谓词）的低维向量。这些向量编码了有关原子的潜在语义信息，而这些信息并未在知识图谱中显式呈现。因此，与仅依赖知识图谱中（显式）陈述和结构的方法相比，知识图谱嵌入通常能在下游任务中取得更好的结果，Rossi 等人 [2021], 例如链接预测、实体分类、聚类等。

为计算知识图谱嵌入，现有的知识图谱表示学习模型包含一个嵌入空间、一个评分函数 f和一个合适的损失函数 l。嵌入空间通常是实数或复数上的向量空间，维度从几十到几百不等。评分函数 f为三元组（s，p，o）赋予一个分数；分数越高，表示该三元组在模型 $\mathcal{G}$ 看来越可能存在于知识图谱中，分数越低则表示其概率越低。损失函数 l 用于拟合模型，并促使模型为真三元组 ——即知识图谱中被认为正确的陈述——赋予高分，为假三元组 ——即知识图谱中不存在且不应存在的陈述——赋予低分。Rossi 等人对最先进的知识图谱嵌入进行了概述。Rossi 等人 [2021]。

由于组件和学习过程各不相同，不同的知识图谱嵌入模型能够捕捉知识图谱中不同的潜在方面，因此某些知识图谱嵌入模型更适用于特定任务。例如，一些知识图谱嵌入模型（如平移模型 Rossi 等人 [2021]）从知识图谱中的单个三元组进行学习，因此非常适合链接预测。其他模型（如 RDF2Vec Ristoski 和 Paulheim [2016] 或 Ridle Weller 和 Acosta [2021]）则可以从子图等更大型结构中学习，因此适用于需要考量原子之间语义相似性和相关性的任务，例如实体类型预测或聚类。在这项工作中，我们关注后一类知识图谱嵌入模型，因为从子图中学习能够有效捕捉知识图谱中原子的相关性。RDF2Vec Ristoski 和 Paulheim [2016] 通过对知识图谱中的原子进行随机游走来实现这一点，其采用了跳字模型 Mikolov 等人 [2013] 。知识图谱上的随机游走从任意节点开始，通过沿着一条边前往另一个节点，依此类推，遍历图而生成。给定一个长度为 W 的知识图谱中的原子随机游走 $w_1, ..., w_W$（其中包括节点和谓词），RDF2Vec 的目标是最大化以下对数概率（即最小化其负值，该值可视为知识图谱嵌入中的损失函数）：

$$\frac{1}{W}\sum_{t=1}^{W}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

也就是说，给定一个目标原子 w_t，RDF2Vec 会尝试最大化游走中距离不超过 c 跳的所有原子（对 j 求和）的概率。RDF2Vec 会对随机游走中的每个原子（对 t 求和）执行此操作。给定另一个原子 w_i 时，原子 w_o 的条件概率可以视为其他知识图谱嵌入方法中评分函数的对应形式，其定义如下：

$$p(w_o|w_i) = \frac{\exp(v'^{\top}_{w_o} v_{w_i})}{\sum_{w=1}^{|V\cup E|}\exp(v'^{\top}_{w} v_{w_i})} \tag{2}$$

这里，v_{w_j} 表示原子 w_j 的输入向量（嵌入），而 v'_{w_j} 表示 w_j 的所谓输出向量。它们与输入向量的维度相同，但不会用作最终嵌入。引入它们是为了增强模型的表达能力，并为目标原子和上下文原子提供不同的表示。我们可以看到，未归一化的条件概率通过两个原子的输入向量与输出向量之间点积的指数来估计。当两个向量之间的夹角为零时，该点积达到最大值。如果两个向量相同，或者位于同一个一维子空间中，就会出现这种情况。因此，式（1）的优化会促使具有相同上下文的原子的嵌入彼此接近。对于那些在图中共同出现的原子，这一点尤其明显，也就是说，它们同时出现的联合概率较高。请注意，通过除以所有可能概率之和（通常称为配分函数），可以得到归一化概率。这个总和难以计算，实际中通常采用近似方法，见 Ristoski 和 Paulheim [2016]。

这里的重要结论是，RDF2Vec 生成的嵌入编码了原子的相关性，并且与观测到这些原子的联合概率相关。乍看之下，嵌入似乎编码的是采样路径，但由于单个实体会参与多次游走，因此相邻节点和边中的丰富上下文信息也被编码其中。因此，我们的假设是，RDF2Vec 嵌入适用于知识图谱上查询基数估计中的原子表示。

2.3 图神经网络

图神经网络（GNN）是一种专为图结构数据设计的特殊形式的神经网络（NN）。GNN 的显著特点是，对于输入图中节点的置换具有不变性 Bronstein 等人 [2021]。这意味着，即使图中的原子重新排序，模型的输出也保持不变。置换不变性使 GNN 在参数和数据方面都更加高效 Bronstein 等人 [2021]。该模型具有参数高效性，因为它所需的可学习参数更少，不必为同一图结构学习不同的置换形式。它也具有数据高效性，因为模型只需相对较少的样本即可取得良好效果。GNN 的一个典型应用任务是图级预测。在此任务中，给定一个作为输入的图，GNN 预测适用于整个图的量。例如，预测一个分子图是否具有毒性，或一个社交网络是否存在种族歧视。对我们而言，给定查询图来预测其基数，可以视为一种图级预测。

图神经网络在消息传递框架下运行，在图神经网络的每一层 k 中，节点的输入表示都会根据来自所有相连邻居的消息 进行更新。节点最通用的消息传递函数 由Team [2022]给出：

$$x_i^{(k)} = \gamma^{(k)} \left(x_i^{(k-1)}, \square_{j \in \mathcal{N}(i)} \phi^{(k)} \left(x_i^{(k-1)}, x_j^{(k-1)}, e_{j,i} \right) \right) \tag{3}$$

这里，γ 和 ϕ 是可微函数，而 $\square$ 既是可微函数，也是置换不变函数。$\mathcal{N}_i$ 是第 i个节点的直接邻居节点集合。对于节点 x_i，其先前表示记为 $x_i^{(k-1)}$，通过结合 $x_i^{(k-1)}$、相邻节点的表示以及它们之间的边（即消息），经过非线性变换后得到新的表示 $x_i^{(k)}$。为了进行图分类，需要将所有节点的个体节点特征合并为一个定长向量。因此，适用于图中节点数可变情形的置换不变函数是合适的。实际应用中的例子包括最大池化、平均池化或求和池化。由此得到的向量表示整个图，可用于预测所需的图级量 Wu 等人 [2019]。

节点 x_i^0 的初始表示取决于具体应用场景。可选方案包括使用独热编码或二进制编码表示节点 ID，也可以使用信息量更丰富的特征。例如，如果节点表示社交图中的一个人，一个维度可以表示此人的身高，另一个维度可以表示年龄，等等。更有意义的特征更为优越，因为它们能为图神经网络提供关键信息，以解决所需的问题。此外，它们还使图神经网络无需学习节点的内部表示；而使用独热编码或二进制编码时则不然（因为特征除了标识节点身份外不承载任何含义），节点的任何语义都需要从数据中学习并整合到模型中。由于知识图谱嵌入能够提供有关给定实体的语义信息，因此它们是作为初始节点特征的一种很有前景的选择。

3 相关工作

基数估计是关系数据库领域一个长期存在的研究问题，相关解决方案也被用作知识图谱等图结构数据的研究起点。传统方法依赖数据集摘要，但近年来，基于学习的解决方案也已陆续提出。下文将讨论关系数据和图结构数据领域的相关工作。

关系数据库的方法 传统上，基数估计基于采样和直方图。理论上，直方图可以提供属性和连接谓词的完整联合概率，但其计算开销过高，因为直方图会随属性数量呈指数级增长。因此，人们通常会对数据采用独立性和均匀性假设，这会导致基数估计误差很高。Deeds 等人 [2023]通过计算基数估计的上界来降低这些误差。其他克服这一问题的方法，例如 Cormode 等人 [2011], 、Vengerov 等人 [2150], 、Li 等人 [2016], 提出的采样方法，则不对数据作任何独立性假设。通过在数据样本上执行查询，可以在该样本上计算完整的联合概率。随着样本规模接近实际数据规模，针对数据的真实基数也会逐渐得到逼近。然而在实践中，为了在合理时间内计算估计值，样本规模通常远小于数据集。因此，估计的准确性高度依赖于采样技术和数据分布，而且可能需要长时间采样才能得到良好的估计。采样领域的方法包括 Vengerov 等人 [2150], 提出的相关采样 、Li 等人[2016], 提出的*Wanderjoin* ，以及 Zhao 等人 [2018]提出的带上界的连接采样（*jsub*） 。Vengerov 等人 [2150]提出的相关采样 通过依据属性的哈希值进行采样，改进了简单的伯努利采样，从而降低估计值的方差。Li 等人 [2016] 提出的*Wanderjoin* 将查询和数据库分别表示为查询图和数据图，并按照查询图的结构在数据图上进行随机游走，以估计基数。带上界的连接采样 采用与*Wanderjoin* 类似的采样方法，但会扩展估计结果以提供基数上界。Park 等人 [2020]总结了这些方法。若要全面了解基数估计中的采样和直方图方法，可参阅 Cormode 等人撰写的综述。Cormode 等人[2011]。

为避免统计摘要带来的计算和内存需求呈指数级增长，人们提出了学习方法。Sun 等人 Sun 等人 [2021] 将这些方法归为所谓的数据模型和查询模型。

数据模型首先尝试预测数据的概率，然后通过从学习得到的概率分布中采样来估计查询基数。这些方法的示例基于贝叶斯网络 Getoor 等人[2001], Tzoumas 等人 [2011], Halford 等人 [2019]。所有这些方法都旨在通过将联合概率分解为较小条件概率的乘积来表示联合概率 Halford 等人 [2019]。问题在于，实际上，这种分解可能需要捕获数据中许多属性的联合相关性，从而使网络规模也呈指数级增长。因此，贝叶斯网络在实践中仍在很大程度上采用独立性假设 Halford 等人 [2019]。为克服这些局限，DeepDB 中实现的和积网络 Hilprecht 等人 [2020] 被用作图模型，以近似数据的联合概率。与贝叶斯网络相比，和积网络的内存占用相对于数据库规模呈多项式增长。

相反，查询模型旨在根据查询预测基数。大多数此类方法都基于神经网络（NN）。例如，Kipf 等人 Kipf 等人 [2019] 提出了多集合卷积网络（MSCN），通过使用表、连接和谓词集合来表示查询。Woltmann 等人 Woltmann 等人 [2019] 也提出了一种基于神经网络的方法，但该方法专注于数据的局部部分。Zhao 等人 Zhao 等人 [2022]提出的解决方案基于神经网络高斯过程（NNGP），用于处理不确定性，从而提高预测的准确性。尽管后者的解决方案与我们的工作更为接近，但这些神经网络是针对关系上的查询定制的，无法直接应用于图。

图结构数据的方法 早期针对图的基数估计研究，借鉴了关系数据库中的解决方案，使用直方图并假设属性独立性 Stocker 等人 [2008],Shironoshita 等人 [2007], Neumann 和 Weikum [2009]。较新的方法则尝试对查询结构之间的相关性进行建模。Neumann 等人 Neumann 和 Moerkotte [2011] 提出了特征集 （*CSET*），这是一种统计摘要，用于统计具有相同谓词集合的实体数量。对于所有谓词均已实例化且宾语不受限制（即为变量）的星形 DISTINCT 查询，*CSET* 能够给出精确的基数估计值。对于其他类的查询，Neumann 等人 Neumann 和 Moerkotte [2011] 将查询分解为星形子查询，使用*CSET*计算其基数，并通过假设子查询之间相互独立来估计最终基数。*SUMRDF* Stefanoni 等人 [2018] 同样基于图摘要，并计算类型化 摘要，将属于同一类[4] 且谓词分布相似的节点归为一组。除了摘要之外，也有人提出了面向知识图谱的基于采样的方法。例如，*impr* Chen 和 Lui [2017] 通过在数据上进行与查询图匹配的随机游走来估计查询基数。最后，也有人提出了面向知识图谱的贝叶斯网络方法 Huang 和 Liu [2011]。上述所有解决方案都是专门针对 RDF知识图谱设计的。不过，与关系数据库中的对应方法类似，这些方法计算出的结构可能会随着数据集规模的增长而迅速膨胀，或者仅对特定类的查询准确。

