arXiv:2303.00286v4 [cs.LG] 5 Mar 2024

区别对待不同的负样本：利用定义域和值域约束丰富链接预测的损失函数

NicolasHubert[1], [2][0000−0002−4682−422X]、Pierre Monnin[3][0000−0002−2017−8426]、Armelle Brun[2][0000−0002−9876−6906]和 Davy Monticolo[1][0000−0002−4244−684X]

[1] 洛林大学，ERPI，南锡，法国[2] 洛林大学，法国国家科学研究中心，LORIA，南锡，法国[3] 蔚蓝海岸大学，Inria，法国国家科学研究中心，I3S，索菲亚-安提波利斯，法国 {nicolas.hubert,armelle.brun,davy.monticolo}@univ-lorraine.fr pierre.monnin@inria.fr

摘要。 知识图谱嵌入模型（知识图谱嵌入模型）被用于知识图谱相关的各种任务，包括链接预测。它们通过损失函数进行训练，这些损失函数会处理由真实三元组和错误三元组组成的批次。然而，错误三元组存在不同类型，近期研究表明，不应对它们一视同仁，因此需要采用特定的负采样方法。基于这一近期假设，我们认为，相对于关系签名（定义域和值域）而言在语义上有效的负三元组是高质量负样本。因此，我们对链接预测的三种主要损失函数进行了扩展，使其能够采样所有类型的负样本，但根据其语义有效性区别对待这些负样本。在广泛且受控的实验设置中，我们表明，所提出的损失函数能够系统性地取得令人满意的结果，这证明了我们所提方法的普适性和优越性。事实上，所提出的损失函数（1）带来更好的平均倒数排名和 Hits@10 值，并且（2）推动知识图谱嵌入模型获得更好的语义正确性，该指标通过 Sem@K metric 进行衡量。这表明，关系签名能够全面改善知识图谱嵌入模型，因此应当被纳入损失函数中。由于模式定义的知识图谱通常广泛提供关系的定义域和值域，这使得我们的方法在实践中既有益又具有广泛的可用性。

关键词： 知识图谱嵌入 · 链接预测 · 基于模式的学习 · 损失函数。

1 引言

知识图谱（KG）是由三元组组成的集合（h， r， t），其中 h（头实体）和 t（尾实体）是图中的两个实体，而 r 是用于限定二者之间关系性质的谓词。在本文中，我们不考虑字面量。知识图谱本质上是不完整的、不正确的或相互重叠的，因此主要的细化任务包括实体匹配和链接预测[29]。后者是本文的重点

。链接预测（LP）旨在利用已有事实推断缺失事实，从而补全知识图谱。在链接预测任务中，给定一组不完整三元组，需要预测缺失的头实体（或尾实体）。这相当于给定一组三元组，其中每个三元组的头实体 h 或尾实体 t缺失。

链接预测任务通常采用知识图谱嵌入模型（KGEMs）来解决。这些模型将实体和关系表示为潜在嵌入空间中的低维向量，尽可能保留图结构信息和图的属性 [15]。文献中已提出大量知识图谱嵌入模型。它们通常在评分函数方面有所不同，也就是说，在实体之间交互方式的建模上有所不同。实体和关系的嵌入通过依赖损失函数的优化过程，经过多个训练轮次学习得到。损失函数旨在最大化知识图谱嵌入模型对正三元组（即图中存在的三元组）的评分，并最小化对负三元组（即图中不存在的三元组）的评分。近期，部分研究开始探讨负三元组对知识图谱嵌入模型训练的影响[17]。事实上，负三元组通常通过扰动真实三元组生成，这一过程称为负采样；也就是说，将其头实体或尾实体替换为从图中随机选取的另一个实体。随后，研究者提出了改进的采样机制，例如选择与待替换实体类型相同的实体 [17]。

后一种方案属于这样一类工作：将支撑知识图谱的语义作为额外信息，以提升链接预测任务的结果。过去几年中，语义信息已被纳入知识图谱嵌入模型的多个部分，例如，如前所述，负采样过程 [14,18], ，以及模型本身 [6,19,23,28,32], 或损失函数 [5,7,10,20]。现有提出将语义信息纳入损失函数的工作展现出了令人鼓舞的结果。然而，它们仅限于特定的损失函数 [7,10], ，或者考虑了不包含定义域和值域公理（或关系签名）的本体公理 [7,20] ；而这些公理在知识图谱中广泛可用。有趣的是，这类公理可用于生成不同类型的负三元组，即，遵循关系签名的负三元组（称为语义有效）以及不遵循关系签名的负三元组（称为语义无效）。此外，据我们所知，负三元组此前仅在采样过程中得到研究：选择特定的负三元组用于训练，而舍弃其他负三元组。对所有类型的负例进行采样、但在损失函数中以不同方式对待它们的可能性尚未得到评估。这一双重观察引出了我们的第一个研究问题：

研究问题1 链接预测中使用的主要损失函数如何纳入定义域和值域约束，根据负三元组的语义有效性对其进行差异化处理?

先前工作也指出，纳入本体信息可提升性能，这一点可通过平均倒数排名（MRR）和 Hits@K [5,7,10,20] 等基于排名的指标来衡量。然而，尽管这些方法将语义信息作为知识图谱嵌入模型的输入，所得知识图谱嵌入模型的语义能力

仍未得到评估，尽管这有助于更全面地了解其性能 [11,13]。因此，我们提出第二个研究问题：

RQ2 将关系签名纳入损失函数会如何影响知识图谱嵌入模型的整体性能？

为回答这两个问题，我们提出了签名驱动损失函数，即包含依赖于某些背景知识（BK）的项的损失函数，这些背景知识涉及实体类型以及关系的定义域和值域。为扩大我们的方法的影响，我们关注文献中最常见的三种损失函数：成对铰链损失（PHL）[3], 1-N 二元交叉熵损失（BCEL） [8], 以及逐点逻辑损失（PLL） [27]（详见第3节）。针对每一种损失函数，我们都提出了量身定制的签名驱动版本。所考虑的背景知识可从许多模式定义的知识图谱中获取 [9],，因此这些新提出的损失函数在实践中具有广泛的适用性。此外，我们使用基于排名的指标和 Sem@K [11,13] 来评估损失函数的影响。Sem@ 是一种衡量知识图谱嵌入模型对链接预测任务的预测与关系签名一致性的指标，即预测的语义正确性。

总而言之，本研究的主要贡献如下：

– 我们针对链接预测任务中最常用的三种损失函数提出了签名驱动版本，利用了关于关系定义域和值域的背景知识。– 我们采用传统的基于排名的指标评估了该方法，同时也使用了 Sem@K，后者能更深入地揭示我们所提方法的优势。– 我们表明，所设计的签名驱动损失函数在大多数情况下，无论是基于排名的指标还是 Sem@K，都能取得更好的性能。因此，我们的研究结果有力地表明，应将签名信息系统地融入损失函数中。

