**中图分类号：TP181**

**论文编号：10006SY1921104**



硕 士 学 位 论 文

**基于图神经网络的知识图谱**

**表示学习方法研究**

作者姓名 刘希阳

学科专业 软件工程

指导教师 谭火彬 副教授

培养学院 软件学院

**Research of Knowledge Graph Representation Learning Algorithm Baesd on Graph Neural Network**

A Dissertation Submitted for the Degree of Master

**Candidate: Liu Xiyang**

**Supervisor: Assoc. Prof. Tan Huobin**

School of Software

Beihang University, Beijing, China

**中图分类号：TP181**

**论文编号：10006SY1921104**

硕 士 学 位 论 文

**基于图神经网络的知识图谱**

**表示学习方法研究**

作者姓名 刘希阳 申请学位级别 学术硕士

指导教师姓名 谭火彬 职 称 副教授

学科专业 软件工程 研究方向 软件工程

学习时间自 2019 年 9 月 5 日 起至 2022 年 x 月 xx 日止

论文提交日期 2022 年 x 月 xx 日 论文答辩日期 2022 年 x 月 xx 日

学位授予单位 北京航空航天大学 学位授予日期 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：        日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

**摘 要**

知识图谱作为描述现实世界的知识库，在众多领域都得到了广泛的应用。然而知识图谱中的原始数据是文本形式，难以被各类机器学习算法直接利用，同时在知识图谱中存在丰富的语义信息需要挖掘。为了解决上述问题，知识图谱表示学习算法被提出并且受到了很多研究者的关注。它能够学习知识图谱中实体和关系在低维、连续空间下的表示，探索如何建模知识图谱蕴含的信息。近期，图神经网络被应用到知识图谱上用于捕获图结构信息，相较于传统的知识表示学习方法取得了更好的效果。在知识图谱中，关系是将离散的实体链接起来组成一张完整的稀疏网络的关键组成部分。但是已有的基于图神经网络的表示学习算法对于如何使用合适的方法充分学习关系的语义信息没有进行深入的探究。为了解决这一问题，本文基于图神经网络相关理论，提出了关系感知图注意力网络和异质知识图谱图神经网络两种模型。论文的主要工作内容及创新点如下：

（1）针对目前的图神经网络没有充分学习关系语义的问题，本文基于直接引入特定的关系参数，增强模型在参数空间上对关系感知的思想，提出基于关系感知的图注意力网络（Relation Aware Graph ATtention network，RAGAT）。该算法通过定义额外的关系参数改进了目前的四种基于组合的消息构造函数，同时还提出了一种新的基于交叉交互的消息构造函数；之后为了学习关系的方向性，该算法将不同关系划分到不同的方向分组，基于图注意力机制自适应的聚合信息。

（2）针对引入过多的关系参数会引起模型过度参数化的问题，本文基于参数动态生成的思想，让模型所需的关系参数直接从外部网络获取，提出异质知识图谱图神经网络（Heterogeneous Knowledge Graph Neural Network, HKGN）。该算法通过定义所有关系共享的外部网络，直接获得所需的关系参数，从而达到控制参数量的目的。在此基础上，该算法引入关系卷积层增加邻居实体与关系的特征交互，之后基于图卷积操作，聚合来自多关系图的信息。

（3）将提出的RAGAT和HKGN模型在常见的知识图谱Freebase和WordNet上进行实验。在不同的数据集下通过与基线模型比较证明模型的整体性能；利用消融实验验证关键设计的有效性。

**关键词**：知识图谱，知识表示学习算法，图神经网络

**Abstract**

Knowledge graphs have been widely utilized in numerous fields as the knowledge base for describing the real world. However, the original data of knowledge graphs is in textual form and is hard to be directly used by various machine learning algorithms. Besides, knowledge graphs contain rich semantic information that can be mined. The knowledge graph representation learning algorithm is proposed to solve the problems and has aroused considerable research attention. The algorithm can learn the representations of entities and relations of knowledge graphs in low-dimensional, continuous space and explore how to make full use of the information contained in knowledge graphs. Aiming to exploit graph structures of knowledge graphs, a new relation aware graph attention neural network and a novel heterogeneous knowledge graph neural network are proposed based on the theories of graph neural networks. The main contents and innovations of the paper are as follows:

(1) A Relation Aware Graph ATtention network (RAGAT) is proposed to solve the problem that current graph neural networks do not thoroughly learn the relational semantics. The model introduces relation-specific parameters to improve the model’s ability to learn the semantics of relations in parameter spaces. The algorithm improves four current composition-based message construction methods by defining additional relational parameters and a new cross-based message function. Then various relations are divided into distinct directional groups to learn the directionality of relations. The information is aggregated adaptively by using the graph attention mechanism.

(2) A Heterogeneous Knowledge Graph Neural network (HKGN) is proposed to solve the problem that too many relational parameters will lead to the over-parameterization of the model. The model obtains the model's relational parameters from an external network by dynamic parameter generation. The algorithm obtains the required parameters through an external network shared by all relations to control the number of parameters. With this foundation, the algorithm introduces a relational convolution layer to increase the feature interaction between neighbor entities and relations and then aggregates the information from multiple relation subgraphs utilizing graph convolution operation.