[4]在 RDF 知识图谱中，点的类由谓词 rdf:type 提供。

表1：学习型方法的特征

方法	机器学习模型	原子表示	模型大小	嵌入	置换不变性	假设	归纳性（经验）
*LMKG*Davitkova 等 [2022]	NN	二进制编码	$O(\|V\|+\|E\|)$	-	✗	–	实体、查询形状
LSS Zhao 等人 [2021]	GNN	ProNE Zhang 等 [2019]	常数	$\|V\|$	✓	类型化实体、无向图	实体、查询形状
GNCE（本文方法）	GNN	RDF2Vec Ristoski 和 Paulheim [2016]	常数	$\|V\|+\|E\|$	✓	–	实体、查询形状

与我们工作最相似的是*LMKG* Davitkova 等人 [2022] 和*LSS* Zhao 等人 [2021], 提出的基于机器学习（ML）的方法。表1概述了这些解决方案，接下来将对其进行详细讨论。
LMKG Davitkova 等 [2022] 是一种监督模型[5]，用于利用神经网络（NN）学习基数；该神经网络由多个非线性全连接层组成。*LMKG* 中知识图谱和查询的表示方式如下。首先，为知识图谱中的每个原子分配一个 ID，并将其转换为该 ID 的二进制向量表示。然后，使用邻接张量以及特征化矩阵来表示包含 n 个节点和 e 条边的图查询，从而将查询中的原子与知识图谱中的原子关联起来。实际上，*LMKG* 假设所有查询的邻接张量和特征化矩阵具有相同的维度，这意味着 n 和 e 被设为工作负载中最大的查询的规模；这可能导致生成的模型较大。一种可行的解决方案是针对不同的查询规模训练多个模型 Davitkova 等 [2022], ，但这会带来管理开销。计算出表示后，神经网络的输入是展平后的查询表示的拼接。因此，*LMKG* 不具备置换不变性，因为即使是同一个查询，只要三元模式的输入顺序不同，得到的表示也会不同。这意味着*LMKG* 需要更多参数和训练数据点来捕获这些排列，并生成准确的预测。

LSS Zhao 等人 [2021]将图神经网络（GNN）应用于图查询。对于查询中的每个节点，*LSS*计算从该节点出发进行 l 跳广度优先搜索（ BFS ）所诱导的树。随后，将 GNN 应用于所有这些树，并通过注意力机制聚合结果，最终预测基数。尽管*LSS* 已被实验证明在（常规）图上的基数估计中有效，但*LSS* 对查询节点的初始表示依赖于两个方面，使其不太适用于知识图谱。首先，*LSS* 使用 ProNE Zhang 等 [2019] 嵌入来计算图查询中节点的初始表示。此类嵌入针对图而非知识图谱设计，因为它们没有考虑有类型有向边。其次，每个原子都通过将知识图谱中对应类的嵌入求和来表示。类嵌入求和表示存在一个问题：形状相同、但包含属于相同类别的不同实体的查询，会得到相同的表示。这不利于基数估计，因为这些查询的选择率可能因查询中提及的实体不同而有很大差异。此外，*LSS* 采用的消息传递函数将图视为无向图。此外，注意力机制引入了大量参数（源于注意力头中的$\mathbf{Q}$，$\mathbf{K}$，$\mathbf{V}$ 矩阵和多层感知机；Vaswani 等人 [2017]），使得最终模型规模较大（在我们的实验中，参数超过 200 万）。
LSS 还支持主动学习，主要用于加速训练。通过聚焦于表现不佳的查询，模型无需在整个数据集上进行穷尽式训练，就能提升性能。不过，主动学习与模型架构正交，因此也可以同样应用于我们的模型。

NeurSC Wang 等 [2022] 将多个图神经网络应用于查询图以及数据图的子图。后者提供了额外信息，但成本更高，因为每次查询都需要对图进行采样。此外，*NeurSC* 专为不带边标签的无向图设计，因此不适用于知识图谱。因此，我们认为，基于机器学习且与我们最接近的工作是*LMKG* 和*LSS*。此外，*LMKG* 、*LSS*以及我们的方法*GNCE*均不对数据作任何独立性或均匀性假设，因为底层神经网络原则上能够近似任意偏斜分布。

最后，在知识图谱上的查询处理中，也有一些相关研究问题已得到探讨，例如最坏情况连接顺序优化 Hogan 等 [2019] ，或查询执行性能预测 Zhang 等 [2018], Hasan 和 Gandon [2014], Casals 等。然而，这些问题的性质及其提出的解决方案与本文所研究的问题有很大不同，因此不在本文的讨论范围之内。

4 我们的方法

我们将知识图谱（KG）上的基数估计任务建模为一个回归过程，直接使用预测模型来估计给定查询图的基数。具体而言，我们关注基于 RDF 知识图谱、以 SPARQL 编写的合取查询。

[5]还有一种采用无监督学习的*LMKG* 版本，但其性能不如监督式对应版本。因此，我们仅关注*LMKG*的监督式设置。

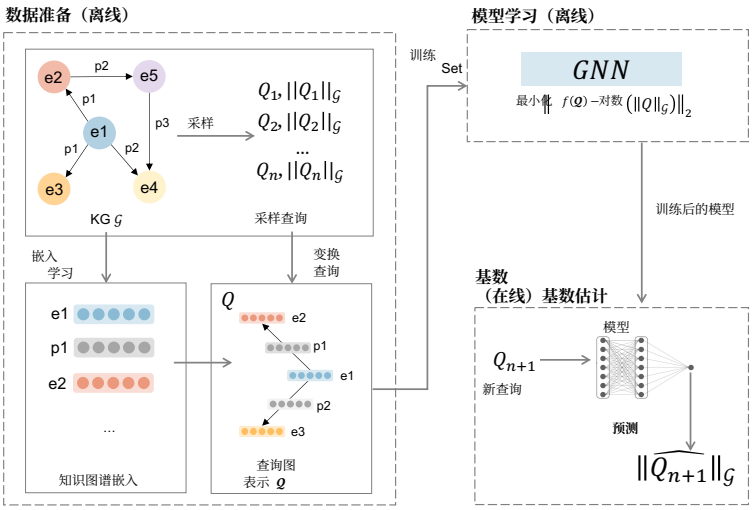


图 1：GNCE 概览。对于知识图谱，我们为已有原子学习嵌入。接下来，使用采样查询训练图神经网络，这些查询以嵌入表示。最后，训练后的模型估计新查询的基数。

我们的模型 f 旨在预测知识图谱 $\mathcal{G}$上的查询 Q 的基数。此预测基于查询的特征表示，记为 $\mathcal{Q}$（其具体内容将在下文详述）：

$$f(\mathcal{Q}) = \widehat{||Q||}_{\mathcal{G}} \approx ||Q||_{\mathcal{G}} \tag{4}$$

在此， $f(\mathcal{Q})$ 得出估计基数 $||Q||_{\mathcal{G}}$，该值近似于图中查询的真实基数 $||Q||_{\mathcal{G}}$。

为了按照公式 (4) 中所述近似真实基数，我们选择神经网络表示，因为它具有学习复杂函数的能力 Maiorov 和 Pinkus [1999]。考虑到知识图谱 (KG) 上的查询本质上构成图结构（参见定义 2），采用图神经网络 (GNN) 是恰当的选择。这一选择确保查询图具有置换不变性，同时也提高了参数效率和数据效率。

图 1 概述了我们的方法。在初始阶段，*GNCE* 准备训练机器学习模型所需的数据。第 4.1 节详细介绍了这一准备阶段，其中包括从 $\mathcal{G}$ 中采样查询、计算知识图谱嵌入，以及构建有意义的查询表示。随后，*GNCE* 使用图神经网络从这些查询表示中学习模型。学习目标是尽可能缩小模型预测基数与查询实际基数之间的差异，具体讨论见第 4.2 节。成功训练后，图神经网络模型便能够预测训练阶段未见过的新查询的基数（参见第 4.3 节）。

4.1 数据准备

数据准备的目标是获得有意义的表示，以用于学习基数估计模型。*GNCE* 中的数据准备在线下完成，包括：（1）从知识图谱中获取用于训练的查询；（2）计算查询中出现的原子的嵌入；（3）利用（1）和（2）构建每个查询的特征化。以下将对此过程进行说明。

GNCE 首先从给定的知识图谱 $\mathcal{G}$中获取一个包含查询及其真实基数的训练集。为此，*GNCE* 从 $\mathcal{G}$ 中随机采样具有不同结构的查询，例如具有不同形状（如星形结构或路径）、数量各异的三元组模式以及不同绑定项的查询。在此过程中，必须在知识图谱中进行均匀采样，以尽可能覆盖数据分布，而不是仅拟合可能的用户查询所在的某些区域。然后，对于每个采样的查询 Q，计算 $\mathcal{G}$ 中每个绑定原子 x 的个体出现次数 $occ(x)$。也就是说， $occ(x)$是 $\mathcal{G}$ 中包含该原子 x 的三元组数量，无论该原子出现在主语、谓词还是宾语位置。这些数量有助于逼近整体查询基数：原子出现次数越多， $\mathcal{G}$ 的子图匹配查询图 Q的概率就越高。由于 $occ(x)$值可以通过简单查询轻松获得，因此，与其让模型自行学习这些值，不如直接将它们提供给模型。

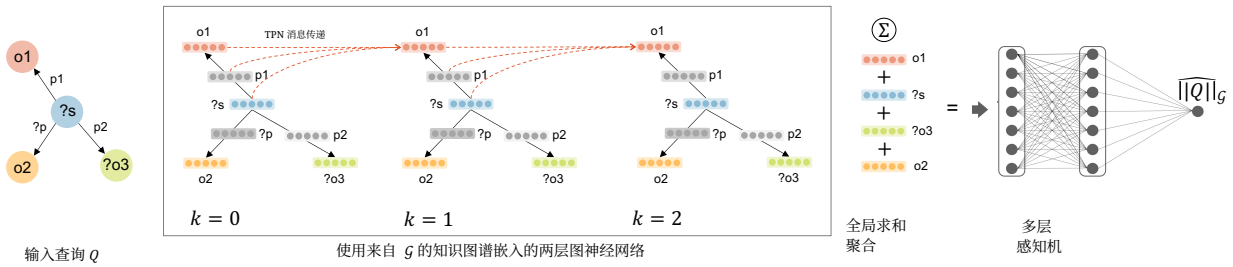


图 2：用于基数估计的图神经网络架构。初始查询由实体和谓词嵌入以及这些原子之间的连接关系表示。为清晰起见，此处仅展示一个节点的 TPN 聚合。节点聚合后，通过对所有节点嵌入求和来表示完整的查询图。最后，将查询表示输入多层感知机并映射到单一维度，从而估计查询基数。

接下来，*GNCE* 使用 RDF2Vec Ristoski 和 Paulheim 为 $\mathcal{G}$ 中的原子（节点和谓词）计算嵌入 $e(x)$。如第 2.2 节所述，由于 RDF2Vec 嵌入能够编码相关性和联合概率，因此我们选择它作为嵌入方法。使用 RDF2Vec 时，节点（包括国际化资源标识符和字面量）及谓词都被视为随机游走中的元素，因此，RDF2Vec 能够为知识图谱中的所有类型原子生成嵌入。这样，*GNCE*也支持字面量，尽管它们与国际化资源标识符被同等对待（与所有比较方法类似）。也就是说，不考虑不同数据类型中字面量之间的序数相似性，我们将特殊处理（例如 Kristiadi 等人 [2019], Preisner 和 Paulhei m [2023]）留待未来工作。