本文其余部分安排如下。第2节介绍相关工作。第3节详细阐述本文提出的签名驱动损失函数。第4节介绍数据集和实验设置。第5 节呈现主要发现，并在第6节进一步讨论。最后，第7 节总结主要发现并概述未来研究方向。

2 相关工作

本节首先介绍利用语义信息提升链接预测任务模型结果的相关研究。重点在于如何将语义信息融入损失函数（第 2.1 节）——与将语义信息融入学习过程的其他环节相比，这一研究方向仍相对缺乏探索，例如负采样或交互函数。其次，本文简要介绍主流损失函数的背景（第 2.2 节）。这有助于将我们的贡献置于实践中所使用的原始损失函数的背景下。

2.1 语义增强方法

大量文献提出了利用语义信息进行链接预测的方法，这些方法借助知识图谱嵌入模型，旨在提升知识图谱嵌入模型相对于传统基于排名的指标的性能。

最直接的方法是在模型本身中嵌入语义信息。例如，AutoETER [23] 是一种自动化的类型表示学习机制，可与任何知识图谱嵌入模型搭配使用，并学习每个实体的潜在类型嵌入。在 TaRP [6], 中，类型信息和实例级信息会同时纳入考量，并分别编码为关系的先验概率和似然。TKRL [32] 将类型信息与层级信息相结合：类型信息用作关系特定约束，而层次类型则编码为实体的投影矩阵。TKRL 允许实体在不同类型下具有不同的表示。类似地，[28], 所提出的知识图谱嵌入模型允许实体根据各自的类型具有不同的向量表示。TransC [19] 将知识图谱中的每个概念编码为一个球体，并将每个实例编码为同一语义空间中的一个向量。概念与实例之间的关系（`rdf:type`）以及概念与子概念之间的关系（`rdfs:subClassOf`），均基于这一共享语义空间中的相对距离。

少数研究将语义信息融入负采样（NS）过程，以生成有意义的负三元组 [18,14,31]。例如，类型约束负采样（TCNS） [18] 会将三元组的头实体或尾实体替换为一个随机实体，该实体与真实实体属于相同类型（rdf:type）。Jain 等人 [14] 更进一步，利用本体推理，通过在不一致预测上重新训练模型，迭代提升知识图谱嵌入模型的性能。值得注意的是，我们的方法采取了与这类研究正交的方向：在训练时，我们提出对所有类型的负样本进行采样，但在损失函数中区别对待它们。

实际上，少数研究提出在学习和优化过程中纳入语义信息。在[10], 嵌入空间中，同一语义类别的实体嵌入被约束在相近的邻域内。然而，他们的方法仅适用于单类型知识图谱。此外，这项研究仅以 TransE 作为主流模型进行基准测试 [3], ，且仅使用成对铰链损失。同样，d'Amato 等人 [7] 也仅考虑成对铰链损失，且其方法仅针对平移模型进行基准测试。此外，背景知识以`equivalentClass`、`equivalentProperty`、`inverseOf`和`subClassOf`公理的形式注入，与 [20] 将`equivalentProperty`和`inverseOf`公理作为损失函数中的正则化项的方法类似。然而，知识图谱中很少提供上述公理 [9]。Cao 等人 [5] 提出了一种名为等变正则器的新型正则化器，通过利用语义信息来限制过拟合。然而，他们的方法是数据驱动的，并不依赖模式。相比之下，第3 节介绍的方法利用了大多数模式定义的知识图谱中都具备的定义域和值域约束。

最后，所有上述语义驱动方法的评估都仅基于基于排名的指标。然而，语义驱动方法

若采用以语义为导向的评估方式，则会K [11,12,13] 受益。据我们所知，围绕 Sem@K [11,12,13] 的研究是唯一提供适当工具来衡量知识图谱嵌入模型语义正确性的方法。因此，我们的实验也将采用这一指标进行评估。

2.2 链接预测任务的损失函数

很少有研究关注损失函数对知识图谱嵌入模型性能的影响 [1,22,21]。Mohamed 等人 [22] 指出，人们尚未考虑损失函数对知识图谱嵌入模型性能的影响。[1] 提供的实验结果表明，没有哪种损失函数始终能取得最佳效果；实际上，评分函数与损失函数的组合才会影响知识图谱嵌入模型的性能。具体而言，某些评分函数与特定损失函数的匹配效果更好。例如，Ali 等人 表明，采用恰当的损失函数时，TransE 的表现可以优于最先进的知识图谱嵌入模型。同样，Mohamed 等人 [22] 表明，损失函数的选择会显著影响知识图谱嵌入模型的性能。因此，他们对文献中使用的主要损失函数进行了全面的基准研究。具体来说，他们的分析依据的是一种普遍认可的分类方法，即将损失函数分为逐点损失函数和成对损失函数。逐点损失函数与成对损失函数的主要区别在于，它们对评分函数、三元组及其各自标签的综合考量方式不同。在逐点方法中，损失函数依据三元组的预测分数及其实际标签值计算，其中正三元组的标签通常为 1，负三元组的标签则为 0（或 -1）。相比之下，成对损失函数则根据真实三元组的预测分数与其对应负三元组的分数之间的差值来定义。

在我们的方法中（第 3 节），我们考虑用于执行 LP 的三种最常用损失函数 [24]：成对铰链损失（PHL） [3]，、1-N 二元交叉熵损失（BCEL） [8]，以及逐点逻辑损失（PLL） [27]。它们的标准公式见公式（1）、（2）和（3）。

$$\mathcal{L}_{PHL} = \sum_{t\in\mathcal{T}^+}\sum_{t'\in\mathcal{T}^-}[\gamma + f(t') - f(t)]_+ \tag{1}$$

其中，$\mathcal{T}$、f 和 $[x]_+$ 分别表示一批三元组、评分函数以及 x的正部。$\mathcal{T}$进一步分为一批正三元组$\mathcal{T}^+$ 和一批负三元组 $\mathcal{T}^-$。γ 是一个可配置的间隔超参数，用于指定正三元组的得分与对应负三元组的得分之间应保持多大的差距。

$$\mathcal{L}_{BCEL} = -\frac{1}{|\mathcal{E}|}\sum_{t\in\mathcal{T}}\ell(t)\log(f(t)) + (1-\ell(t))\log(1-f(t)) \tag{2}$$

其中，$\ell(t) \in \{0,1\}$ 表示 t 的真实标签，$\mathcal{T}$是包含所有可能的（h、r、$*$）的批次。$|\mathcal{E}|$ 是知识图谱中的实体数量。

$$\mathcal{L}_{PLL} = \sum_{t\in\mathcal{T}}\log(1+\exp^{-\ell(t)\cdot f(t)}) \tag{3}$$

其中， $\ell(t) \in \{-1, 1\}$ 表示 t的真实标签。

3 签名驱动的损失函数

基于以往工作的局限性（第2.1节），我们提出了签名驱动损失函数，对三种最常用的损失函数 [24] 进行扩展，并利用有关关系的定义域和值域的背景知识；文献中使用的许多知识图谱都提供了这类信息。