(3) The RAGAT and HKGN are evaluated on two standard knowledge graphs: Freebase and WordNet. The model's overall performance is verified by comparing baseline models in different data sets. The effectiveness of key designs is demonstrated by conducting ablation experiments.

**Keywords**: Knowledge Graph, Knowledge Graph Representation, Graph Neural Network

**目 录**

[第一章 绪论 1](#_Toc100257482)

[1.1 研究背景与意义 1](#_Toc100257483)

[1.2 国内外研究现状 2](#_Toc100257484)

[1.2.1 相关研究发展现状 2](#_Toc100257485)

[1.2.2 对比分析 6](#_Toc100257486)

[1.3 研究目标及内容 7](#_Toc100257487)

[1.3.1 研究目标 7](#_Toc100257488)

[1.3.2 研究内容 7](#_Toc100257489)

[1.4 论文组织安排 8](#_Toc100257490)

[第二章 相关理论基础 9](#_Toc100257491)

[2.1 图神经网络 9](#_Toc100257492)

[2.1.1 图神经网络的介绍 9](#_Toc100257493)

[2.1.2 异质图神经网络 13](#_Toc100257494)

[2.2知识图谱表示学习算法 14](#_Toc100257495)

[2.2.1 传统的知识图谱表示学习算法 14](#_Toc100257496)

[2.2.2 基于图神经网络的知识图谱表示学习算法 16](#_Toc100257497)

[2.3 超网络 18](#_Toc100257498)

[2.4 本章小结 19](#_Toc100257499)

[第三章 基于关系感知的图注意力网络 20](#_Toc100257500)

[3.1 现有问题分析 20](#_Toc100257501)

[3.2 RAGAT总体结构 22](#_Toc100257502)

[3.2.1 符号定义 22](#_Toc100257503)

[3.2.2 网络结构 24](#_Toc100257504)

[3.2 RAGAT关键步骤 26](#_Toc100257505)

[3.2.1 基于关系感知的消息构造 26](#_Toc100257506)

[3.2.2 基于注意力的多方向消息聚合 30](#_Toc100257507)

[3.2.3 得分函数的选择 34](#_Toc100257508)

[3.3 本章小结 36](#_Toc100257509)

[第四章 异质知识图谱图神经网络 37](#_Toc100257510)

[4.1 现有问题分析 37](#_Toc100257511)

[4.2 HKGN模型结构 39](#_Toc100257512)

[4.2.1 符号定义 39](#_Toc100257513)

[4.2.2超网络的引入 40](#_Toc100257514)

[4.2.3 消息构造方法 42](#_Toc100257515)

[4.2.4 多关系图聚合方法 44](#_Toc100257516)

[4.4 本章小结 47](#_Toc100257517)

[第五章 实验与验证 48](#_Toc100257518)

[5.1 实验设计 48](#_Toc100257519)

[5.1.1 数据集介绍 48](#_Toc100257520)

[5.1.2 评估策略 48](#_Toc100257521)

[5.1.3 实验环境 50](#_Toc100257522)

[5.2 RAGAT实验结果分析 50](#_Toc100257523)

[5.2.1 超参数设置 50](#_Toc100257524)

[5.2.2 对比方法 50](#_Toc100257525)

[5.2.3 整体性能分析 51](#_Toc100257526)

[5.2.4 关键设计分析 52](#_Toc100257527)

[5.3 HKGN实验结果分析 55](#_Toc100257528)

[5.3.1 超参数设置 55](#_Toc100257529)

[5.3.2 对比方法 56](#_Toc100257530)

[5.3.3 整体性能分析 56](#_Toc100257531)

[5.3.4 关键设计分析 58](#_Toc100257532)

[5.3.5 超参数影响分析 61](#_Toc100257533)

[5.4 本章小结 63](#_Toc100257534)

[总结与展望 64](#_Toc100257535)

[参考文献 66](#_Toc100257536)

[攻读硕士期间取得的学术成果 74](#_Toc100257537)

[致谢 75](#_Toc100257538)

**图 目 录**

[图1 研究内容示意图 7](#_Toc100259290)

[图2 HyperRNN示意图 19](#_Toc100259291)

[图3 多关系知识图谱示例 20](#_Toc100259292)

[图4 关系的方向性示例 22](#_Toc100259293)

[图5 RAGAT整体结构 25](#_Toc100259294)

[图6 基于关系感知的消息构造函数 26](#_Toc100259295)

[图7 针对CrossE的一种代数角度解释 29](#_Toc100259296)

[图8 基于注意力的多方向消息聚合函数 31](#_Toc100259297)

[图9 RAGAT中关系的分层结构 34](#_Toc100259298)

[图10 InteractE解码器结构 35](#_Toc100259299)

[图11 MAGNN方法示意图 38](#_Toc100259300)

[图12 使用SVD对关系矩阵进行分解 41](#_Toc100259301)

[图13 引入超网络的消息构造方法 42](#_Toc100259302)

[图14 一般的线性转换操作 43](#_Toc100259303)

[图15 关系卷积操作 43](#_Toc100259304)

[图16 HKGN整体结构 45](#_Toc100259305)

[图17 引入关系感知参数在FB15k-237上的MRR结果 53](#_Toc100259306)

[图18 引入关系感知参数在FB15k-237上的Hit@1结果 53](#_Toc100259307)