此外，*GNCE* 仅针对采样查询中出现的实体训练 RDF2Vec 嵌入，而非整个知识图谱，这大幅提升了大型图的训练速度（详情请参阅 Portisch 等人 [2020] ）。这样做的依据是，我们观察到，在整个图上训练时，模型最终的性能与此相近。不过，在生产环境中，可能需要为所有实体（或至少为查询中频繁出现的实体）计算嵌入。请注意，嵌入学习过程发生在基数估计模型训练之前，之后已有实体的嵌入将保持固定。第 4.3 节将讨论训练后如何处理新实体。我们假设，测试期间以及之后对新查询进行估计时会出现的所有谓词，都已包含在训练集中。

最后，对于每个采样的查询 Q，*GNCE* 会计算一个表示 $\mathcal{Q}$。该表示 $\mathcal{Q}$ 通过将查询图 Q 中的每个原子替换为一个对学习基数模型具有实际意义的最终输入特征来构成。在这里，*GNCE*为原子 x构建最终输入特征 $i(x)$，该原子可以是绑定原子或变量原子。对于绑定原子 x，*GNCE*将其对应的嵌入 $e(x)$与计算得到的出现次数 $occ(x)$进行拼接，即 $i(x) = e(x)||occ(x)$。查询图中的变量原子需要一种特殊的向量表示。对于每个查询，*GNCE*会随机为每个变量分配一个递增的数字标识符（例如，*?s1* 获得标识符 2，*?p1* 获得标识符 1，依此类推）。然后，*GNCE*按如下方式为变量原子构建向量 $i(x)$。将向量的第一维设为该标识符，其余维度全部设为 1。这样，模型就能获知哪些原子是变量，以及哪些变量彼此相同。由于分配过程是随机的，模型不会错误地学习到某个特定标识符对应某个特定位置或标签。这一点不同于其他方法，例如 Davitkova 等 [2022], Zhao 等人 [2021], 的方法，在这些方法中，查询中的不同变量会用 0 进行编码，并被视为同一个变量。最后，特征化 $\mathcal{Q}$ 由 $i(x)$ $\forall x$给出。通过这种方式，*GNCE* 会联合编码局部邻域、与其他原子的相似性，以及有关实体的统计信息。

4.2 模型学习

本节介绍模型定义，说明*GNCE* 如何从数据准备期间获得的表示 $\mathcal{Q}$ 中学习，随后介绍模型训练 的细节，说明 *GNCE* 如何利用真实查询基数进行监督学习并做出预测。

模型定义。 图 2 展示了*GNCE* 的架构，该架构由一个图神经网络（GNN）组成，它通过多层感知机（MLP）实现消息传递层。

表示 $\mathcal{Q}$ 是我们图神经网络的输入（$k = 0$）。首先，对输入应用两个消息传递层（$k = 1$， $k = 2$）。在初步研究中，我们发现，增加层数并不能提升模型性能。对此可能有以下解释：（i）查询图规模较小，因此两步消息传递已足够；（ii）较少的层数可以避免过平滑问题 Chen 等人 [2020], ，即节点表示变得越来越相似，从而降低模型性能。

对于图神经网络中的消息传递层，我们提出了一种自定义的消息传递函数。我们将该函数称为*TPN* 消息传递，它是对 *GINECONV* Hu 等人 [2020] 定义的消息传递函数的调制，具体定义如下：

$$x_i^{(k)} = h_\theta^{(k)}\left((1+\epsilon)x_i^{(k-1)} + \sum_{j\in\mathcal{N}(i)} \mathrm{ReLU}(x_j^{(k-1)} + e_{j,i})\right) \tag{5}$$

我们最初评估了多个不同的消息传递层，但发现 *GINECONV* 优于其他方法。这与 Xu 等人 Xu 等人 [2019], 的理论发现一致：*GINECONV* 的表达能力严格强于其他消息传递层。这也与*LSS* Zhao 等人 [2021], 的结果相符，他们同样使用*GINECONV* 作为消息传递函数。

在公式 (5) 中，每个 $h_\theta^{(k)}$ 都是带有 ReLU 激活函数的多层感知机（MLP）。 $\mathcal{N}(i)$ 是节点 i 的所有相邻节点及其对应的边$e_{j,\ i}$。参数 ϵ 是一个可选的可学习参数，默认值为零。因此，该函数首先分别对相邻节点和边求和，然后通过求和聚合这些结果以及节点 $x_i^{(k-1)}$ 当前的表示，最后通过 h_θ对其进行非线性变换。遗憾的是，这个消息传递函数将图视为无向图，因为它聚合邻居时不考虑边的方向。然而，在基数估计中，边的方向可能是重要信息：三元组 $(s \to p \to o)$ 和 $(o \to p \to s)$ 的基数可能大相径庭。使消息传递具有方向性的一种方法是仅在每个节点聚合入边，即，不对 $\mathcal{N}(i)$ 求和，而是对 $\mathcal{N}^+(i)$ 求和。然而，我们发现，与直接对 $\mathcal{N}(i)$ 求和相比，这种方法效果更差，可能是因为它限制了信息流动。因此，我们采用以下定制的消息传递函数：

定义 5（TPN 消息传递）。三元模式网络（*TPN*）消息传递通过分别变换传入消息（来自 $\mathcal{N}^+(i)$）和传出消息（来自 $\mathcal{N}^-(i)$），来变换节点表示 $x_i^{(k-1)}$，形式如下

$$x_i^{(k)} = h_\theta^{(k)}\Bigg(x_i^{(k-1)} + \sum_{j\in\mathcal{N}^+(i)} \mathrm{ReLU}(\mathbf{W}(x_i^{(k-1)}||e^{j,i}||x_j^{(k-1)})) + \\ + \sum_{j\in\mathcal{N}^-(i)} \mathrm{ReLU}(\mathbf{W}(x_j^{(k-1)}||e^{i,j}||x_i^{(k-1)}))\Bigg)$$

此函数区分传入边和传出边。该函数不通过求和来聚合节点和边特征，而是将它们拼接起来（||），再使用矩阵 $\mathbf{W}$ 将其线性投影到与 x_i 相同的维度。请注意，无需使用两个不同的矩阵 $\mathbf{W}^+$、$\mathbf{W}^-$，因为 $\mathbf{W}$ 始终接收同一三元模式 $\mathbf{O}||\mathbf{P}||\mathbf{S}$的编码，其中 $\mathbf{O}$ 是边所指向节点的特征，$\mathbf{P}$ 是节点之间边的特征，而 $\mathbf{S}$ 是边起始节点的特征。最后，多层感知机$h_\Theta^{(k)}$ 以三元模式表示之和作为输入。图节点经两个消息传递层变换后，需要聚合为一个定长向量。为此，*GNCE* 对这些节点应用全局求和聚合。也就是说，节点表示按维度求和，得到一个与节点嵌入维度相同的单一向量 x： $x = \sum_l x_l^{(k=2)}$。

GNCE 模型的最后一步是通过最终的 2 层多层感知机（隐藏层使用 ReLU 激活函数，最终层使用线性激活函数）将该向量 x 变换为一个单一维度（即一个值）。该值表示 GNCE 对查询基数的预测。

模型训练。为了学习查询的基数，*GNCE* 采用监督式方法，使用查询表示 $\mathcal{Q}$（按前文所述经图神经网络处理）和真实基数 $||Q||_\mathcal{G}$。训练期间的目标是最小化预测基数（记为 $f(\mathcal{Q})$）与真实值之间的差异。为此，*GNCE* 采用均方误差（MSE）作为损失函数，计算模型输出 $f(\mathcal{Q})$ 与真实基数的对数 $||Q||_\mathcal{G}$ 之间的均方误差： $\mathcal{L} = ||f(\mathcal{Q}) - \log(||Q||_\mathcal{G})||_2$。因此，模型学习预测基数的对数，这能够缓解可能的基数范围较大所导致的训练不稳定问题。尽管模型以 q-误差进行评估，我们发现，使用均方误差作为损失函数比直接使用 q-误差效果更好。在所有实验中，我们以 32 作为批量大小，训练 50 轮。为了优化模型，我们使用 Kingma和Ba 的 Adam优化器 [2015]，学习率为 e^{-4}。模型训练在线下进行[6],，随后便可使用训练好的模型对不属于训练样本的新查询进行在线预测，如下文所述。

[6]相关　方法还会预先计算统计信息或样本，　线下完成。

4.3 新查询上的基数估计

现在可以使用训练好的模型来预测新查询的基数。收到查询 Q_{n+1} 后，该预测会在线计算。为此，*GNCE* 会识别 Q_{n+1} 中提到的绑定原子，并检索预先计算好的对应嵌入。对于变量原子，*GNCE* 会构建其对应向量，以获得表示 $\mathcal{Q}_{n+1}$。随后将该表示输入训练后的模型，以计算$\widehat{}$预测值 $||Q_{n+1}||_{\mathcal{G}}$。

生产环境中的模型学习与部署。 现在，我们将介绍如何在真实环境中实现 *GNCE* 。假设 KG $\mathcal{G}$ 存储在三元组存储中。为了获取样本查询，*GNCE*生成随机查询，并直接在三元组存储中执行这些查询，或通过 SPARQL 端点执行。查询的元数据（例如真实基数和 occ 值）也由 *GNCE* 通过查询三元组存储获得。随后，对于采样查询中出现的原子，*GNCE* 生成随机游走——同样查询三元组存储——并利用这些随机游走学习 RDF2Vec 嵌入。预先计算的嵌入存储在可快速访问的存储系统中，例如向量数据库。然后，*GNCE* 训练图神经网络模型，并将其编译后部署到生产环境中。当新查询到达时，从向量数据库中检索相关嵌入，以构建查询的特征化，并由图神经网络直接预测基数。

生产环境中的一个重要方面是在知识图谱中添加新实体。如果查询中的某个新实体尚无对应的嵌入，*GNCE* 会生成一个随机嵌入来表示该实体。从理论上讲，由于模型是在未知实体上进行预测，这可能会降低性能。不过在实践中，*GNCE* 的性能仍然令人满意（见第 5.5 节）。这确保了该方法能够处理任何查询，无论所有嵌入是否都已存在。最终，经过一段时间或添加一定数量的新实体后，*GNCE* 可以离线学习新实体的嵌入。在此过程中，*GNCE* 还会更新与新实体相关的原子的出现次数 $occ(x)$；不过，这是一项轻量级操作。随后，*GNCE* 会将新嵌入添加到向量数据库中，使部署的图神经网络模型能够使用这些嵌入。请注意，所有依赖预计算数据（例如摘要、样本或机器学习模型）的方法也会受到知识图谱变化的影响，因此也需要更新其结构以涵盖新实体。

5 实验

我们通过实证研究考察所研究方法的有效性（§5.2、§5.3、§5.4）、这些方法对未见实体的泛化能力，即归纳情形（§5.5）、这些方法的运行效率（§5.6）、GNCE 的消融研究（§5.7），以及 GNCE 在查询优化器中的应用（§5.8）。原始数据集、查询、结果和代码可在 GitHub[7] 上获取。

5.1 实验设置

5.1.1 数据集与查询。

对于数据集，我们使用 SWDF（Möller 等人 [2007], ）、LUBM（Guo 等人 [2005], ）、YAGO（Suchanek 等人 [2008], ）以及维基数据（Wang 等 [2021]）的一个子集，该子集包含 2000 万个三元组。

表2：数据集特征

KG	三元组	实体	谓词	类	类型化实体
SWDF	242256	76711	170	118	28%
LUBM	2688849	664048	18	15	66%
YAGO	58M	13M	92	189K	98%
维基数据	21M	5M	872	19K	84%

对于 LUBM 和 SWDF，我们使用与 Davitkova 等 [2022], 相同的数据集，以便更轻松地与文献中的先前结果进行比较。对于 YAGO，我们使用基准测试工具 *G-CARE* Park 等人[2020]提供的数据集。LUBM 是一个模拟大学领域的合成基准数据集。SWDF 是一个规模较小但密集的真实世界数据集，对作者、论文和会议进行建模。最后，YAGO 和维基数据是大型跨领域知识图谱。表2总结了这些数据集的统计信息。对于查询，我们生成了[8] 星形和路径查询，如.....中所述。