所提出的损失函数旨在区分语义有效的负例与语义无效的负例。前者定义为满足关系 r 的定义域和值域的三元组（h， r， t），即，

$$\text{type}(h) \cap \text{domain}(r) \neq \emptyset \wedge \text{type}(t) \cap \text{range}(r) \neq \emptyset$$

而后者至少违反其中一项约束，即

$$\text{type}(h) \cap \text{domain}(r) = \emptyset \vee \text{type}(t) \cap \text{range}(r) = \emptyset.$$

domain(r)（分别地，range(r)）定义为关系 r 的头实体（分别地，尾实体）的预期类型。例如，关系总统所属国家要求头实体为人物，尾实体为国家。从表示真实事实的正三元组（埃马纽埃尔·马克龙，总统所属国家，法国）出发，（巴拉克·奥巴马，总统所属国家，法国）和（埃马纽埃尔·马克龙，总统所属国家，德国）是语义有效的负三元组示例，而（阿迪达斯，总统所属国家，法国）和（埃马纽埃尔·马克龙，总统所属国家，圣诞节）是语义无效的负三元组示例。在本研究中，实体具有多种类型。因此，type(e) 返回实体 e 所属类型（又称类别）的集合。

我们引入一个与损失函数无关的 ϵ因子，称为语义因子，旨在使语义有效的负三元组的得分更接近正三元组。这个通用语义因子能够适用于不同的损失函数，体现了我们方法的普适性，也表明该方法有望扩展到其他损失函数。有趣的是，这个因子还使我们能够在一定程度上考虑知识图谱所遵循的开放世界假设（OWA）。在开放世界假设下，知识图谱中未表示的三元组可能是错误的，也可能是缺失的正三元组。在传统训练过程中，这些三元组一概被视为负例，这与封闭世界假设相对应。与此相反，我们的方法将语义无效的三元组视为真正的负例，而使语义有效的三元组（在开放世界假设下，它们可能是缺失的正例，也可能是假负例）更接近真正的正三元组。这一做法假设实体类型是完整且准确的。

$\mathcal{L}_{\mathbf{PHL}}$ 在公式（1）中定义的函数取决于间隔超参数 γ。增大（分别减小）γ 的值会增大（分别减小）正三元组与负三元组得分之间设定的间隔。然而，这种单一的 γ 对所有负三元组一视同仁：语义有效和语义无效负三元组的得分，与它们各自源自的正三元组得分之间的间隔相同。我们建议

应区别对待这两类负三元组的得分。因此，我们的方法将 $\mathcal{L}_{PHL}$ 重新定义如下（公式（4））：

$$\mathcal{L}^{S}_{PHL} = \sum_{t \in \mathcal{T}^{+}} \sum_{t' \in \mathcal{T}^{-}} [\gamma \cdot \ell(t') + f(t') - f(t)]_{+}$$
$$\text{where } \ell(t') = \begin{cases} 1 \text{ if } t' \text{ is semantically invalid} \\ \epsilon \text{ otherwise} \end{cases} \tag{4}$$

公式（4）中的损失函数现在带有上标 S，以明确表明这是 vanilla $\mathcal{L}_{PHL}$（定义见公式（1））的签名驱动版本。选择 $\epsilon < 1$ 会使知识图谱嵌入模型对正三元组与语义无效三元组的分数之间采用比正三元组与语义有效三元组之间更大的间隔。对于给定的正三元组，这样可以使其语义有效负例对应的分数相对更接近，而语义无效负例对应的分数则相对更远。直观地说，当知识图谱嵌入模型输出错误预测时，我们仍然期望其中更多预测满足关系施加的定义域和值域约束。因此，我们假设错误预测更有意义，并且在某种意义上，在语义上更接近真实三元组。

$\mathcal{L}_{\mathbf{BCEL}}$ 公式（2）中定义的内容被调整为 $\mathcal{L}^{S}_{BCEL}$，方法是重新定义标注函数 ℓ。具体而言，对于具有带类型实体、且提供关系定义域和值域信息的知识图谱，标注函数 ℓ 不再是二元函数。相反，可以将语义有效负三元组的标签设为介于正三元组和语义无效负三元组的标签值之间的某个中间值，从而得到公式（ ℓ5）中定义的标注函数:

$$\ell(t') = \begin{cases} 1 \text{ if } t' \in \mathcal{T}^{+} \\ \epsilon \text{ if } t' \in \mathcal{T}^{-} \text{ and } t' \text{ is semantically valid} \\ 0 \text{ if } t' \in \mathcal{T}^{-} \text{ and } t' \text{ is semantically invalid} \end{cases} \tag{5}$$

其中，语义因子 ϵ 是一个可调超参数，表示语义有效负三元组的标签值。改进标注函数 ℓ 背后的直觉，是有意在语义有效负三元组与正三元组之间制造一定程度的混淆。通过弥合二者各自的标签值，期望知识图谱嵌入模型能够在某种程度上将前者视为“负性较弱的三元组”，并赋予其相较于正三元组更高的得分。

$\mathcal{L}_{\mathbf{PLL}}$ 在公式（3）中定义的函数可以调整为 $\mathcal{L}^{S}_{PLL}$ 类似于 $\mathcal{L}^{S}_{BCEL}$。换句话说，标注函数 ℓ 也可以为语义有效的负三元组输出一个中等标签值 ϵ。尽管这种方法在 Sem@K 值方面取得了非常好的结果，但在不同数据集上的表现并不稳定。此外，在 MRR 和 Hits@K 方面取得的结果可能远低于使用原始模型得到的结果（更多细节请参阅补充材料）。因此，为了区别对待语义有效和无效的负三元组，这里不再修改标注

函数 ℓ，语义因子 ϵ 对于 $\mathcal{L}^S_{PLL}$ 定义了将语义有效的负三元组视为正三元组并以相同方式进行标注的概率。例如，使用 $\epsilon = 0.05$时，在每个训练轮次中，对于每个批次，该训练批次中的语义有效负三元组将以 5% 的概率被视为正三元组。

值得注意的是，我们的方法在实践中适用于具有或不具有类型、定义域和值域的知识图谱。事实上，在缺少此类背景知识时，我们的签名驱动损失函数会退化为各自的原始对应方法。请注意，我们的方法并不关注负采样或复杂的负样本生成器，例如 KBGAN [4], NSCaching [34], 以及自对抗负采样 [26]。尽管这些工作与我们相关，但它们在上游限制了负采样。相比之下，我们的方法不限制负三元组的采样，而是根据其语义有效性，将负三元组动态分配到损失函数的不同部分。

4 实验设置

4.1 评估指标

在我们的实验中，知识图谱嵌入模型的性能依据平均倒数排名、Hits@K 和 Sem@K 进行评估，其中 $K = 10$。**平均倒数排名（MRR）**对应于真实三元组排名倒数的算术平均值。MRR 的取值范围为 $[0, 1]$，数值越高越好。**Hits@K** 表示排名得分最高的前 K 个三元组中真实三元组所占的比例。该指标的取值范围为 $[0, 1]$，其数值随K增大而增大，数值越高越好。**Sem@K**[11,13] 表示排名得分最高的前 K 个三元组中语义有效三元组所占的比例：

$$\text{Sem@}K = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \frac{1}{K} \sum_{q' \in \mathcal{S}^K_q} \text{compatibility}(q, q') \tag{6}$$