[图19 引入关系感知参数在FB15k-237上的Hit@3结果 54](#_Toc100259308)

[图20 引入关系感知参数在FB15k-237上的Hit@10结果 54](#_Toc100259309)

[图21 关系卷积层对FB15k-237中不同度实体的影响 60](#_Toc100259310)

[图22 关系卷积层对WN18RR中不同度实体的影响 60](#_Toc100259311)

[图23 超参数对模型性能的影响 62](#_Toc100259312)

[图24 超参数对模型性能的影响 62](#_Toc100259313)

[图25 超参数对模型性能的影响 62](#_Toc100259314)

**表 目 录**

[表1 现有的知识图谱表示学习算法优缺点总结 6](#_Toc100254224)

[表2 目前图神经网络知识表示学习算法所采用的消息函数 20](#_Toc100254225)

[表3 RAGAT中的符号定义 23](#_Toc100254226)

[表4 五种关系感知函数变体 29](#_Toc100254227)

[表5 异质图神经网络采用的数据集 38](#_Toc100254228)

[表6 HKGN中的符号定义 39](#_Toc100254229)

[表7 数据集统计信息 48](#_Toc100254230)

[表8 实验环境配置 50](#_Toc100254231)

[表9 RAGAT超参数设置 51](#_Toc100254232)

[表10 RAGAT整体性能 52](#_Toc100254233)

[表11 关系感知消息构造函数变体的实验结果 52](#_Toc100254234)

[表12 消融实验结果 55](#_Toc100254235)

[表13 HKGN超参设置 55](#_Toc100254236)

[表14 HKGN整体性能 57](#_Toc100254237)

[表15 FB15k-237关系分类统计 58](#_Toc100254238)

[表16 HKGN关系分类的实验结果 58](#_Toc100254239)

[表17 HKGN无关系卷积层的实验结果 59](#_Toc100254240)

[表18 数据集实体度分布的统计 59](#_Toc100254241)

[表19 HKGN超网络的消融实验结果 61](#_Toc100254242)

# 第一章 绪论

## 1.1 研究背景与意义

本文来源于国家重点研发项目“科技大数据理论与技术研究”，该项目旨在对科技大数据的理论和技术进行研究。实验室参与的课题是面向服务的知识服务构件开发，课题目标是研究数据的统一语义表示，分析与挖掘资源间的知识语义关联关系，研究针对数据的聚类、搜索、推荐等应用算法，并且在研究的基础上，实现分布式结构的知识服务构件的开发。本文所研究的基于图神经网络的知识图谱学习算法研究，目的在于研究数据的语义表示，为上层科技服务构件的开发提供支撑。

在开发应用算法时存在两个问题，首先原始的科技大数据形式无法被各类服务算法直接利用。学者、论文、专利等科技资源都是以图的形式存储，来自不同来源的数据共同组成了描述现实的知识图谱（Knowledge Graph）。原始形式的知识图谱是文本形式的三元组，无法被各类应用算法例如推荐系统、智能问答系统等直接利用。因此，在知识图谱的原始形式与基于知识图谱的应用算法需要的数据形式之间存在差异。

其次，在知识图谱中存在丰富的语义信息，比如描述知识图谱的文本信息、知识图谱的结构信息以及实体关系的链接信息等，如何充分挖掘知识图谱中丰富的语义信息从而能够更好的支持应用算法的开发也是一个需要解决的问题。

知识图谱表示学习（Knowledge Graph Representation Learning）是解决上面两个问题最直接的手段[1]。知识图谱表示学习方法的核心思想是将知识图谱中的所有实体与关系表示为连续的低维空间中的嵌入（Embedding），学习到的嵌入可以很容易的被其它的机器学习、深度学习等方法所利用。比如知识图谱补全（Knowledge Graph Completion）[2]、节点分类（Node Classification）[3]、推荐系统（Recommender System）[4]等。

目前的知识表示学习方法大多是首先以某种形式，将实体和关系表示为某个连续向量空间下的嵌入，然后通过定义某个得分函数（Scoring Function）预测知识图谱中事实存在的概率。通过概率得分和实际存在的事实进行对比，最大化对于已有事实存在的预测概率来更新实体和关系的嵌入表示。除了单纯的利用知识图谱中的事实，为了获取更丰富的信息，也有很多的研究者关注如何利用来自其它源的信息，比如文本描述，关系路径，逻辑规则等。本文讨论的所有知识表示学习方法都是只使用知识图谱中本身包含的事实的模型。

基于图神经网络（Graph Neural Network）的知识图谱表示学习算法能够充分挖掘知识图谱中的图结构信息，属于该领域最新的研究方向。但是，目前的基于图神经网络的知识表示学习算法在消息构造过程中，很多最新的方法对知识图谱中关系（Relation）的语义信息建模能力不足；但在另一方面，如果单纯的扩大模型建模粒度，又会引入过多的参数，增加模型复杂度，提高模型训练成本。针对上述问题，本文首先尝试使用一种统一的基于图注意力机制的框架，综合了目前出现的建模知识图谱实体与关系的消息构造方法，然后引入关系特定参数，尝试探究单纯的增加关系参数能否为模型效果带来提升。之后，为了解决模型关系参数可能随着关系数量增加而急剧增多的问题，本文提出了另一种新的异质知识图谱图神经网络模型，在引入超网络控制模型参数量的基础上，通过新增关系卷积层而进一步提升了模型对于关系语义的捕获能力。本文研究的知识图谱表示学习算法提供了一种学习知识图谱统一表示的途径，能够将原始的知识图谱形式转换为便于被各类上层应用算法所利用的表示形式，从而为科技大数据服务构件的开发提供了理论支撑。