[7]https://github.com/DE-TUM/GNCE[8]
https://github.com/DE-TUM/rdf-subgraph-sampler

表 3。我们还在 YAGO 上生成了更多复杂的形状，并获取了维基数据的真实查询（详见第 5.4 节）。LUBM 和 SWDF 的查询基数范围为 $[e^0, e^6]$，YAGO 和维基数据的查询基数范围为 $[e^0, e^8]$。我们随机打乱所有查询，使用其中 80% 进行训练，并在剩余 20% 上评估所有方法。在此之前，我们从数据集中删除了重复查询（将变量名不同的同构查询视为重复项）。

表 3：用于训练和测试的查询

KG	星形查询				路径查询			
SWDF	总计：116645				总计：60739			
	2tp	3tp	5tp	8tp	2tp	3tp	5tp	8tp
	27843	29461	29690	29651	538	19033	550	40618
LUBM	总体：113855				总体：55336			
	2tp	3tp	5tp	8tp	2tp	3tp	4tp	8tp
	25356	29127	29727	29645	17784	33047	4505	0
YAGO	总体：91162				总体：87775			
	2tp	3tp	5tp	8tp	2tp	3tp	5tp	8tp
	13968	24416	24509	28269	19378	33807	17990	16600
维基数据	总体：109545				总计：101313			
	2tp	3tp	5tp	8tp	2tp	3tp	5tp	8tp
	29975	29936	29898	19736	29948	29951	29997	11417

5.1.2 对比方法。

我们将*GNCE* 与*G-CARE* Park 等人 [2020] 提供的实现进行比较，涵盖基于总结的方法（*SUMRDF* Stefanoni 等人 [2018] 和特征集（*CSET*）Neumann 和 Moerkotte [2011]）以及基于采样的方法（*impr* Chen 和 Lui [2017], *Wanderjoin* Li 等人 [2016], 和*jsub* Zhao 等人 [2018]），使用推荐的默认参数；详情请参阅 Park 等人 [2020] 的论文。我们还与最先进的基于学习的方法（*LMKG* Davitkova 等人 [2022] 和*LSS* Zhao 等人 [2021]）进行比较。对于基于采样的方法，我们报告其在 30 个样本上的平均性能（*G-CARE*中的默认设置）。对于基于学习的方法，我们针对每个数据集和查询类型训练一个模型，并按照作者的建议设置参数数量。下表总结了基于学习的方法的参数数量。请注意，*GNCE* 的参数数量最少，比*LSS*少 18 倍。参数更少通常意味着该方法的内存占用更小、执行时间更短。这也表明该模型不易过拟合。

模型参数数量

KG	*GNCE*（我们的方法）	*LMKG*	*LSS*
SWDF	$1.2 \cdot e^5$	$4.5 \cdot e^5$	$2.2 \cdot e^6$
LUBM	$1.2 \cdot e^5$	$2.9 \cdot e^5$	$2.2 \cdot e^6$
YAGO	$1.2 \cdot e^5$	$3.8 \cdot e^5$	$2.2 \cdot e^6$
维基数据	$1.2 \cdot e^5$	$3.1 \cdot e^5$	$2.2 \cdot e^6$

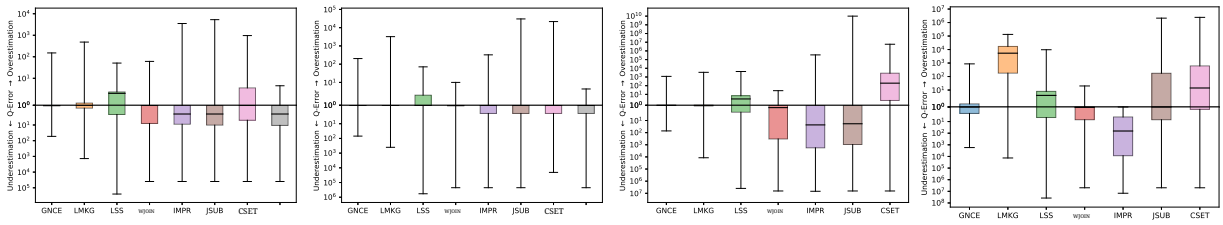
5.1.3 实现细节

GNCE 使用 Python 3.8 和 PyTorch Geometric 实现。计算 RDF2Vec 嵌入时，我们使用 Vandewiele 等人提供的实现。Vandewiele 等人 [2022]。我们对 RDF2Vec 模型进行 10 个训练轮数。对于每个实体，我们采样 5 条最大深度为 4 的随机游走，并按照第 2.2 节所述学习嵌入。嵌入的维度设为 100。最后，我们存储嵌入，供*GNCE*后续使用。除*LSS* 在 YAGO 上的实验外，所有实验均在配备 16 GB 随机存取存储器、Intel Core i7-11800H@2.30GHz 和 NVIDIA GeForce RTX 3050 GPU 的计算机上进行。由于*LSS* 需要大量随机存取存储器，因为它会在训练时将整个知识图谱加载到内存中，因此该模型是在配备 512 GB 随机存取存储器、Intel Xeon Silver 4314 CPU@2.40GHz 和 Nvidia A100 GPU 的计算机上训练的。

5.1.4 评估指标

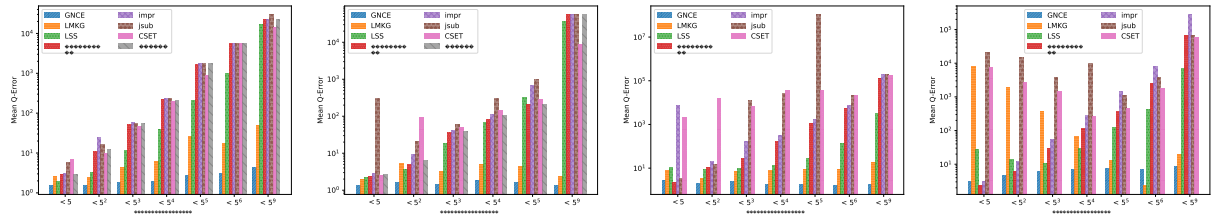
与以往的工作类似，Zhao 等人 [2021], 我们根据 q-误差评估这些方法的有效性：

$$\text{q-Error}(Q) = \max\left(\frac{||Q||_{\mathcal{G}}}{\widehat{||Q||}_{\mathcal{G}}}, \frac{\widehat{||Q||}_{\mathcal{G}}}{||Q||_{\mathcal{G}}}\right) \tag{6}$$



(a) SWDF (b) LUBM (c) YAGO (d) 维基数据

图 3：星形查询的 q-误差箱线图



(a) SWDF (b) LUBM (c) YAGO (d) 维基数据

图4：按查询的真实基数分组的星形查询 q-误差

表4：星形查询的平均 q-误差

	方法	SWDF	LUBM	YAGO	维基数据
采样	*Wanderjoin*	129.00	232.00	18713.00	34564.00
	impr	141.00	237.00	2605.00	169241.00
	jsub	169.00	409.00	$15 \cdot e^6$	40055.00
总结	*CSET*	94.00	904.00	41586.00	37310.00
	SUMRDF	130.00	233.00	-	-
学习型	*LMKG*	3.00	2.00	9.00	10267.00
	LSS	1187.00	283.00	502.00	4089.00
	GNCE	**1.42**	**1.22**	**1.96**	**4.20**

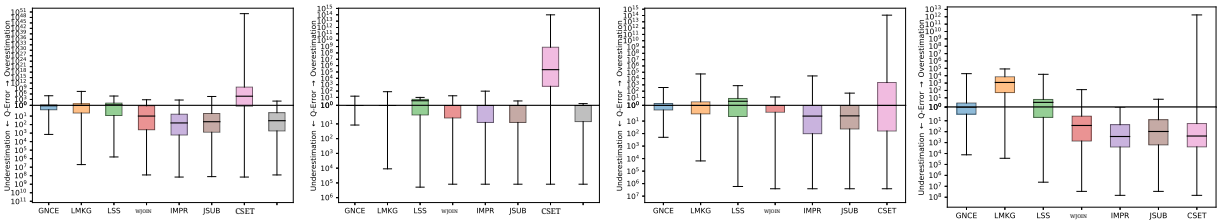
5.2 星形查询的基数估计

在本节中，我们研究这些方法对星形查询的有效性，其中主语位置是连接变量。

在表4中，我们报告了各数据集上所有方法的平均 q-误差。[9] 总体而言，我们观察到，随着知识图谱规模增大，q-误差也随之增加。
这一现象在意料之中，因为在较大的数据集中捕捉相关性更具挑战性，而且这类数据集中自然会更频繁地出现基数较大的查询。这些因素会对估计结果产生负面影响。采样方法和摘要方法在一定程度上表现相似，但*jsub* 除外，它在 YAGO 上产生了较大的误差。*CSET* 在测试的星形查询上表现不佳，因为这些查询可能包含绑定对象，从而导致*CSET*出现高估。

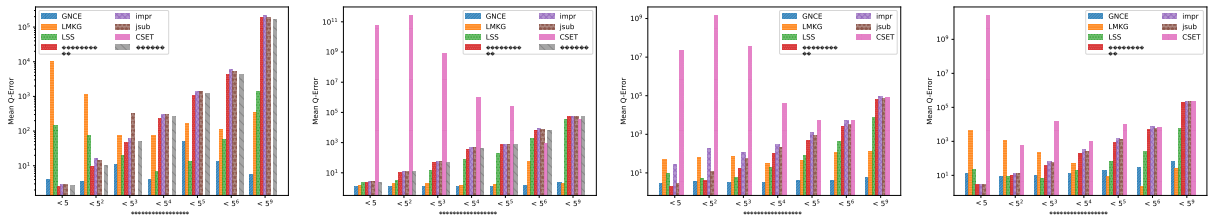
在学习型方法中，*GNCE* 整体表现最佳。有趣的是，*LSS* 在这些知识图谱中的表现并不好。我们推测，这是因为基于 *LSS*类所学习到的节点表示，无法区分形状相似但基数不同的查询。*LSS* 的这一表现，在 SWDF 中尤为明显。其他所有方法都在 SWDF（规模最小的数据集）上取得了最佳表现，而*LSS* 却产生了较大的 q-误差，因为类型化实体的数量相当少（仅占 22%，参见表2）。对于无类型实体，*LSS* 的实体表示默认为全零通用向量，不包含任何信息。相比之下，*GNCE* 不对实体类型作任何假设，仍可通过知识图谱嵌入为实体编码出有意义的表示。

[9]由于针对该图的摘要构建在 12 小时后仍未完成，因此无法报告*SUMRDF* 在 YAGO 和维基数据上的结果。



(a) SWDF (b) LUBM (c) YAGO (d) 维基数据

图 5：路径查询的 q-误差箱线图



(a) SWDF (b) LUBM (c) YAGO (d) 维基数据

图 6：按查询的真实基数分组的路径查询 q-误差

为进一步了解这些方法的表现，我们分析了 q-误差的分布。图3以箱线图展示了 q-误差的分布；低估和高估分别以低于和高于水平线（q-误差为 1，即理想值）的数值表示。我们首先考察采样方法（*Wanderjoin*、*impr*和*jsub*），这些方法往往会低估基数，尤其是*impr*。Park 等人此前也在其他数据集中观察到了这一现象。Park 等人[2020]基于总结的方法也往往会产生低估，YAGO 和维基数据上的*CSET*除外。进一步检查结果后发现，*CSET* 会大幅高估带有绑定对象的高选择性查询的基数。对于基于学习的方法，*GNCE* 和*LMKG*的 q-误差集中在最优值 1 附近。*LSS* 往往会对相当一部分查询略微高估，但也会出现严重低估。与*LSS* 和*LMKG*相比，*GNCE*的最大低估和高估幅度以及分位数，围绕原点的分布更加对称，这表明它在所有数据集上的表现更为稳定。