其中，对于真实三元组 $q = (h, r, t)$，$\mathcal{S}^K_q$是由知识图谱嵌入模型评分的前K个候选三元组列表，即，通过预测（$h, r,$?）中的尾实体或（? r, t）中的头实体得到。候选三元组 q' 根据 q 进行评估，所采用的兼容性函数会检查预测的头实体（分别为尾实体）是否属于该关系的定义域（分别为值域）。本文考虑了类层次结构：如果某个关系的定义域（分别为值域）是给定类别，则来自其子类的实体被视为语义有效。Sem@K 的取值范围为 $[0, 1]$ 区间。

4.2 数据集和模型

尽管我们的方法可以应用于具有或不具有关系签名的知识图谱，但我们在理想的实验设置下评估该方法，以准确说明考虑定义域和值域的意义。首先，训练集中的所有关系都具有明确的定义域和值域。其次，二者

训练三元组的头实体 h 和尾实体 t 都有另一个语义有效的对应实体用于负采样。这两个条件保证每个正训练三元组至少可以与一个语义有效三元组配对。最后，验证集和测试集包含这样的三元组：其关系具有明确的定义域（相应地为值域），并且分别作为头实体（相应地为尾实体）时具有超过 10 个语义有效的候选实体。这确保 Sem@K 不会受到不当惩罚，并且在 $K = 10$ 之前可以在与 Hits@K 和平均倒数排名相同的实体集合上计算。

为确保满足这些要求，我们对 FB15k237-ET、DBpedia93k 和 YAGO4-19k 进行了筛选 [13]，使其符合上述标准。下文分别将筛选后的版本称为 FB15k187、DBpedia77k 和 Yago14k。表 1 列出了这些数据集的统计信息，体现了它们特征的多样性。本文考虑了若干知识图谱嵌入模型：TransE [3], TransH [30], 和 DistMult [33] 使用 $\mathcal{L}_{PHL}$；ComplEx [27] 和 SimplE [16] 使用 $\mathcal{L}_{PLL}$；ConvE [8], TuckER [2], 和 RGCN [25] 使用 $\mathcal{L}_{BCEL}$。数据集和代码可在我们的 Git Hub 仓库中获取。[4]

表 1. 实验中使用的数据集。这些数据集是对标准 FB15k237-ET、DBpedia93k 和 YAGO4-19k 筛选后的版本。

数据集	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}_{train}\rvert$	$\lvert\mathcal{T}_{valid}\rvert$	$\lvert\mathcal{T}_{test}\rvert$	划分比例
FB15k187	14, 305	187	245, 350	15, 256	17, 830	88%/5.5 %/6.5%
DBpedia77k	76, 651	150	140, 760	16, 334	32, 934	74%/ 9 %/ 17%
Yago14k	14,178	37	18, 263	472	448	95%/ 2.5%/2.5%

4.3 实现细节

为便于比较，在从头开始训练模型后计算平均倒数排名、Hits@K 和 Sem@K 。实验中使用的知识图谱嵌入模型均采用 PyTorch 实现。在经过大量训练轮次后，我们注意到，最佳结果通常出现在训练轮次 400 或更少时。因此，参照 LibKGE，训练轮次上限设为 400 轮[5]（RGCN 除外，由于其收敛到最佳结果的速度较慢，训练 4,000 轮）。除使用无需负采样的二元交叉熵损失外，均采用均匀随机负采样 [3]，将每个训练三元组分别与两个对应的负三元组配对：一个语义无效，另一个语义有效。对于二元交叉熵损失，每个正三元组都会与通过图中所有其他实体构造的负三元组进行评分。因此，负三元组既包括语义有效的三元组，也包括语义无效的三元组。需要注意的是，每种模型与数据集组合的最佳 epsilon 值均在验证集上确定。确定最佳 epsilon 值后，它们

[4] https://github.com/nicolas-hbt/semantic-lossfunc/[5] https://github.com/uma-pi1/kge/

在所有三元组中保持不变。为确保模型之间进行公平比较，嵌入采用相同的随机种子初始化，并且每个模型在每个训练轮次接收完全相同的负三元组集合。基于预定义超参数进行了网格搜索。最佳超参数和完整的超参数空间已发布在GitHub 上。[4]

5 结果

5.1 整体性能

表2。 基于排名和基于语义的结果。粗体表示在给定指标下表现最佳的模型。后缀 V 和 S 分别表示模型是在损失函数的原始版本或签名驱动版本下训练的。Hits@10 和 Sem@10 分别缩写为 H@10和 S@10。带下划线的单元格表示在第 5.1 节中有更具体讨论的结果。当比较一场对决（† 例如， TransE-V 与 TransE-S）的结果在给定指标和数据集上具有统计显著性时，我们使用符号$\alpha = .05$。

	FB15k187			DBpedia77k			Yago14k		
	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分
TransE-V	.260	.446	.842	.274	.438	.936	.868	**.945**	.795
TransE-S	**.315**†	**.497**†	**.973**†	**.275**	**.440**	**.985**†	**.876**	.944	**.968**†
TransH-V	.266	.450	.855	.270	.437	.907	.836	.944	.581
TransH-S	**.319**†	**.501**†	**.973**†	**.274**†	**.442**†	**.980**†	**.857**†	**.945**	**.831**†
DistMult-V	.291	.457	.824	.295	.405	.784	.904	**.930**	.409
DistMult-S	**.332**†	**.504**†	**.971**†	**.300**†	**.416**†	**.901**†	**.912**†	.929	**.449**†
ComplEx-V	.280	.416	.472	**.309**†	**.415**†	.769	**.925**	**.932**	.333
ComplEx-S	**.316**†	**.476**†	**.796**†	.297	.409	**.897**	.923	.931	**.667**†
SimplE-V	.261	.387	.462	**.259**†	**.346**†	**.883**†	**.926**	**.931**	.355
SimplE-S	**.268**†	**.409**†	**.759**†	.230	.302	.850	.924	.927	**.769**†
ConvE-V	.273	.470	.973	.273	.382	.935	**.934**	**.942**	.904
ConvE-S	**.283**†	**.476**†	**.996**†	**.283**†	**.405**†	**.985**†	.933	.940	**.997**†
TuckER-V	.316	.516	.985	.311	.410	.912	.923	.927	.781
TuckER-S	**.320**†	**.522**†	**.996**†	**.312**	**.421**†	**.969**†	**.931**†	**.943**†	**.929**†
RGCN-V	.241	.386	.775	.194	.297	.872	.911	.923	.349
RGCN-S	**.260**†	**.415**†	**.860**†	**.197**†	**.320**†	**.957**†	**.927**†	**.934**†	**.828**†