## 1.2 国内外研究现状

### 1.2.1 相关研究发展现状

虽然知识图谱这个词语早在1972年[5]就已经在文献中出现，但是直至2012年谷歌知识图谱[6]出现才赋予了它在当代表示的含义[7]。在此之后，一大批的知识图谱出现，包括Amazon[8]，eBay[9]，Facebook[10]等。知识图谱的核心思想是，使用简单的三元组存储数据，利用图来表示事物之间的联系。相比于关系型模型或者非关系型数据等传统方式，使用图来表示数据的好处在于，首先图结构符合人类的几何直觉，也更加符合现实的情况，融合了知识图谱的方法，往往能够比传统的方法提供更强的解释性。同时，由于图结构具有更强的灵活性，多种领域的数据都可以用图来表示，便于融合不同领域来源的数据。更重要的是随着时间的推移，新的数据可以随时补充到图中。知识图谱已经在搜索系统、对话系统、推荐系统等各领域都得到了广泛的应用。

在本文讨论的知识图谱属于最常见的三元组形式，从图的角度来看是一个带有标签边的有向图，由实体（entity）和关系（relation）组成，实体对应于现实世界的具体事物，关系对应于事物之间的联系，属于二元关系。在这种情况下的知识图谱具体表现形式是三元组<头实体，关系，尾实体>，这样的一个具体的三元组也被叫做事实（fact）。

为了充分挖掘知识图谱中蕴含的语义信息，将原始的知识图谱三元组文本形式转化为便于被其它机器学习或者深度学习算法直接使用的数据形式，知识图谱表示学习算法受到了大量的关注。目前一般的算法核心思想是首先将实体和关系表示为某个连续向量空间下的嵌入（Embedding），然后利用某个得分函数（scoring function）预测知识图谱中事实存在的概率。因此，知识图谱表示学习算法也叫知识图谱嵌入算法（Knowledge Graph Embedding）。

根据目前对于知识图谱表示学习算法的研究，按照不同方法的原理和结构，可以分为四大类别：基于翻译的表示学习算法，基于张量分解的表示学习算法和基于神经网络的表示学习算法和融合多源信息的表示学习算法。

基于翻译的知识表示学习方法相对于其它的方法来说是最早，最简单，也是最经典的方法，这一系列方法核心思想的起源是2013年提出的TransE[1]方法，它将关系视为在嵌入空间内实体到实体的翻译。在此之后，衍生除了一系列的Trans方法：TransH[11]，TransR[12]，TransD[13]，TransA[14]，HAKE[15]等。

基于翻译的知识表示学习方法的优点在于方法简单，容易理解，数学的可解释性强，容易与其它的学习方法结合。缺点在于它是一个浅层的结构，编码知识图谱中蕴含信息的能力相对深度的学习方法来说更弱。

基于张量分解的表示学习方法中最经典的方法是RESCAL[16]方法，它将整个知识图谱表示成为一个高维稀疏的张量。假设实体数量是，关系数量为，那么知识图谱为一个的张量，第行列深度为的值为1表示实体与实体之间存在关系；其次它的得分函数不是计算某种距离，而是直接计算边是否存在的得分。这种得分函数叫做语义匹配得分。

在此之后，基于RESCAL[16]的思想，有很多其它效果更好的模型提出，包括DistMult[17]，HolE[18]，ComplEx[19]，RotatE[20]，CrossE[21]等。其中DistMult[17]将RESCAL[16]的矩阵参数简化为了对角矩阵，在取得更好的效果的同时降低了计算复杂度。ComplEx[19]方法将表示学习算法从实数域拓展到复数域。

基于张量分解的表示学习方法的优点在于利用张量分解等数学方法进行学习，解释性强，能够学习到实体和关系的双线性关系。缺点在于它们统一处理所有知识图谱中的事实，无法充分学习图结构信息，同时属于浅层的模型，模型表达能力比较弱。

基于神经网络的学习方法是目前研究的主流，它使用神经网络学习知识图谱中的知识，按照模型的结构，可进一步大致分为：基于多层神经网络、基于循环卷积网络和基于卷积神经网络。

使用多层神经网络的表示学习方法原理非常直接，直接使用多层的神经网络去拟合知识图谱，输入层是三元组，输出是该三元组存在的概率。这一类型的方法包括SME[22]，NTN[23]，MLP[24]等。这一类型的方法网络结构较为简单，效果比起之前的浅层方法效果更好，但是没有充分结合知识图谱的特点，同时由于知识图谱的数据量很大，很容易过度参数化，可解释性也较差。

使用卷积神经网络的表示学习方法是将三元组的嵌入拼接成张量的形式，然后构造卷积核进行卷积操作，之后经过神经网络层，输出得分。这一类型的方法包括ConvE[25]，ConvKB[26]，ConvR[27]，CapsE[28]，InteractE[29]等。最经典的是ConvE[25]方法，该方法首先通过直接堆积实体和关系的表示重组为矩阵，之后在这个特征矩阵上进行卷积操作，随后进行向量化，最终经过一个全连接层，利用得到的输出与其它目标实体的内积作为预测概率。