接下来，我们分析不同基数大小下各方法的平均 q-误差（参见图 4）。结果表明，随着查询基数增大，q-误差往往也会增大，但 *GNCE* 和*LMKG*除外。对于结果大小范围 < 5内、选择性很高的查询，非学习型方法表现良好。这类查询通常包含许多绑定原子，因此*impr* 和*Wanderjoin* 等基于采样的方法可能收集到了（近乎）精确的统计信息。唯一的例外是 YAGO 中的*CSET*；不过，正如前文所解释的，这些结果是由*CSET* 产生的高估所导致的大 q-误差造成的。对于基于学习的方法，高基数范围内训练查询数量较少，可能导致较大的误差。尽管如此，*GNCE*凭借针对查询图结构特性设计的归纳偏置，以及由嵌入带来的更具信息量的特征表示，只需较少的训练数据就能做出准确预测。因此，在所有结果大小范围内，*GNCE* 的 q-误差始终最低；唯一例外是 YAGO 和维基数据中结果大小范围为 < 5 的情况，此时其表现略逊于*Wanderjoin*。

5.3 路径查询的基数估计

接下来，我们考察这些方法对路径查询的有效性，其中连接发生在三元模式的主语和宾语之间。表5给出了平均 q-误差。与星形查询相比，大多数方法的 q-误差更高（参见表4）。这与 Park 等人此前报告的结果一致。Park 等人 [2020]。此外，有趣的是，与其他数据集相比，所有方法在 SWDF 上的表现至少差一个数量级。这可能是由于 SWDF 图的密度较高，即从每个起点出发都有大量可能的路径，从而使基数估计更容易产生显著误差。数据集中的（训练）查询较少也影响了基于学习的方法的有效性（参见表3）。[10] 总体而言，学习型方法表现最佳

[10]请注意，这并不是查询生成器的问题，而是因为知识图谱通常具有较小的直径，Zloch 等人[2019],，这使得不同路径查询的数量远小于星形查询的数量。

表 5：路径查询的平均 q-误差

	方法	SWDF	LUBM	YAGO	维基数据
采样	*Wanderjoin*	$22 \cdot e^4$	389.00	9730.00	67001.00
	impr	$73.7 \cdot e^4$	429.00	1722.00	146623.00
	jsub	$37.4 \cdot e^4$	414.00	10952.00	69822.00
总结	*CSET*	$1 \cdot e^{47}$	$63 \cdot e^{10}$	$2 \cdot e^{29}$	$9 \cdot e^{30}$
	SUMRDF	$2 \cdot e^5$	$393 \cdot e^6$	-	-
学习式	*LMKG*	1820.00	2.00	69.00	4220.00
	LSS	1187.00	179.00	1075.00	1204.00
	GNCE	**12.50**	**1.14**	**3.78**	**14.00**

表 6：复杂查询和用户查询的平均 q-误差

查询	*Wanderjoin*	*impr*	*jsub*	*CSET*	*SUMRDF*	*LMKG*	*LSS*	*GNCE*
User	3312	21984	3426	$46 \cdot e^{11}$	-	268	198	**13**
复杂	226282	508309	227163	$3 \cdot e^{27}$	-	113	2519	**8.4**

，其次是基于采样的方法，最后是基于摘要的方法。不过，*GNCE* 在路径查询上的表现仍比当前最先进水平高出数个数量级。

接下来，我们分析图 5 中 q-误差的分布。正如前文所述，由于 SWDF 数据集的密度较高，所有采样方法都倾向于低估查询基数。此外，与 LUBM 相比，由于 YAGO 和维基数据规模较大且这些知识图谱已知具有半结构化特性，它们的表现显著变差。*CSET* 计算出的估计值高得不切实际，这一现象 Davitkova 等早已报告过。Davitkova 等 [2022]。这些结果证实，正如预期，*CSET* 摘要在捕捉主语与宾语之间的连接相关性方面表现不佳。在基于学习的方法中，*LSS*倾向于产生高估。这一现象在星形查询中也曾观察到（参见第 5.2 节），表明这是*LSS*的一种持续性表现。相比之下，*GNCE* 和*LMKG* 都倾向于低估，唯独*LMKG* 在维基数据上并非如此。尽管如此，须线所示的误差分布范围中，*GNCE* 仍比*LMKG*更小。

在图 6 中，我们按查询基数对 q-误差进行分类分析。由于数值极高，图 6(a) 省略了 *CSET* 的柱状图。与星形查询中观察到的趋势类似，我们发现，除*GNCE*外，所有方法的 q-误差都会随着查询基数的增加而增大；而*GNCE*的平均 q-误差基本保持一致。其次，在 SWDF、YAGO 和维基数据中，在较小的基数范围 <5内，基于采样的方法略优于*GNCE* 。有趣的是，在 SWDF 上，对于基数介于 5^4 和 5^5, 之间的查询，*LSS* 在这一特定情况下的表现优于所有其他方法。最后，在维基数据上，*GNCE* 在较小的基数范围内优于*LMKG* ，而在较大的基数范围内，*LMKG*表现更好。

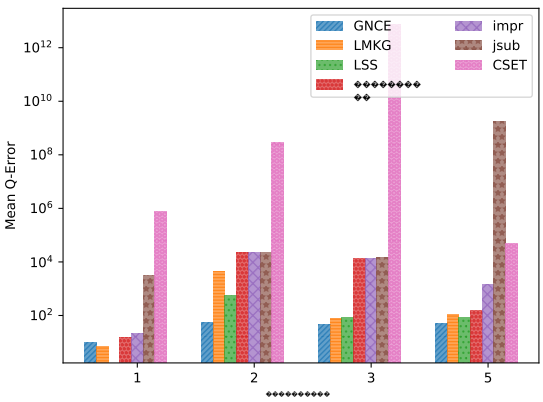
5.4 用户和复杂查询的基数估计

为了分析这些方法在合成星形查询和路径查询之外的有效性，我们还研究了更具挑战性的查询类型，即真实的用户查询和更复杂的查询形状。

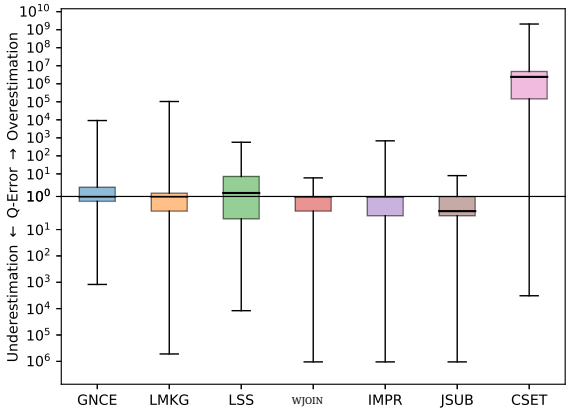
真实用户查询取自维基数据 SPARQL 查询服务日志[11]。我们筛选出在所用的 Wikidata 子集上能够生成答案的合取查询，共得到 11,832 个查询，其中 90% 只有一个三元模式。需要指出的是，这一比例较高，可能是由于筛选条件要求查询为合取查询，且必须能在该子集上生成答案；其他研究报告的比例则较低，为 ∼77% Bonifati 等人 [2020]。基于学习的方法首先在合成星形和路径查询上训练，然后使用 80% 的用户查询进行微调。所有方法均在剩余 20% 的查询上进行评估。*LSS* 只能在包含多个三元模式的查询上进行评估，因为所提供的代码不支持单三元模式查询。表 6 展示了平均 q-误差，其中 *GNCE* 的表现优于所有其他方法。图 7(a) 按查询规模分组展示了平均 q-误差。*LMKG* 在单三元模式的情况下表现最佳，而*GNCE* 在其余范围内表现最佳。相应的箱线图（图 7(b)）显示了与合成查询相似的结果。

对于更复杂的形状查询，我们在 YAGO 数据集上生成了花朵查询和雪花形查询。花朵由一个路径查询组成，且一端连接一个星形结构。雪花形查询是一条路径，两端都连接着星形结构。我们总共生成了 72,742 个包含 2 至 7 个三元组模式的复杂查询。表 6 展示了平均 q-误差。与 YAGO 数据集上的星形查询和路径查询相比，所有方法的表现都更差，尽管查询规模的分布相近。这表明，查询结构的复杂程度会直接影响基数估计器的性能。此外，就平均 q-误差而言，*GNCE*再次优于其他方法。图 7 (c) 和 (d) 详细展示了误差分布，

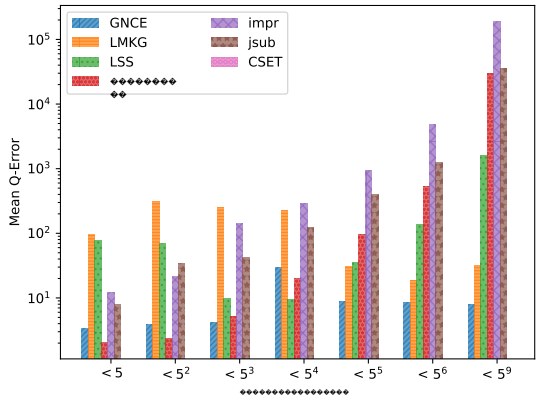
[11]https://iccl. inf.tu-dresden.de/web/维基数据_SPARQL_日志 s/en



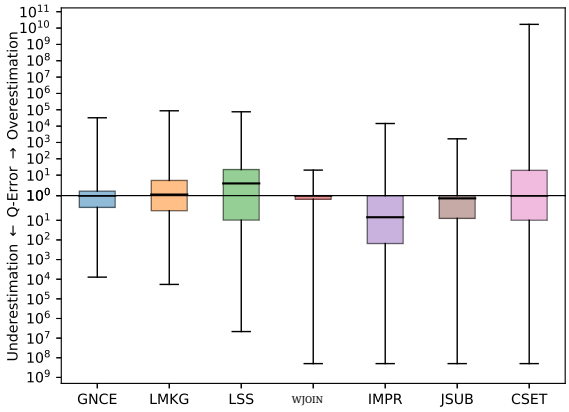
(a) 维基数据用户查询



(b) 维基数据用户查询



(c) YAGO 复杂查询



(d) YAGO 复杂查询

图 7：YAGO 复杂查询和维基数据用户查询的结果

再次表明 *Wanderjoin* 在低基数情况下表现良好，而 *GNCE* 在高基数情况下占据优势。我们还观察到，*Wanderjoin* 的箱线图在完美预测值周围的箱体最窄，也就是说，它能为大量查询生成高质量预测，但同时伴有较大的离群值（如须线所示）。这与 Park 等人的结果一致。Park 等人 [2020]。

5.5 归纳情形

现在，我们测试这些方法在归纳情形下的有效性。在这种情形中，测试查询仅包含训练期间未见过的实体。请注意，这是归纳性的最坏情形，因为通常有些实体可能曾经出现过。归纳设置对生产环境至关重要，因为这意味着模型可以使用包含知识图谱实体（真）子集的查询样本进行训练。相反，模型需要使用涉及知识图谱中所有实体的查询样本，这使得部署此类模型不太实际。

我们针对 SWDF 和 YAGO 的星形查询进行了这项实验，具体步骤如下。对于每个数据集，查询集被划分为两个互不相交的子集（训练集和测试集），确保两个子集中的实体互不重叠；也就是说，一个子集中的任何查询都不涉及另一个子集中出现的实体（不过，谓词是已知的，这符合归纳性研究中的常见设置）。此外，在为 *G-CARE* 中基于总结和基于采样的方法构建索引结构之前，我们从输入知识图谱中移除了所有涉及这些实体的三元组。对于 *GNCE* 和*LSS*，测试集中的实体没有提供嵌入或出现次数作为特征化信息，而是使用随机（但固定）的向量来表示这些实体。这些步骤确保了互不相交的测试集中的实体从未被任何方法见过，包括统计量/采样构建器、嵌入生成器或神经网络。