表2 展示了知识图谱嵌入模型（KGEM）的性能、数据集和我们所关注的评估指标。我们在显著性水平 $\alpha = .05$下进行了t检验（当根据夏皮罗检验，与预测相关的数据服从正态分布时）和Wilcoxon检验（当数据不服从正态分布时）。表2 清楚地表明，除了在DBpedia77k上使用SimplE的情况外，签名驱动损失函数 $\mathcal{L}^{S}_{PHL}$、$\mathcal{L}^{S}_{BCEL}$和 $\mathcal{L}^{S}_{PLL}$ 均能在Sem@10方面带来显著提升。

值得注意的是，在某些情况下，与相应的标准损失函数相比，Sem@10的相对提升幅度非常大（表2中以下划线标出）：在规模较小的Yago14k数据集上，RGCN、SimplE和ComplEx分别提升了 $+137$%、 $+117$%和 $+100$%；在FB15k187上，ComplEx和SimplE分别提升了 $+69$%和 $+64$%。无论采用哪种损失函数，都能观察到Sem@10方面的这些提升，这表明我们设计的签名驱动损失函数均可推动知识图谱嵌入模型的语义正确性取得更令人满意的结果。值得注意的是，在大多数情况下，按照平均倒数排名和Hits@10衡量时，它们也能带来更好的知识图谱嵌入模型性能。在同一知识图谱嵌入模型分别使用标准损失函数和签名驱动损失函数进行训练的24次一对一比较中，有19次（≈ 79%）显示，使用签名驱动损失函数训练的模型具有更高的平均倒数排名。在其余比较中（≈ 21%），只有两次显示平均倒数排名方面存在统计显著的下降。然而，这些下降幅度很小，而且通常换来了显著更高的Sem@10值。这一观察引出了一个问题：是否应不惜一切代价追求更高的平均倒数排名和Hits@K，抑或如果这能带来显著更高的知识图谱嵌入模型语义正确性，那么相对于这些指标的小幅下降是否可以接受（见第6.3节）。此外，这些令人鼓舞的结果表明，即使预期目的只是最大化平均倒数排名和Hits@K 值，也强烈建议考虑可用的签名信息：这不仅能提升知识图谱嵌入模型的语义正确性，还能在平均倒数排名和Hits@K方面带来性能提升。

下面，我们对结果进行更细致的分析，重点关注实验中使用的不同损失函数和数据集。尽管我们此前已展示了该方法的有效性，但以关系定义域和值域的形式考虑背景知识所带来的益处，因损失函数而异。具体而言，使用 $\mathcal{L}^S_{PHL}$ 和 $\mathcal{L}^S_{BCEL}$ 所取得的提升十分显著。采用这些损失函数时，我们观察到 Sem@10 持续改善。在绝大多数情况下，基于排名的指标也有所提升。唯一的例外是 Yago14k：语义损失有时略逊一筹，但与原始对应方法相比仍具竞争力。不过，平均倒数排名和 Hits@10 的差异微乎其微，且在统计上并不显著。这可能是因为 Yago14k 中的三元组数量较少，新增的签名信息虽提升了 Sem@K，却无助于区分真实实体与其他实体。因此，将有关关系定义域和值域的背景知识纳入 $\mathcal{L}^S_{PHL}$ 和 $\mathcal{L}^S_{BCEL}$ 是一种可行的方法，能够在平均倒数排名、Hits@10 和 Sem@10 方面持续带来提升。至于在 $\mathcal{L}^S_{PLL}$下这样做的益处，我们注意到某些情况下平均倒数排名和 Hits@10 略有下降。不过，另一方面，Sem@10 的得分显著更高：除 DBpedia77k 上的 SimplE 外，其余一对一比较中的 Sem@10 得分提升幅度为 $+17$% 至 $+117$%。因此，只要能够接受基于排名的指标略有下降，以换取知识图谱嵌入模型语义正确性的显著提升，采用 $\mathcal{L}^S_{PLL}$ 的方法同样能取得令人满意的结果。此外，值得注意的是以下几点：我们的超参数

调优策略基于在 Yago14k 上选择最佳的 ϵ 值——这是出于计算资源限制。在 Yago14k 上表现最佳的 ϵ 值随后被用于所有其余场景。若能在其他数据集上对 ϵ 进行更充分的调优，则在 $\mathcal{L}^S_{PLL}$ 下或许能取得更令人满意的结果，从而进一步彰显我们方法的价值。

5.2 消融研究

在本节中，我们将知识图谱嵌入模型应用于三组关系进行测试，这些关系分别具有狭窄（B1）、中等（B2）和大型（B3）的语义有效头实体或尾实体集合。阈值由人工设定，详见补充材料[4]。对 B1 的分析有助于我们衡量签名驱动损失函数对这类关系的影响，因为对于这些关系，预测语义有效实体的难度更高。表3 中报告的 B1 结果清楚地表明，将背景知识注入损失函数对这类关系的影响更为显著，从而验证了我们的方法在稀疏且困难的场景中的价值。有人可能会认为，$\mathcal{L}^S_{PHL}$、$\mathcal{L}^S_{BCEL}$和 $\mathcal{L}^S_{PLL}$取得的较高平均倒数排名值与较高的 Sem@10 值高度相关。这种看法有一定道理：对于 B1 中的关系，将所有语义有效候选实体排在排名列表前列，很可能会提升真实实体本身的排名。然而，我们可以看到，在 Yago14k 上，RGCN-V 和 RGCN-S 的平均倒数排名值几乎相同，而 RGCN-S 的 Sem@10 值却远高于 RGCN-V（$+254\%$）。在 Yago14k 上的 ComplEx、DBpedia77k 上的 ConvE，以及 DBpedia77k 和 Yago14k 上的 TransE，也有类似发现。这表明，在许多情况下，签名驱动损失函数能够提升知识图谱嵌入模型对小型关系的语义正确性，同时不影响其基于排名指标的性能。B2 和 B3 的结果见补充材料[4]。特别值得注意的是，在这些分组中，我们的方法在平均倒数排名和 Hits@10 方面的相对收益较小。对于 Sem@K，原始损失函数取得的结果已经较高，因此注入本体背景知识带来的提升相对较小。这些已经较高的 Sem@K 值，可能是由数量更多的语义有效候选实体所致。

6 讨论

6.1 区别对待不同负例（研究问题1）

所提出的损失函数 $\mathcal{L}^S_{PHL}$、$\mathcal{L}^S_{BCEL}$和 $\mathcal{L}^S_{PLL}$ 为知识图谱嵌入模型提供了充分的训练目标，表 2对此给出了证明。最重要的是，在第3 节中，我们清楚地展示了如何借助共同定义的语义因子，将签名信息纳入 $\mathcal{L}^S_{PHL}$、$\mathcal{L}^S_{BCEL}$和$\mathcal{L}^S_{PLL}$ 之中。尽管这一语义因子会因损失函数不同而作用于不同层面，但其共同目的是区分对于语义有效和语义无效的负三元组，相较于正三元组应如何处理；而传统方法则对所有负三元组一视同仁。