以上的ConvE[25]方法的框架在ConvR[27]，InteractE[29]等方法上得到了进一步的应用，ConvE[25]方法的评估策略也成为了一种新的评估策略。ConvR[27]直接使用关系的嵌入来构造卷积核，从而减少了引入的参数。InteractE[29]将ConvE[25]方法的矩阵修改为棋盘状分布，使用循环卷积代替常规的卷积操作。

使用卷积神经网络应用到知识图谱上在结果上要优于简单的多层神经网络，这是因卷积操作可以捕获实体嵌入和关系嵌入之间的高维联系。但是单纯的卷积神经网络无法捕获图的结构，比如它无法感知节点和邻居节点这样的结构。

基于图神经网络的知识图谱表示学习算法属于知识表示学习领域目前最新的研究方向。传统的知识图谱表示学习方法主要考虑如何在单纯的三元组上进行学习，从而忽略了三元组之间的联系，忽略了知识图谱本身是一个带标签的有向图，无法利用图结构进行学习。实际上，随着图神经网络的提出，基于图结构的学习已经在众多算法领域例如图分类、链路预测等方面证明了图结构能够提供的丰富的信息。在基于图神经网络的表示学习算法这一方向上的研究一方面需要考虑图神经网络的特点，另一方面需要结合知识图谱本身的特点。R-GCN[30]是首个在知识图谱上应用图卷积神经网络的模型，它本身是一个编码器+解码器的结构，编码器负责利用图神经网络建模图结构信息，解码器则是直接利用传统的知识图谱表示学习方法进行评估。后续的研究基本是沿着R-GCN[30]提出的编码器-解码器结构进行。SACN[31]方法将GCN[32]扩展至带权值的多个子图聚合的形式；VR-GCN[33]结合TransE[1]的思想将不同方向的信息进行不同的翻译处理；TransGCN[34]设计了实数域和复数域下两种基于翻译的思想的编码器。

融合多源信息的表示学习方法是为了获取更丰富的信息，考虑利用更多的额外信息，比如文本描述，关系路径，逻辑规则等。之前的知识图谱嵌入方法（除基于图神经网络的方法）基本都是在研究如何能够仅仅利用单纯的知识图谱三元组进行表示学习。事实上，在知识图谱当中还存在很多不同来源的信息，对这些额外来源的信息进行建模，可以增强知识表示方法的学习效果。

SSE[35]方法尝试利用实体分类信息，它认为属于同一语义分类的实体在向量空间当中应该更加接近，使用局部线性嵌入和拉普拉斯特征映射两种流型学习算法进行建模。

PTransE[36]方法在TransE[1]的基础上，加入了关系路径，即多跳连续的三元组。一方面，它通过定义路径约束资源分配算法评估关系路径的可靠程度；另一方面它使用关系路径中存在的多跳关系表示的组合作为关系路径的表示。最终得到了更好的效果。

在知识图谱中，实体和关系都是由对应的文本描述的，比如关系性别，“性别”这个词应该本身是有特定的文本语义的。在NTN[23]方法中，使用额外的新闻语料库得到词汇的表示，然后用词汇向量的平均值初始化实体的表示。在DKRL[37]方法中，学习得到实体基于结构的表示和基于文本描述的表示，两种表示结合起来进行预测。在实验中发现由于学习到了基于文本描述的表示，对于从来没有训练过的实体也表现出了较好的预测效果。

通常在知识表示学习过程中，会将实体属性也看做一般的关系处理，比如性别、年龄等。但是在某些情况下，这样的统一处理可能并不合适。有研究人员将RESCAL[16]进行了改进，对实体属性进行单独建模处理[38]。在SACN[31]方法中，引入了实体属性来增强实体嵌入的表达能力。

ILP[39]方法将一般的基于嵌入的方法和规则结合起来，使用整数线性规划问题进行学习，将一般的知识图谱嵌入方法的优化作为目标函数，将一系列的规则作为约束。同时，在ILP[39]方法中，还使用松弛方式实现了知识图谱中噪声的探测。

在实际当中，知识图谱是会不断更新的，已有的知识会存在过时的情况，新的实体和关系会被添加到知识图谱中，因此实际的知识图谱应该是一个动态知识图。而之前的方法都是将知识图谱看做静态的图，没有考虑时序信息。有研究人员在传统的基于翻译的知识图谱嵌入方法上进行改进[40]，三元组改为增加了时间维度的四元组，将TransE、TransH和TransR改造为tTransE、tTransH和tTransR。

### 1.2.2 对比分析

通过对比几种主流的知识图谱表示学习方法，本文总结了各种算法的优势与不足，如表1所示。在科技大数据研究课题中，由于各类科技资源共同组成了一张有向图，图的结构能够提供丰富的信息，实体与关系、实体与邻居实体、甚至于关系之间都存在可以通过图结构相连。同时，图神经网络还可以充分结合之前的方法共同提升学习效果，比如使用卷积神经网络作为解码器。因此，本文拟基于图神经网络探索如何学习得到具有更多信息的知识表示。