表 7：星形查询完整归纳情形的平均 q-误差

数据集：SWDF							
$\|\|Q\|\|_{\mathcal{G}}$	*GNCE*	*LMKG*	*LSS*	*Wanderjoin*	*impr*	*jsub*	*CSET*
< 5	5.23	3.06	**1.74**	2.82	2.82	2.82	9.73
$< 5^2$	3.16	**2.88**	4.26	12.51	12.51	12.51	11.68
$< 5^3$	**3.26**	7.16	15.57	54.81	54.81	54.81	37.15
$< 5^4$	**4.16**	29.92	89.02	264.14	274.87	284.91	176.37
$< 5^5$	**7.98**	159.26	596.94	1008.72	1010.95	1011.22	538.91
$< 5^6$	**11.23**	575.22	3567.94	1649.89	1655.85	1657.48	779.65
$< 5^9$	**335.51**	840.27	65214.02	14198.33	14204.75	14198.37	-
Avg	**52.93**	231.11	9927.07	2455.89	2459.51	2460.30	2250.30
数据集：YAGO							
$\|\|Q\|\|_{\mathcal{G}}$	*GNCE*	*LMKG*	*LSS*	*Wanderjoin*	*impr*	*jsub*	*CSET*
< 5	189.48	319.94	8.58	**2.51**	**2.51**	**2.51**	12133.80
$< 5^2$	229.71	138.35	**3.94**	11.60	13.55	13.55	13244.83
$< 5^3$	73.98	47.19	**8.10**	33.96	50.27	42.74	25952.18
$< 5^4$	**38.18**	145.65	41.76	185.62	262.63	218.33	13092.83
$< 5^5$	**44.89**	595.18	196.23	628.17	1036.19	1012.65	64086.77
$< 5^6$	**86.65**	2585.38	917.44	4249.70	6114.76	4277.01	119876.0
$< 5^9$	**280.94**	49462.4	19016.3	141838.6	171508.1	142063.0	145802.6
Avg	**134.83**	7613.44	2884.61	20992.88	25569.72	21089.98	56312.71

随后，三个学习型方法 *GNCE, LMKG, LSS* 在训练集上进行训练（为*GNCE* 和*LSS*提供嵌入），并在测试集上进行评估（不提供嵌入和出现次数）。*G-CARE* 中的方法则使用基于排除测试集实体后的知识图谱构建的统计信息，在测试集上进行评估。

平均 q-误差如表 7 所示。结果表明，对于特定的查询规模，*GNCE*、*LMKG*、*LSS* 或*Wanderjoin* 表现最佳。与图 4(a)中的非归纳情形相比，所有方法的性能都有所下降，这符合预期，因为缺少有关实体的统计信息。总体而言，*GNCE* 表现最佳。

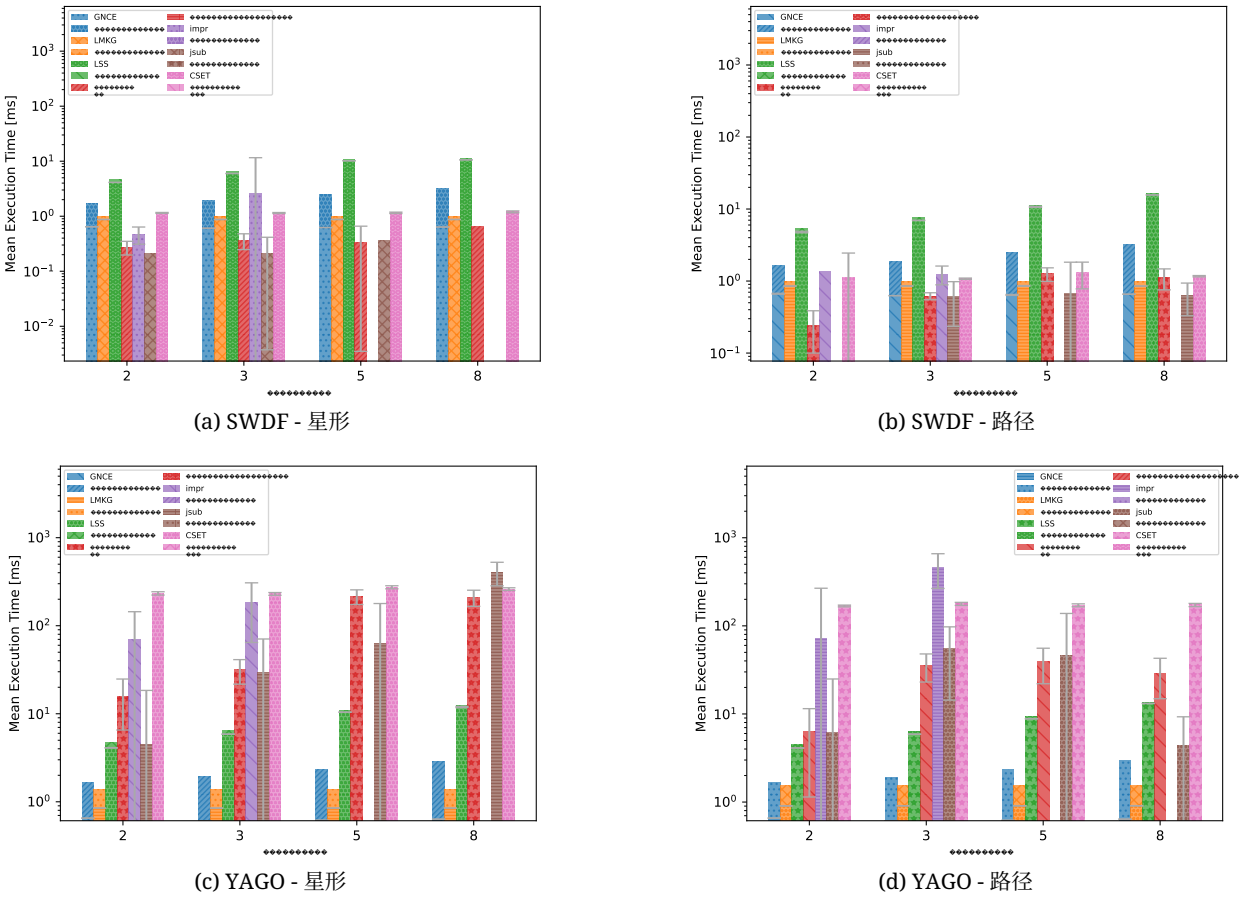
总之，这些结果表明，*GNCE* 可用于涉及新实体的归纳情形，无需频繁重新训练。

5.6 运行效率

5.6.1 预测时间（在线）。

我们研究了来自最小和最大数据集（即 SWDF 和 YAGO）的不同查询的基数预测运行时间。为测量运行时间，我们使用 Python 中的 *time* 库 Foundation [2021] 。对于学习方法，我们测量：1）收到查询后，从磁盘加载嵌入并创建查询特征化所需的时间；2）查询表示经过网络前向传播的时间，由此生成最终的基数估计。对于 *G-CARE* Park 等人 [2020], 提供的方法，我们使用各方法提供的计时功能，测量从收到查询到生成基数估计的总时间。对于基于采样的方法，按照*G-CARE* 的默认设置执行 30 次采样，并采用其总运行时间。对于所有方法，我们排除基数预测失败的情况，即基数估计为≤ 0的情况。

图 8 按查询中的三元组模式数量（查询大小）对结果进行了分组。灰色柱表示 95% 置信区间。比较 SWDF 和 YAGO 的结果后，我们发现，非学习型方法的运行时间会受到数据集大小和查询类型的影响。基于样本的方法，即 *impr*、*jsub* 和 *Wanderjoin*，仅在规模较小的 SWDF 数据集上表现出更高的效率。*CSET* 的性能与基于样本的方法相近，因为它们都需要多次访问（大型）摘要来进行预测。值得注意的是，对于查询大小为 5 和 8 的情况，我们没有报告 *impr* 的运行时间，因为这些规模下的所有基数估计均失败。学习型方法的模型预测时间在不同数据集、查询规模和类型下表现稳定。无论从均值还是狭窄的置信区间来看，都可以观察到这一点，后者表明运行时间几乎没有变化。*GNCE* 和 *LSS* 的加载时间会随查询大小的增加而延长。这是预期之中的，因为需要从磁盘加载更多嵌入。我们发现，*GNCE* 的图神经网络运行时间在学习型方法中表现最佳，并且优于基于采样/统计的方法。我们还注意到，*GNCE* 的总运行时间大部分用于加载嵌入和查询图特征化，但总体上仍能在 1–2 毫秒内完成。采用高效存储和编译代码可以显著缩短 *GNCE* 的运行时间。总体而言，*LMKG*



(a) SWDF - 星形

(b) SWDF - 路径

(c) YAGO - 星形

(d) YAGO - 路径

图 8：不同方法的平均预测时间 [毫秒]，按查询中的三元组模式数量分组。置信区间（95%）以灰色条显示

在各方法中，LMKG 的运行时间表现最佳。在较大的数据集上，学习型方法的运行时间比基于摘要/统计的方法快几个数量级。将运行时间控制在个位数毫秒范围内是理想的，因为极快的运行速度对于基数估计器至关重要，例如，它必须能够在查询优化器中运行。

5.6.2 模型/统计信息构建时间（离线）。

所有研究的方法在用作基数估计器之前，都需要构建/训练模型，或构建统计信息/样本。因此，我们现在测量完成这些准备工作所需的时间。我们使用 YAGO 进行评估，但为了便于在不同方法及其他图之间进行插值，报告每个原子的构建/训练时间（下文简称训练时间）。对于学习型方法，批量大小设为 1，总训练轮数为 20。

表8：每个原子的嵌入和训练时间（毫秒）

	GNCE	*LMKG*	*LSS*	*Wanderjoin*	*impr*	*jsub*	*CSET*
嵌入	6.40	-	3.94	-	-	-	-
训练	2.52	1.16	114.03	0.00019	0.00020	0.00019	0.00022

表8展示了本研究的结果。比较测试用例的运行时间和训练时间的不同数值可以发现，基于采样/摘要的方法的训练时间明显短于基于学习的方法，而基于学习的方法的测试时间更短（在较大的图上）。对此，一个直观的解释是，在神经网络训练期间，关于知识图谱的大量统计信息和模式被压缩到嵌入和神经网络参数中，这一过程耗时较长，却能加快执行时间。基于采样/摘要的方法预先计算的统计信息较少，但在预测期间需要进行更多计算，尤其是在知识图谱

表 9：GNCE 在 YAGO 上的消融研究结果

方法	星形查询	路径查询
基线	1.96	3.78
-RDF2Vec	2.40 (↓ 22%)	6.14 (↓ 62%)
-有向消息传递	1.93 (↑ 1%)	3.87 (↓ 3%)

较大时。使用前，学习型方法较长的训练时间可以接受，但执行时间必须非常快（尤其是在查询优化的情况下）。我们还想指出，*GNCE* 耗时显著最长，这是因为嵌入计算是在一台相对较小的机器上完成的。在嵌入学习期间，大部分时间都花在随机游走生成上；如果硬件条件允许，这一过程可以并行化。为了说明这一点，使用本文所用的硬件，为 700 万个实体（DBpedia 的近似规模，Lehmann 等人，[2015]）计算嵌入需要 ∼12 小时。随后，使用 100,000 个平均包含 5 个三元组模式的查询训练 *GNCE* 图神经网络需要 ∼ 63 分钟。