表3. 关系桶中狭窄的语义有效头实体或尾实体集上的基于排名和基于语义的结果（B1）。阈值由人工设定，详见补充材料。

	FB15k187			DBpedia77k			Yago14k		
	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分
TransE-V	.535	.727	.647	.498	.578	.460	.914	**.979**	.620
TransE-S	**.646**	**.805**	**.937**	**.539**	**.626**	**.910**	**.922**	.975	**.924**
TransH-V	.541	.734	.661	.475	.555	.425	.897	.972	.436
TransH-S	**.655**	**.814**	**.936**	**.541**	**.643**	**.828**	**.923**	**.981**	**.684**
DistMult-V	.589	.735	.628	.466	.462	.244	.949	**.959**	.304
DistMult-S	**.667**	**.802**	**.929**	**.498**	**.547**	**.451**	**.965**	.958	**.372**
ComplEx-V	.530	.567	.116	**.424**	**.425**	.161	.955	.956	.133
ComplEx-S	**.637**	**.723**	**.537**	.421	.399	**.198**	**.961**	.956	**.423**
SimplE-V	.507	.553	.136	**.396**	**.370**	**.273**	**.959**	.882	**.932**
SimplE-S	**.576**	**.671**	**.505**	.324	.259	.206	.958	**.883**	.930
ConvE-V	.549	.779	.973	.518	**.569**	.789	.969	**.972**	.915
ConvE-S	**.562**	**.783**	**.986**	.518	.566	**.927**	.969	.965	**.960**
TuckER-V	.597	.811	.969	.519	.568	.740	.949	.970	.846
TuckER-S	**.598**	**.815**	**.973**	**.526**	**.582**	**.797**	**.964**	.970	**.892**
RGCN-V	.510	.629	.468	.386	.387	.254	.963	.959	.141
RGCN-S	**.549**	**.705**	**.682**	**.396**	**.415**	**.398**	**.966**	**.967**	**.499**

此外，我们将原始损失函数转化为签名驱动损失函数的方法也适用于其他损失函数，例如逐点合页损失或成对逻辑损失，如 [22], 所示；同样，只需加入语义因子 ϵ 即可。针对这些损失函数的调整及相关实验留待未来工作。

请注意，与 KBGAN [4],、NSCaching [34]，和自对抗负采样 [26]，等复杂负采样器相比，我们无需因维护缓存 [34] 或训练中间对抗学习框架 [4,26] 来生成高质量负样本而引入任何潜在开销。相反，负三元组会根据其语义有效性动态进入损失函数的不同部分。在未来的工作中，我们将比较我们的方法与 NS 过程的性能和算法复杂度，从而突出与复杂 NS 相比，在计算资源和执行时间方面可能节省的成本。我们的方法也不依赖于底层的 NS 过程，可以与简单的均匀随机负采样 [3] 以及更复杂的过程 [4,34,26] 协同工作。此外，即使不存在背景知识，我们的方法也可以应用。在这种情况下，签名驱动损失函数会退化为其原始版本。

6.2 签名驱动损失对性能的影响（RQ2）

Mohamed 等人 [21] 探讨了不同损失函数选择对知识图谱嵌入模型在基于排名的指标方面的可扩展性和性能的影响。在本研究中，我们还评估了此类模型的语义能力。基于

第3节给出并分析的结果，将关系定义域和值域的背景知识纳入损失函数，显然有助于提升知识图谱嵌入模型的语义正确性，这一点从 Sem@K指标经常出现的大幅提升中可见一斑。尽管我们提出的方法似乎直观上会带来更好的语义预测，但这并非理所当然。Hubert 等人证明，在链接预测仅针对单一目标关系这一特殊情形下（例如，推荐系统），在训练过程中引入基于模式的信息（在他们的研究中，是在负采样过程中引入）实际上会降低知识图谱嵌入模型的语义正确性[12]。这一结果使我们认为，在训练过程中注入本体信息并不一定能提升语义正确性。效果是否改善可能取决于：（1）信息被用于何处（例如，用于损失函数中，或在负采样过程中）；（2）当前执行的任务是什么。

还应注意，这种提升并非在所有关系中均匀分布：对于作为头实体或尾实体的语义有效实体集合较小的关系而言，Sem@K 方面的挑战更大（见表3 以及补充材料中 B2 和 B3 的结果[4]）。对于这类关系，签名驱动损失函数带来的相对收益更为显著。此外，签名驱动损失函数还能使大多数知识图谱嵌入模型的平均倒数排名和 Hits@K指标取得提升。如表2所示，在使用 $\mathcal{L}^S_{PHL}$和 $\mathcal{L}^S_{BCEL}$ 时，这种情况尤为明显。这一结果特别有意义，因为它表明，当实体和关系的语义信息可用时，利用这些信息是有益的，即使最终目标仍然只是提升知识图谱嵌入模型在基于排名的指标上的性能。

此外，研究指出，在损失函数中纳入签名信息对小型关系（B1）的影响最大（表3），无论是在基于排名的指标（平均倒数排名、Hits@10）还是语义正确性（Sem@10）方面都是如此。对于语义有效实体候选池较大的关系，其对Sem@10的影响仍为正面，但有时会以基于排名的指标略有下降为代价。如果后者是唯一的优化目标，那么设计一种自适应训练策略会是合理的：根据当前关系以及作为头实体或尾实体的语义有效候选实体数量，交替使用常规损失函数和签名驱动损失函数。

6.3 关于链接预测中知识图谱嵌入模型预测的评估

需要注意的是，Sem@K 衡量模型根据关系签名预测语义正确三元组的能力，但并不衡量错误性，这与平均倒数排名或 Hits@K不同。可以认为，在电子商务推荐系统等特定用例中，为了提供良好的用户体验，模型既应预测用户期待的物品（高 MRR/Hits@K），也应预测语义有效的物品（高 Sem@K）。在医疗保健、航空航天或安全等其他应用中，用户希望在模型出错时能够察觉，以便接手处理。这对应于高 MRR/Hits@K 和低 Sem@K，从而让错误清晰可见。

我们的实验和这些用例表明，需要同时使用 Hits@K 和 Sem@K 指标，才能根据所考虑应用的需求准确评估模型性能

（例如，电子商务、医疗保健）。这为负三元组的采样方式以及针对不同类型负三元组进行区别处理提供了思路，从而使模型性能的特定方面符合应用需求。例如，根据我们的实验结果，我们可以设想在电子商务应用中对负例采取不同处理方式，以获得较高的 Sem@K；而航空航天应用则需要采取相反的做法。