表1 现有的知识图谱表示学习算法优缺点总结

|  |  |  |
| --- | --- | --- |
| 技术类型 | 优点 | 缺点 |
| 基于翻译 | 结构简单，模型复杂度低，易于应用 | 模型表达能力弱，只单纯的考虑到单个三元组的信息 |
| 基于张量分解 | 模型易于应用，模型复杂度低，多个张量相乘可以隐式表示规则的连接 | 将整个图谱表示为0/1值的张量，丢失很多信息，表达能力受限 |
| 基于多层神经网络 | 比浅层的模型表达能力更强 | 容易过拟合，受限于嵌入的大小 |
| 基于卷积神经网络 | 能够建模实体和关系之间的非线性的联系，减少参数规模，更适用于大规模知识图谱 | 没有建模知识图谱内部的结构、外部文本等非常有用的信息 |
| 基于图神经网络 | 能够充分利用知识图谱的图结构，模型的信息编码能力更强，取得了更好的结果 | 模型复杂度高，很难直接应用到大规模知识图谱 |
| 融合多源信息 | 能够融合不同来源的信息，与现有的各种方法结合，丰富知识图谱表示学习的效果 | 融合多源信息通常需要额外的资源（文本、属性等） |

## 1.3 研究目标及内容

### 1.3.1 研究目标

本文的研究目标是在目前基于图神经网络的知识图谱表示学习算法基础上，针对知识图谱中关系蕴含丰富语义信息的特点，研究如何设计合理的模型以充分捕获知识图谱的图结构信息。本文所提出的算法能够在利用知识图谱图结构的基础上，对关系进行更细粒度的建模，让实体和关系的低维嵌入能够表达更多的原始信息，最终为科技大数据项目上层应用算法的开发提供基础的理论支撑。

### 1.3.2 研究内容

本文的研究内容如图1所示。为了实现研究目标，本文基于图神经网络的基本理论，结合知识图谱的特点，开展的主要研究内容包括以下几方面：

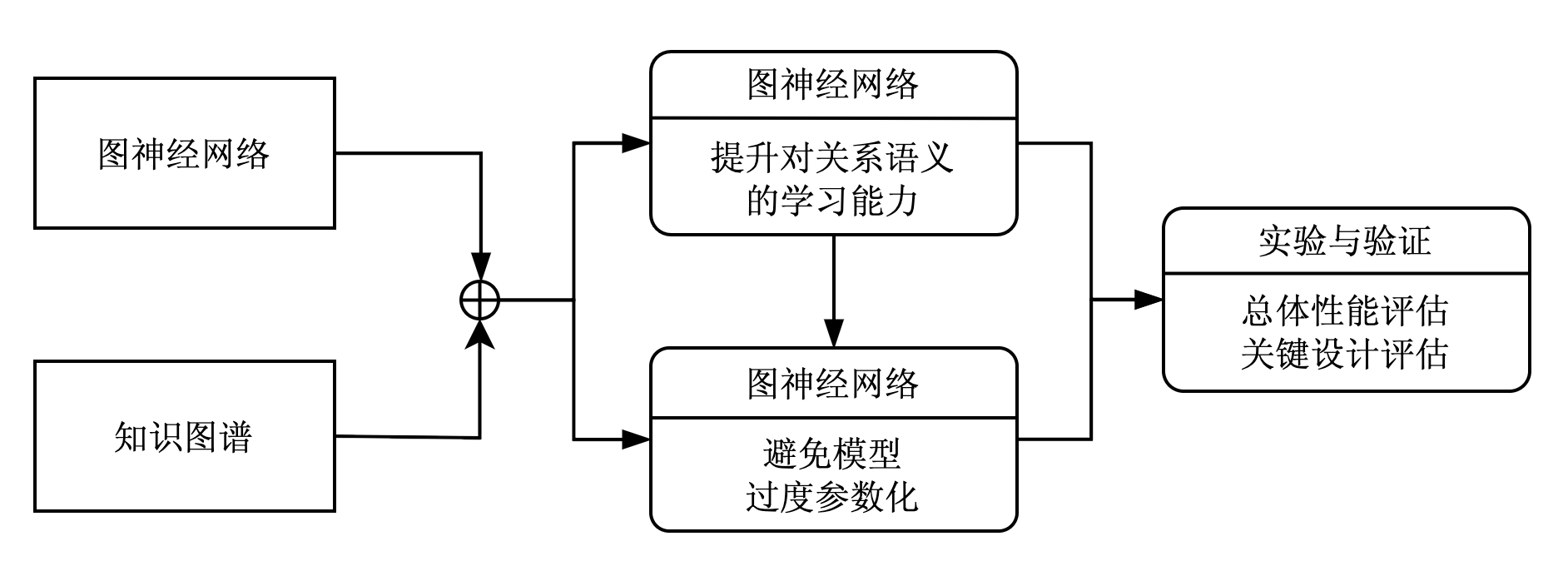


图1 研究内容示意图

（1）模型对关系语义的建模策略

目前出现的基于图神经网络的表示学习方法通常将关系和实体表示为维度等长的嵌入向量，限制了模型对关系语义的建模能力。本文的主要研究内容之一是在图神经网络的关键阶段，设计合适的建模策略学习知识图谱中的关系信息，实现对关系的细粒度建模。