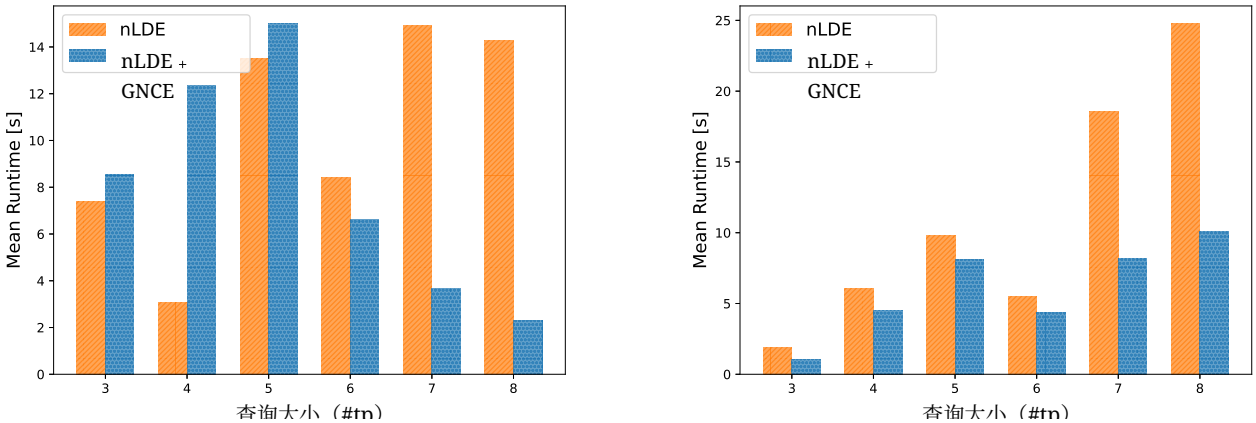
5.7 消融研究

为了确定*GNCE*不同组成部分的影响，我们进行了消融研究。为此，我们将基线模型（*GNCE*）与同一模型进行比较；该基线模型用于前文各节，而同一模型的变化包括：（1）用二进制编码替换*RDF2Vec*嵌入；（2）将有向 TPN 消息传递简化为无向消息传递，不再区分入边和出边。对于（1），我们为知识图谱中的所有实体编号并分配 ID，然后将实体对应的嵌入替换为 ID 的二进制表示，并在末尾补 0，使其长度达到 100。考虑到篇幅，本研究仅在最大的知识图谱 YAGO 上进行。

表 9 显示，从查询图中移除RDF2Vec嵌入会降低（↓）在星形查询和路径查询上的性能。不过，对路径查询的影响要大得多。这可以用以下事实来解释：星形查询的谓词本身已能很好地预测查询基数（类似于特征集）。出于同样的原因，移除有向消息传递也只会降低路径查询上的性能。

5.8 GNCE 在查询优化中的应用

如前所述，基数估计的总体目标是为查询优化器提供估算结果，以尽量缩短查询运行时间。尽管查询优化是一个与本研究所解决的问题正交的问题，我们
展示 *GNCE* 辅助实际查询优化器的初步结果。为此，我们选择了 nLDE Acosta 和 Vidal [2015], , 这是一个同样使用 Python 实现的开源 SPARQL 引擎。nLDE 使用子查询的基数估计值，在通过贪心启发式遍历查询计划空间时选择物理连接算子。我们比较了使用 nLDE 基数估计器和 *GNCE* 制定的查询计划。



（a）星形查询　　（b）路径查询

图9：SWDF 上的查询优化结果

在本研究中，我们扩展了 SWDF 查询集，纳入包含 3–8 个三元组模式的查询。我们测试了 243 个星形查询和 258 个路径查询。其中，使用

GNCE 作为估计器时，15% 的星形查询和 14% 的路径查询采用了不同的计划。正如预期的那样，在两种基数估计器下计划相同的查询，其运行时间也相近。图9 展示了生成不同计划的星形查询和路径查询的运行时间。总体而言，查询大小越大，使用*GNCE*执行查询的速度就越快。对于较小的星形查询，*GNCE* 并未缩短执行时间。对于较大的查询（包含 7 和 8 个三元组模式），执行时间缩短了 100% 以上。如图 8(a) 所示，对于一个包含 8 个三元组模式（即 7 个连接）的查询，*GNCE* 在优化过程中生成基数预测需要 14 毫秒。这约占总执行时间的 0.1%。

6 结论与展望

我们提出了*GNCE*，这是一种基于知识图谱嵌入（KGE）和图神经网络（GNN），用于准确估计知识图谱（KG）上合取查询基数的方法。KGE 为 GNN 提供了具有语义意义的特征，而 GNN 可直接预测经嵌入增强的查询图的基数。

我们的实验*GNCE* 表明，*GNCE* 平均而言，在合成及真实世界知识图谱和查询上的 q-误差方面优于当前最先进的方法。*GNCE* 也具有鲁棒性，不会产生严重的预测误差。*GNCE*对未见实体也有很好的泛化能力，因此适合实际部署。在此背景下，我们还表明，该模型轻量高效，预测运行时间也低于 1 毫秒。

未来的工作中，我们计划将带有 FILTER过滤器 的查询纳入研究。这需要扩展查询采样和 KGE 随机游走，以便更好地捕捉底层范围分布；还需要扩展查询特征化，以表示过滤操作和值。此外，我们将研究如何针对数值和文本字面量进行专门处理，并为其提供更有意义的嵌入。另一个有前景的方向是采用主动学习，根据表现不佳的查询制定自适应采样策略，以加快图神经网络训练。最后，我们还将研究专为基数估计定制的 KGE 方法。

7 致谢

作者谨向 Joscha Klimpel 和 Hanno Gottschalk 教授致谢，感谢他们在论文准备过程中提供的宝贵意见。

参考文献

Richard Cyganiak、David Wood 和 Markus Lanthaler。RDF 1.1 概念和抽象语法。W3C 建议书，W3C，2014年2月。https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/。Natasha Noy、Yuqing Gao、Anshu Jain、Anant Narayanan、Alan Patterson 和 Jamie Taylor。产业规模知识图谱：经验与挑战。*ACM* 通讯，62(8):36–43，2019年7月。ISSN 0001-0782。doi:10.1145/3331166。URL https://doi.org/10.1145/3331166。Eric Prud'hommeaux 和 Andy Seaborne。面向 RDF 的 SPARQL 查询语言。W3C 建议书，W3C，2008年1月。https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/。Thomas Neumann 和 Guido Moerkotte。特征集：针对多个连接的 RDF 查询进行精确基数估计。载于 Serge Abiteboul、Klemens Böhm、Christoph Koch 和 Kian-Lee Tan 编，第*27*届国际数据工程会议（*ICDE 2011*）论文集，*2011*年*4*月*11*日至*16*日，德国汉诺威，第984–994页。IEEE计算机学会，2011年。doi:10.1109/ICDE.2011.5767868。URL https://doi.org/10.1109/ ICDE.2011.5767868。Giorgio Stefanoni、Boris Motik 和 Egor V. Kostylev。使用图摘要估计 RDF 数据上合取查询的基数。第1043–1052页。计算机协会，2018年4月。ISBN 9781450356398。doi:10.1145/3178876.3186003。Angjela Davitkova、Damjan Gjurovski 和 Sebastian Michel。LMKG：用于知识图谱基数估计的学习模型。2022年。doi:10.48786/edbt.2022.07。URL https://www.w3.org/TR/sparql11-overview/。Kangfei Zhao、Jeffrey Xu Yu、Hao Zhang、Qiyan Li 和 Yu Rong。用于子图计数的学习式草图。第2142–2155页。计算机协会，2021年。doi:10.1145/3448016.3457289。Keyulu Xu、Weihua Hu、Jure Leskovec 和 Stefanie Jegelka。图神经网络的能力有多强？载于第*7*届国际学习表征会议（*ICLR 2019*），美国路易斯安那州新奥尔良，*2019*年*5*月*6*日至*9*日。OpenReview.net，2019年。URL https://openreview.net/forum?id=ryGs6iA5Km。Andrea Rossi、Denilson Barbosa、Donatella Firmani、Antonio Matinata 和 Paolo Merialdo。用于链接预测的知识图谱嵌入：比较分析。*ACM*数据知识发现汇刊，15，2021年。ISSN 1556472X。doi:10.1145/3424672。

Aidan Hogan、Eva Blomqvist、Michael Cochez、Claudia d'Amato、Gerard de Melo、Claudio Gutierrez、Sabrina Kirrane、José Emilio Labra Gayo、Roberto Navigli、Sebastian Neumaier、Axel-Cyrille Ngonga Ngomo、Axel Polleres、Sabbir M. Rashid、Anisa Rula、Lukas Schmelzeisen、Juan F. Sequeda、Steffen Staab 和 Antoine Zimmermann。知识图谱。*ACM*计算调查，54(4):71:1–71:37，2022年。doi:10.1145/3447772。URL https://doi.org/ 10.1145/3447772。Petar Ristoski 和 Heiko Paulheim。RDF2Vec：用于数据挖掘的 RDF 图嵌入。载于国际语义网会议，第498–514页。Springer，2016。Tobias Weller 和 Maribel Acosta。使用随机神经网络预测知识图谱中的实例类型断言。载于 Gianluca Demartini、Guido Zuccon、J. Shane Culpepper、Zi Huang 和 Hanghang Tong 编，*CIKM '21*：第*30*届*ACM*信息与知识管理国际会议，线上活动，澳大利亚昆士兰，*2021*年*11*月*1*日至*5*日，第2111–2118页。ACM，2021年。doi:10.1145/3459637.3482377。URL https://doi.org/10.1145/3459637.3482377。Tomás Mikolov、Kai Chen、Greg Corrado 和 Jeffrey Dean。在向量空间中高效估计词表示。载于 Yoshua Bengio 和 Yann LeCun 编，第*1*届国际学习表征会议（*ICLR 2013*）论文集，研讨会专题，美国亚利桑那州斯科茨代尔，*2013*年*5*月*2*日至*4*日，2013年。URL http://arxiv.org/ abs/1301.3781。Michael M. Bronstein、Joan Bruna、Taco Cohen 和 Petar Veli˘ckovi´c。几何深度学习：网格、群、图、测地线和规范。2021年4月。URL http://arxiv.org/abs/2104.13478。PyG团队。创建消息传递网络。https://pytorch-geometric.readthedocs.io/en/latest/ notes/create_gnn.html，2022年。访问日期：2022年11月29日。Zonghan Wu、Shirui Pan、Fengwen Chen、Guodong Long、Chengqi Zhang 和 Philip S. Yu。图神经网络综合综述。2019年1月。doi:10.1109/TNNLS.2020.2978386。URL http://arxiv.org/abs/1901.00596http://dx.doi.org/10.1109/TNNLS.2020.2978386。Kyle Deeds、Dan Suciu、Magda Balazinska 和 Walter Cai。用于连接基数估计的度序列界。载于 Floris Geerts 和 Brecht Vandevoort 编，第*26*届国际数据库理论会议（*ICDT 2023*），*2023*年*3*月*28*日至*31*日，希腊约阿尼纳，第255卷 LIPIcs，第8:1–8:18页。Schloss Dagstuhl - Leibniz信息学中心，2023年。doi:10.4230/LIPIcs.ICDT.2023.8。URL https://doi.org/10.4230/LIPIcs.ICDT.2023.8。Graham Cormode、Minos Garofalakis、Peter J. Haas 和 Chris Jermaine。海量数据摘要：样本、直方图、小波、草图。*Foundations andTrends in Databases*，4:1–294，2011年。ISSN 19317883。doi:10.1561/1900000004。David Vengerov、Andre Cavalheiro Menck、Mohamed Zait 和 Sunil P Chakkappen。带筛选条件的连接大小估计，2150。Feifei Li、Bin Wu、Ke Yi 和 Zhuoyue Zhao。Wander join：通过随机游走实现在线聚合。第26-June-2016卷，第615–629页。计算机协会，2016年6月。ISBN 9781450335317。doi:10.1145/2882903.2915235。Zhuoyue Zhao、Robert Christensen、Feifei Li、Xiao Hu 和 Ke Yi。重新审视连接上的随机抽样。载于*2018*年国际数据管理会议论文集，第1525–1539页，2018年。Yeonsu Park、Seongyun Ko、Sourav S. Bhowmick、Kyoungmin Kim、Kijae Hong 和 Wook Shin Han。G-care：用于子图匹配基数估计技术性能基准测试的框架。第1099–1114页。计算机协会，2020年6月。ISBN 9781450367356。doi:10.1145/3318464.3389702。Ji Sun、Jintao Zhang、Zhaoyan Sun、Guoliang Li 和 Nan Tang。学习式基数估计：设计空间探索与比较评估。第15卷，第85–97页。VLDB基金会，2021年。doi:10.14778/3485450.3485459。Lise Getoor、Ben Taskar 和 Daphne Koller。使用概率模型进行选择率估计，2001。Kostas Tzoumas、Amol Deshpande 和 Christian S. Jensen。无需独立性假设的轻量级图模型选择率估计。*VLDB*基金会会刊，4:852–863，2011年8月。ISSN 21508097。doi:10.14778/3402707.3402724。Max Halford、Philippe Saint-Pierre 和 Frank Morvan。一种基于贝叶斯网络的查询选择率估计方法。2019年7月。doi:10.1007/978-3-030-18579-4_1。URL http://arxiv.org/abs/1907.06295http://dx. doi.org/10.1007/978-3-030-18579-4_1。Benjamin Hilprecht、Andreas Schmidt、Moritz Kulessa、Alejandro Molina、Kristian Kersting 和 Carsten Bin-nig。Deepdb：从数据学习，而非从查询学习！第13卷，第992–1005页。VLDB基金会，2020年。doi:10.14778/3384345.3384349。