7 结论

在本研究中，我们聚焦于知识图谱中用于链接预测的主要损失函数。基于负三元组并非都同样有助于学习更优嵌入这一假设，我们将关系签名信息纳入损失函数，根据负三元组相对于关系定义域和值域的语义有效性对其加以区分。随后，我们在原始损失函数和签名驱动损失函数下分别训练了多种知识图谱嵌入模型。在三个具有不同特征的公开知识图谱上的实验表明，所提出的签名驱动损失函数取得了令人鼓舞的结果：在大多数情况下，它们不仅带来了更高的平均倒数排名和 Hits@10 值，还推动知识图谱嵌入模型实现了更高的语义正确性，这一点通过 Sem@10 进行了衡量。这表明，只要能够获得语义信息，就应进一步将其注入损失函数中。未来，我们将研究所提出的损失函数如何容纳其他类型的本体约束，以及字面量节点。

参考文献

1. Ali, M., Berrendorf, M., Hoyt, C.T., Vermue, L., Galkin, M., Sharifzadeh, S., Fischer, A., Tresp, V., Lehmann, J.：揭开黑暗：在统一框架下对知识图谱嵌入模型进行大规模评估。IEEE《模式分析与机器智能汇刊》**44**(12), 8825–8845 (20223124805 21011092021)。https://doi.org/./TPAMI.. Balazevic, I., Allen, C., Hospedales, T.M.：Tucker：用于知识图谱补全的张量分解。载于：2019年自然语言处理经验方法会议与第九届自然语言处理国际联合会议（EMNLP-IJCNLP 2019）会议论文集，中国香港，2019年11月3日至7日。第5184–5193页。计算语言学协会（2019）。https://doi.org/101522 318653119./v/D-. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.：用于建模多关系数据的嵌入平移。载于：神经信息处理系统会议（NeurIPS）。第2787–2795页（2013）。4. Cai, L., Wang, W.Y.：KBGAN：面向知识图谱嵌入的对抗学习。载于：2018年计算语言学协会北美分会会议：人类语言技术（NAACL-HLT 2018）会议论文集，美国路易斯安那州新奥尔良，2018年6月1日至6日，第1卷（长篇论文）。第1470–1480页。计算语言学协会（2018）。https://doi.org/10.18653/v1/n18-1133

5. Cao, Z., Xu, Q., Yang, Z., Huang, Q.：ER：用于知识图谱补全的等变正则器。载于：第三十六届AAAI人工智能会议（AAAI 2022）、第三十四届人工智能创新应用会议（IAAI 2022）、第十二届人工智能教育进展研讨会（EAAI 2022），线上会议，2022年2月22日至3月1日。第5512–5520页。AAAI出版社（2022）。6. Cui, Z., Kapanipathi, P., Talamadupula, K., Gao, T., Ji, Q.：知识图谱中的类型增强关系预测。载于：第三十五届AAAI人工智能会议（AAAI 2021）、第三十三届人工智能创新应用会议（IAAI 2021）、第十一届人工智能教育进展研讨会（EAAI 2021），线上会议，2021年2月2日至9日。第7151–7159页。AAAI出版社（2021）。7. d'Amato, C., Quatraro, N.F., Fanizzi, N.：将背景知识注入嵌入模型，以执行知识图谱上的预测任务。载于：《语义网》——第十八届国际会议（ESWC 2021），线上会议，2021年6月6日至10日，会议论文集。计算机科学讲义，第12731卷，第441–457页。Springer（2021 10 1007 978 3 030 77385 4_26）。8 https://doi.org/./----. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.：卷积二维知识图谱嵌入。载于：第三十二届AAAI人工智能会议（AAAI-18）、第三十届人工智能创新应用会议（IAAI-18）及第八届AAAI人工智能教育进展研讨会（EAAI-18）会议论文集，2018年，美国路易斯安那州新奥尔良。第1811–1818页。AAAI出版社（2018）。9. Ding, B., Wang, Q., Wang, B., Guo, L.：利用简单约束改进知识图谱嵌入。载于：第56届计算语言学协会年会（ACL 2018）会议论文集，澳大利亚墨尔本，2018年7月15日至20日，第1卷：长篇论文。第110–121页。计算语言学协会（20181011 101018653118）。https://doi.org/./v/P-. Guo, S., Wang, Q., Wang, B., Wang, L., Guo, L.：语义平滑的知识图谱嵌入。载于：计算语言学协会年会与亚洲自然语言处理联合会第七届自然语言处理国际联合会议（ACL 2015）第53届会议论文集，中国北京，2015年7月26日至31日，第1卷：长篇论文。第84–94页。计算语言学协会（2015）。https://doi.org/101009 113115115./v/p-. Hubert, N., Monnin, P., Brun, A., Monticolo, D.：用于链接预测的知识图谱嵌入：谨防语义！载于：与第二十一届国际语义网会议（ISWC 2022）同期举行的知识图谱深度学习研讨会（DL4KG 2022）会议论文集。线上会议（2022年10月）。12. Hubert, N., Monnin, P., Brun, A., Monticolo, D.：学习知识图谱嵌入的新策略：推荐场景。载于：知识工程与知识管理——第二十三届国际会议（EKAW 2022），意大利博尔扎诺，2022年9月26日至29日，会议论文集。计算机科学讲义，第13514卷，第66–80页。Springer（2022）。https://doi.org/ 10 1007 978 3 031 17105 5_5 ./----. Hubert, N., Monnin, 13 P., Brun, A., Monticolo, D.：Sem@k：我的知识图谱嵌入模型是否具备语义感知能力？（2023）。14. Jain, N., Tran, T., Gad-Elrab, M.H., Stepanova, D.：通过本体推理改进知识图谱嵌入。载于：《语义网》——ISWC 2021——第二十届