（2）模型过度参数化问题的解决方案

由于知识图谱中可能存在成百甚至上千的关系，因此设计过于复杂的关系建模策略可能会导致过度参数化问题，增大模型训练难度。因此，本文计划研究设计合适的方案，满足在不简化模型结构的同时，控制和关系数量相关的网络参数量，缓解过度参数化问题。

（3）模型的实验与验证

在实现完整的知识图谱表示学习算法之后，通过在标准公开数据集上与基线模型对比，证明本文提出的模型的整体性能；并且通过设计合适的消融实验，验证模型关键设计的有效性。

## 1.4 论文组织安排

本论文内容总共分为五章和总结与展望，各章节内容安排组织如下：

第一章 作为论文的开篇，主要介绍研究背景与意义，介绍目前国内外相关领域的研究现状，对比分析了不同类别方法的优缺点；之后，描述论文的研究目标以及研究内容；最后，介绍论文的整体组织安排。

第二章 介绍本文涉及的相关理论基础。首先介绍了图神经网络的基本概念与一般过程，特别介绍了最新出现的异质图神经网络。其次，描述了不同分类的知识图谱表示学习算法，包括其核心思想以及数学公式；最后，介绍了超网络的基本概念以及其在不同领域的应用。

第三章 详细叙述提出的基于关系感知的图注意力网络。首先总结目前常用方法存在的问题；随后，给出符号定义和模型的总体结构；最后，对关键设计进行了说明，包括基于关系感知的消息构造函数、基于注意力的多方向聚合函数和不同解码器的选择。

第四章 详细介绍提出的异质知识图谱图神经网络。首先指出第三章提出的方法以及目前的异质图神经网络都存在的共性问题；之后，介绍模型的总体结构设计，包括如何引入超网络、增加了关系卷积层的消息构造函数以及多关系图的聚合函数。

第五章 详细说明本文提出的解决方案对应的实验部分。首先，详细描述了实验的整体设计，包括数据集的选择、模型的评估策略、实验环境等基本情况；接下来，分别对第三章和第四章中提出的方法进行实验，包括整体性能对比、关键设计分析、超参设置等方面。

总结与展望 对论文提出的关系感知注意力网络和异质知识图谱图神经网络的研究过程、结构设计以及验证过程进行总结回顾，提出进一步的研究方向和展望。

# 第二章 相关理论基础

本章详细描述本文涉及到的关键技术和基础理论，首先介绍图神经网络的基本概念与关键步骤，并介绍异质图神经网络的经典方法；之后说明传统的知识图谱表示学习算法和基于图神经网络的表示学习算法的核心思想与优缺点；最后介绍超网络的基本概念和其在不同领域的应用情况。

## 2.1 图神经网络

### 2.1.1 图神经网络的介绍

2017年提出的图卷积神经网络GCN[32]将之前基于谱空间的图神经网络中的图卷积算子简化为简单的深度学习线性变换，图神经网络迅速成为了研究热点，有关于图神经网络的应用在推荐系统，对话系统，知识图谱等领域都有大量的论文出现。GCN是目前各种最新图神经网络的核心，因此本节将首先详细介绍GCN的基本思想。

为了能够利用图结构进行学习，首先我们需要采取某种方式表示图。一个图的常用表达形式是，是节点（vertices）的集合，是边（edge）的集合。如果存在边，那么可以被称作的邻居，可以说节点和是邻接的（adjacent）。

图有几种不同的代数表达形式：

邻接矩阵（Adjacency matrix）：对于一个简单图，邻接矩阵定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

度矩阵（Degree matrix）：度矩阵是一个对角矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

拉普拉斯矩阵（Laplacian matrix）：拉普拉斯矩阵定义为度矩阵减去邻接矩阵，：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

对称归一化拉普拉斯矩阵（Symmetric normalized Laplacian）：将上面的拉普拉斯矩阵归一化：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

最终得到了归一化拉普拉斯矩阵，矩阵中的元素定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

接下来我们讨论的图是无向图，那么拉普拉斯矩阵是对称矩阵，归一化之后的图拉普拉斯矩阵是半正定对称矩阵。

基于归一化拉普拉斯矩阵，我们可以将它对角化，，其中是特征向量矩阵，是特征值。这一对角化实对称矩阵的操作在线性代数中叫做谱分解，称作谱（spectrum）。实对称矩阵的特征向量一定是正交的（orthogonal），特征值一定是实数。特征向量构成了谱空间的基。在图的信号处理中，一个图信号（graph signal）定义为，每个节点都有一个对应的数值，这一图信号也叫做节点的特征，可以拥有多个特征，那么把多个图信号按列排序就得到了一个图的特征矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示第个节点的特征，表示第个节点的第个特征。

在介绍GCN之前，需要介绍首个定义在图上进行的卷积操作原理。

在图信号处理中，模仿一般的傅里叶变换定义了图傅里叶变换（graph Fourier transform），就是将处于原来特征空间下的图信号投影转换到谱空间中，

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

傅里叶逆变换定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

由于图不满足平移不变性，我们希望直接在特征空间下定义图的卷积算子比较困难，是输入的图信号（signal），是过滤器（filter）。但利用卷积定理，信号卷积的傅里叶变换等价于信号傅里叶变换的乘积。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