Andreas Kipf、Thomas Kipf、Bernhard Radke、Viktor Leis、Peter A. Boncz 和 Alfons Kemper。学习得到的基数：利用深度学习估计相关连接。发表于第*9*届创新数据系统研究双年会议，*CIDR 2019*，美国加利福尼亚州阿西洛马，*2019*年*1*月*13*日至*16*日，在线论文集。www.cidrdb.org，2019年。URL http: // cidrdb.org/cidr2019/papers/p101-kipf-cidr19.pdf。Lucas Woltmann、Claudio Hartmann、Maik Thiele、Dirk Habich 和 Wolfgang Lehner。利用局部深度学习模型进行基数估计。计算机协会，2019年7月。ISBN 9781450368025。doi:10.1145/3329859.3329875。Kangfei Zhao、Jeffrey Xu Yu、Zongyan He、Rui Li 和 Hao Zhang。通过神经网络高斯过程实现轻量级且准确的基数估计。发表于*2022*年国际数据管理会议论文集，第973–987页，2022年。Markus Stocker、Andy Seaborne、Abraham Bernstein、Christoph Kiefer 和 Dave Reynolds。使用选择率估计进行 SPARQL 基本图模式优化。第595–604页，2008年。ISBN 9781605580852。doi:10.1145/1367497.1367578。E Patrick Shironoshita、Michael T Ryan 和 Mansur R Kabuka。用于优化本体查询的基数估计。*ACM SIGMOD Record*，36(2):13–18，2007年。Thomas Neumann 和 Gerhard Weikum。用于可扩展管理 RDF数据的 RDF-3X 引擎，2009年。Xiaowei Chen 和 John C.S. Lui。在线社交网络中的图元计数挖掘。第71–80页。电气电子工程师学会，2017年1月。ISBN 9781509054725。doi:10.1109/ICDM.2016.99。Hai Huang 和 Chengfei Liu。连接的RDF三元组模式的选择率估计。第1435–1444页，2011年。ISBN 9781450307178。doi:10.1145/2063576.2063784。Jie Zhang、Yuxiao Dong、Yan Wang、Jie Tang 和 Ming Ding。ProNE：快速且可扩展的网络表示学习。发表于第二十八届国际人工智能联合会议论文集，*IJCAI-19*，第4278–4284页。国际人工智能联合会议组织，2019年7月。doi:10.24963/ijcai.2019/594。URL https://doi.org/10.24963/ijcai.2019/594。Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。注意力机制就是一切。神经信息处理系统进展，30，2017年。Hanchen Wang、Rong Hu、Ying Zhang、Lu Qin、Wei Wang 和 Wenjie Zhang。采用 Wasserstein估计器的神经子图计数。第160–175页。计算机协会，2022年6月。ISBN 9781450392495。doi:10.1145/3514221.3526163。Aidan Hogan、Cristian Riveros、Carlos Rojas 和 Adrián Soto。一种面向 SPARQL 的最坏情况最优连接算法。载于 Chiara Ghidini、Olaf Hartig、Maria Maleshkova、Vojtech Svátek、Isabel F. Cruz、Aidan Hogan、Jie Song、Maxime Lefrançois 和 Fabien Gandon 编，语义网：*ISWC 2019*——第*18*届国际语义网会议，新西兰奥克兰，*2019*年*10*月*26*日至*30*日，论文集，第*I*部分，第11778卷，属于《计算机科学讲义》，第258–275页。Springer，2019年。doi:10.1007/978-3-030-30793-6_15。URL https://doi.org/ 10.1007/978-3-030-30793-6_15。Wei Emma Zhang、Quan Z. Sheng、Yongrui Qin、Kerry Taylor 和 Lina Yao。基于学习的 SPARQL查询性能建模与预测。万维网，21(4):1015–1035，2018年。doi:10.1007/s11280-017-0498-1。URL https://doi.org/10.1007/s11280-017-0498-1。Rakebul Hasan 和 Fabien Gandon。SPARQL查询性能预测。载于 Valentina Presutti、Eva Blomqvist、Raphaël Troncy、Harald Sack、Ioannis Papadakis 和 Anna Tordai 编，语义网：*ESWC 2014* 卫星活动——*ESWC 2014* 卫星活动，希腊克里特岛阿尼萨拉斯，*2014*年*5*月*25*日至*29*日，修订选编论文，第8798卷，属于《计算机科学讲义》，第222–225页。Springer，2014年。doi:10.1007/978-3-319-11955-7_23。URL https://doi.org/10.1007/978-3-319-11955-7_23。Daniel Casals、Carlos Buil-Aranda 和 Carlos Valle。利用深度学习进行 SPARQL查询执行时间预测。Vitaly Maiorov 和 Allan Pinkus。多层感知器神经网络逼近的下界。*Neurocomputing*，25(1-3): 81–91，1999年。Petar Ristoski 和 Heiko Paulheim。RDF2Vec：用于数据挖掘的 RDF图嵌入。URL http://www.rapidminer. com/。Agustinus Kristiadi、Mohammad Asif Khan、Denis Lukovnikov、Jens Lehmann 和 Asja Fischer。将字面量纳入知识图谱嵌入。载于 Chiara Ghidini、Olaf Hartig、Maria Maleshkova、Vojtech Svátek、Isabel F. Cruz、Aidan Hogan、Jie Song、Maxime Lefrançois 和 Fabien Gandon 编，语义网：*ISWC 2019*——第*18*届国际语义网会议，新西兰奥克兰，*2019*年*10*月*26*日至*30*日，论文集，第*I*部分，第11778卷，属于《计算机科学讲义》，第347–363页。Springer，2019年。doi:10.1007/978-3-030-30793-6_20。URL https://doi.org/10.1007/978-3-030-30793-6_20。

Patryk Preisner 和 Heiko Paulheim。用于嵌入含字面量知识图谱的通用预处理算子，2023。Jan Portisch、Michael Hladik 和 Heiko Paulheim。RDF2Vec Light——一种用于知识图谱嵌入的轻量级方法。*CoRR*，abs/2009.07659，2020年。URL https://arxiv.org/abs/2009.07659。Deli Chen、Yankai Lin、Wei Li、Peng Li、Jie Zhou 和 Xu Sun。从拓扑视角衡量并缓解图神经网络的过平滑问题。发表于*AAAI* 人工智能会议论文集，第34卷，第3438–3445页，2020年。Weihua Hu、Bowen Liu、Joseph Gomes、Marinka Zitnik、Percy Liang、Vijay S. Pande 和 Jure Leskovec。图神经网络的预训练策略。发表于第*8*届国际学习表征会议，*ICLR 2020*，埃塞俄比亚亚的斯亚贝巴，*2020*年*4*月*26*日至*30*日。OpenReview.net，2020年。URL https://openreview.net/forum?id=HJlWWJSFDH。Diederik P. Kingma 和 Jimmy Ba。Adam：一种随机优化方法。载于 Yoshua Bengio 和 Yann LeCun 编，第*3*届国际学习表征会议，*ICLR 2015*，美国加利福尼亚州圣迭戈，*2015*年*5*月*7*日至*9*日，会议轨道论文集，2015年。URL http://arxiv.org/abs/1412.6980。Knud Möller、Tom Heath、Siegfried Handschuh 和 John Domingue。语义网狗粮的配方——ESWC 和 ISWC 元数据项目。载于 Karl Aberer、Key-Sun Choi、Natasha Fridman Noy、Dean Allemang、Kyung-Il Lee、Lyndon J. B. Nixon、Jennifer Golbeck、Peter Mika、Diana Maynard、Riichiro Mizoguchi、Guus Schreiber 和 Philippe Cudré-Mauroux 编，语义网，第*6*届国际语义网会议，第*2*届亚洲语义网会议，*ISWC 2007 + ASWC 2007*，韩国釜山，*2007*年*11*月*11*日至*15*日，第4825卷，属于《计算机科学讲义》，第802–815页。Springer，2007年。doi:10.1007/978-3-540-76298-0_58。URL https://doi.org/10.1007/978-3-540-76298-0_58。Yuanbo Guo、Zhengxiang Pan 和 Jeff Heflin。LUBM：OWL 知识库系统的基准测试。*J.Web Semant.*，3(2-3):158–182，2005年。doi:10.1016/j.websem.2005.06.005。URL https://doi.org/10.1016/j.websem.2005. 06.005。Fabian M. Suchanek、Gjergji Kasneci 和 Gerhard Weikum。YAGO：源自维基百科和 WordNet 的大型本体。*WebSemantics*，6:203–217，2008年9月。ISSN 15708268。doi:10.1016/j.websem.2008.06.001。Xiaozhi Wang、Tianyu Gao、Zhaocheng Zhu、Zhengyan Zhang、Zhiyuan Liu、Juanzi Li 和 Jian Tang。KEPLER：知识嵌入与预训练语言表征的统一模型。*Trans. Assoc. Comput. Linguistics*，9:176–194，2021年。doi:10.1162/tacl_a_00360。URL https://doi.org/10.1162/tacl_a_00360。Gilles Vandewiele、Bram Steenwinckel、Terencio Agozzino 和 Femke Ongenae。pyrdf2vec：RDF2Vec 的 Python 实现与扩展。2022年。doi:10.48550/ARXIV.2205.02283。URL https://arxiv.org/abs/2205.02283。Matthäus Zloch、Maribel Acosta、Daniel Hienert、Stefan Dietze 和 Stefan Conrad。用于分析 RDF图图度量的软件框架和数据集。发表于语义网：第*16*届国际会议，*ESWC 2019*，斯洛文尼亚波多罗兹，*2019*年*6*月*2*日至*6*日，论文集*16*，第523–539页。Springer，2019年。Angela Bonifati、Wim Martens 和 Thomas Timm。大型 SPARQL查询日志的分析研究。《*VLDB*期刊》，29(2-3):655–679，2020年。Python软件基金会。Python标准库——time。https://docs.python.org/3/library/time.html，2021年。[在线；访问日期：27-February-2023]。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——从维基百科提取的大规模多语言知识库。语义网，6(2):167–195，2015年。doi:10.3233/SW-140134。URL https://doi.org/10.3233/SW-140134。Maribel Acosta 和 Maria-Esther Vidal。关联数据涡流网络：面向 RDF数据的自适应网页查询处理引擎。载于 Marcelo Arenas、Óscar Corcho、Elena Simperl、Markus Strohmaier、Mathieu d'Aquin、Kavitha Srinivas、Paul Groth、Michel Dumontier、Jeff Heflin、Krishnaprasad Thirunarayan 和 Steffen Staab 编，语义网：*ISWC 2015*——第*14*届国际语义网会议，美国宾夕法尼亚州伯利恒，*2015*年*10*月*11*日至*15*日，论文集，第*I*部分，第9366卷，属于《计算机科学讲义》，第111–127页。Springer，2015年。doi:10.1007/978-3-319-25007-6_7。URL https://doi.org/10.1007/978-3-319-25007-6_7。