国际语义网会议，ISWC 2021，线上活动，2021年10月24日至28日，会议论文集。计算机科学讲义，第12922卷，第410–426页。Springer（ 2021 10 1007 978 3 030 88361 4_24 ）。
15
https://doi.org/./----. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.：知识图谱综述：表示、获取与应用。IEEE神经网络与学习系统汇刊 **33**3070843 162494514 2022021011092021( ), – ( )。https://doi.org/./TNNLS.. Kazemi, S.M., Poole, D.：用于知识图谱链接预测的简易嵌入。载于：神经信息处理系统进展第31卷：神经信息处理系统年会 2018，NeurIPS 2018，2018年12月3日至8日，加拿大蒙特利尔。第4289–4300页（2018）17. Kotnis, B., Nastase, V.：负采样对知识图谱链接预测影响的分析。arXiv预印本 1708.06816（2017）18. Krompaß, D., Baier, S., Tresp, V.：知识图谱中的类型约束表示学习。载于：语义网——第14届国际语义网会议（ISWC）。第9366卷，第640–655页。Springer（2015）19. Lv, X., Hou, L., Li, J., Liu, Z.：区分知识图谱嵌入中的概念与实例。载于：2018年计算语言学实证方法会议论文集，比利时布鲁塞尔，2018年10月31日至11月4日。第1971–1979页。计算语言学协会（2018）。
https://doi.org/101222 2018653118./v/d-. Minervini, P., Costabello, L., Muñoz, E., Novácek, V., Vandenbussche, P.：通过等价公理和求逆公理对知识图谱嵌入进行正则化。载于：数据库中的机器学习与知识发现——欧洲会议 ECML PKDD 2017，马其顿斯科普里，2017年9月18日至22日，会议论文集，第一部分。计算机科学讲义，第10534卷，第668–683页。Springer（2017）。https://doi.org/ 10 1007 978 3 319 71249 9_40 ./----. Mohamed, S.K.,
21
Muñoz, E., Novacek, V.：关于知识图谱嵌入模型训练。Information **12**12040147 22420211033900( ) ( )。https://doi.org/./info. Mohamed, S.K., Novácek, V., Vandenbussche, P., Muñoz, E.：知识图谱嵌入模型中的损失函数。载于：知识图谱深度学习研讨会（DL4KG2019）论文集，与第16届扩展语义网会议 2019（ESWC 2019）同期举行，斯洛文尼亚波尔托罗日，2019年6月2日。CEUR研讨会论文集，第2377卷，第1–10页。CEUR-WS.org（2019）23. Niu, G., Li, B., Zhang, Y., Pu, S., Li, J.：Autoeter：通过关系感知注意力实现自动化实体类型表示，用于知识图谱嵌入。载于：计算语言学协会成果：EMNLP 2020，线上活动，2020年11月16日至20日。ACL成果，第EMNLP 2020卷，第1172–1181页。计算语言学协会（2020）。
https://doi.org/10105 241865312020./v/.findings-emnlp.. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.：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**(2), 14:1–14:49 (2021) 25. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.：利用图卷积网络对关系数据建模。载于：语义网——第15届国际会议，ESWC 2018，希腊克里特岛伊拉克利翁，2018年6月3日至7日，会议论文集。计算机科学讲义，第10843卷，第593–607页。Springer（ 2018 10 1007 978 3 319 93417 4_38 ）。
26
https://doi.org/./----. Sun, Z., Deng, Z., Nie, J., Tang, J.：RotatE：通过复数空间中的关系旋转进行知识图谱嵌入。载于：第7届国际表征学习会议，ICLR（2019）

27. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.：用于简易链接预测的复数嵌入。载于：第33届国际机器学习会议论文集，ICML。第48卷，第2071–2080页（2016）28. Wang, P., Zhou, J., Liu, Y., Zhou, X.：Transet：结合实体类型的知识图谱嵌入。《Electronics》**10**(12), 1407 (2021) 29. Wang, Q., Mao, Z., Wang, B., Guo, L.：知识图谱嵌入：方法与应用综述。IEEE知识与数据工程汇刊 **29**(12), 2724–2743 (2017) 30. Wang, Z., Zhang, J., Feng, J., Chen, Z.：通过在超平面上平移进行知识图谱嵌入。载于：第二十八届AAAI人工智能会议论文集。第1112–1119页（2014）31. Weyns, M., Bonte, P., Steenwinckel, B., Turck, F.D., Ongenae, F.：知识图谱嵌入的条件约束。载于：知识图谱深度学习研讨会（DL4KG@ISWC）论文集。第2635卷（2020）32. Xie, R., Liu, Z., Sun, M.：结合层级类型的知识图谱表示学习。载于：第二十五届国际人工智能联合会议论文集，IJCAI 2016，美国纽约，2016年7月9日至15日。第2965–2971页。IJCAI/AAAI出版社（2016）33. Yang, B., Yih, W., He, X., Gao, J., Deng, L.：为知识库中的学习与推理嵌入实体和关系。载于：第3届国际表征学习会议，ICLR（2015）34. Zhang, Y., Yao, Q., Shao, Y., Chen, L.：NSCaching：面向知识图谱嵌入的简单高效负采样。载于：第35届IEEE国际数据工程会议，ICDE 2019，中国澳门，2019年4月8日至11日。第614–625页。IEEE（2019）。https://doi.org/10.1109/ICDE.2019.00061

修改版 $\mathcal{L}^{S}_{BCEL}$ 和 $\mathcal{L}^{S}_{PLL}$

表 4报告了使用 $\mathcal{L}^{S'}_{BCEL}$ 和$\mathcal{L}^{S'}_{PLL}$训练的知识图谱嵌入模型取得的结果，其中上标 S' 表示纳入语义信息的不同方式。实际上， $\mathcal{L}^{S'}_{PLL}$ 采用了修改后的标注函数，如 $\mathcal{L}^{S}_{BCEL}$ 中所用，并在论文公式（5）中定义。相反，$\mathcal{L}^{S'}_{BCEL}$ 使用二元（未修改的）标注函数 ℓ，但采用与$\mathcal{L}^{S}_{PLL}$相同的流程：以概率 $\epsilon\%$将语义有效的负三元组视为正例。 $\mathcal{L}^{S}_{BCEL}$、 $\mathcal{L}^{S}_{PLL}$、 $\mathcal{L}^{S'}_{BCEL}$和 $\mathcal{L}^{S'}_{PLL}$的超参数见 https://github.com/nicolas-hbt/semantic-lossfunc/。上述所有损失函数取得的结果见表 4。结果表明，论文提出的签名驱动损失函数表现最佳。

表4. FB15k187、DBpedia77k 和 Yago14k 上基于排名和基于语义的结果。粗体表示在给定指标上表现最佳的模型。后缀 S 和 S'分别表示模型是在损失函数的最佳签名驱动版本（如论文所示）还是最差签名驱动版本（如本文所示）下训练的。

	FB15k187			DBpedia77k			Yago14k		
	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分	平均倒数排名	前10命中率	前10语义得分
ComplEx-S	**.316**	**.476**	**.796**	**.297**	**.409**	**.897**	**.923**	**.931**	**.667**
ComplEx-S'	.227	.384	.777	.252	.350	.918	.907	.930	.603
SimplE-S	**.268**	**.409**	.759	.230	**.302**	**.850**	**.924**	**.927**	**.769**
SimplE-S'	.169	.288	**.827**	.230	.297	.583	.885	.915	.290
ConvE-S	**.283**	**.476**	**.996**	**.283**	**.405**	**.985**	.933	.940	**.997**
ConvE-S'	.271	.472	.975	.273	.383	.935	.933	**.941**	.894
TuckER-S	**.320**	**.522**	**.996**	**.312**	**.421**	**.969**	**.931**	**.943**	**.929**
TuckER-S'	.316	.517	.983	.311	.412	.912	.918	.938	.867
RGCN-S	**.260**	**.415**	**.860**	**.197**	**.320**	**.957**	**.927**	**.934**	**.828**
RGCN-S'	.243	.391	.780	.146	.246	.862	.912	.922	.385

B 桶分析

关系被划分为三个互不相交的桶：分别包含语义有效头实体或尾实体集合较狭窄（B1）、中等（B2）和大型（B3）的关系。阈值由人工设定，用于将给定关系归入相应的桶。这些桶见 https://github.com/nicolas-hbt/semantic-l

论文中报告了在 B1 上取得的结果，而 DBpedia77k、FB15k187 和 Yago14k 上 B2 和 B3 桶的结果见 https://github.com/nicolas-hbt/semantic-lo