那么图卷积算子可以定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

这样我们无需考虑原特征空间中的卷积，只需要先将信号转换为谱空间中的信号，在谱空间中与谱空间中的过滤器做乘法，最后利用傅里叶逆变换转换到原来的空间中，就实现了图卷积。

更具体的图卷积算子定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是一个向量，可以看做是对角矩阵，最后谱卷积算子定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

所有基于谱空间的图卷积方法都是在尝试寻找更合适的。

Spectral CNN[43]（Spectral Convolutional Neural Network）直接将作为一系列可以学习的参数，得到了下面的卷积公式：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

这里仍然需要计算特征向量，计算代价很大。

之后切比雪夫网络[44]（Chebyshev Spectral CNN，ChebNet）对Spectral CNN进行简化，使用切比雪夫多项式简化，令，其中得到了下面的卷积公式：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，。这样ChebNet不需要再计算特性向量。

GCN[32]在ChebNet的基础上，进一步将K阶多项式限制到了1阶，假设得到了新的卷积算子：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

由于半监督图节点分类学习的数据量少，令0阶系数和1阶系数相同，减小学习的参数量。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

最终提出了具有多层结构的图卷积神经网络（GCN），每一层使用的卷积函数定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中和是区别于之前的度矩阵和邻接矩阵，

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

使用上面公示的原因是GCN在实验中发现，如果使用原来的形式会导致训练的不稳定性以及梯度消失/爆炸的问题，因此重新归一化。

GCN用来做半监督的节点分类任务，在实现的时候使用了两层GCN，最后经过Softmax输出预测值。损失函数使用交叉熵。GCN的一个缺点是它只能处理无向图，无法处理知识图谱这样的有向带标签的图。

在GCN之后，有众多的图神经网络被提出，但是它们本质上都没有脱离GCN的模式——消息传递机制[45]。本文所提出的新方法同样属于消息传递机制，主要的改进点是在消息传递的两个关键阶段——消息构造与消息聚合，因此有必要对这两个阶段进行详细说明。

消息传递网络（MPNN）[45]提出在图神经网络中存在信息传递的两个关键过程：消息构造（Message Construction）和节点更新（Vertex Updating）。消息构造函数用来建模学习单个邻居的信息，即从邻居节点的特征中导出隐含的中间状态。具体到知识图谱上，就是需要对于图中的一个节点，是邻居信息编码方法，假设是节点的一个邻居，输入，邻居信息编码方法输出某种特定的数值形式，向量或者矩阵等，记作，方法在图神经网络中被称为消息构造函数（Message Construction Function），包含了单个邻居的信息，被称作消息（Message）。数学形式定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

图神经网络的另一种关键设计是节点更新函数，该函数利用聚合来的消息表示更新原有节点的表示，在之后的很多论文中直接使用消息聚合（Message Aggregation）代替这一过程，本文也采用了这样的描述。具体的说，在获得了对于单个邻居的信息表示后，为了实现对于图结构信息的编码，对于节点需要聚合其周围邻居的信息，即消息聚合。消息聚合方法是决定基于图神经网络的算法能否准确捕获图结构信息的关键因素。数学形式定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

### 2.1.2 异质图神经网络

异质图神经网络是最近兴起的用于处理异质图（heterogeneous graph）的图神经网络模型，比如引文网络、商品-评论网络等。之前介绍的GCN[32]、GAT[46]等方法对全图进行相同的图卷积操作，图卷积核相当于是用于导出全局的特征，生成更新后的图表示。这样的设计策略背后隐含的假设是图本身是一个同质图（homogeneous graph），也就是说一个图上的顶点、边等元素都来源于同一领域，拥有相同的特征。但是现实世界当中绝大多数的图都是异质图，即顶点和边都可能来自于不同的领域。以一个引文网络为例，顶点论文和顶点作者很明显是来源于不同的领域，特征是不相等的，在这种情况下，GCN等同质图神经网络就无法较好的处理图的信息。

HAN[47]方法，尝试在基于元路径的图上，设计了节点层次和语义层次的两层注意力机制进行异质图的信息聚合，节点层注意力首先根据节点类型，将不同类型的节点投影到相同的隐空间下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

然后计算一个元路径下所有邻居顶点的权重，根据注意力进行消息聚合：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

HAN[47]在语义层通过计算各个顶点在不同元路径图下的聚合结果，计算不同元路径的权重在全局意义下的权重：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

最后得到聚合结果的输出：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

HetGNN[48]方法通过随机游走采样节点，之后根据节点类型进行节点特征的转换；HGT[49]方法设计了一种基于Transformer的注意力机制，实现了在大规模异质图上的学习，表示出较好的结果；HGSL[50]是首个为异质图神经网络寻最优图结构的方法，它不是直接在原始图上进行学习，而是进行三种图的融合：特征相似图、特征传播图和语义图，在这种学习计算得到的图基础上，再进行图卷积的操作。

以上的异质图神经网络比起原来的GCN等方法能够更好的处理具有更强异质性的图。但是经过调研发现，目前的异质图神经网络局限在处理边类型数量较少的图上，而无法应用与知识图谱这种能够拥有成百上千的关系类型的异质图。具体问题在4.1小节中进行了详细介绍。

## 2.2知识图谱表示学习算法

### 2.2.1 传统的知识图谱表示学习算法

知识图谱表示学习算法是将原始的知识图谱数据转换为数值的表示形式。目前这一系列算法的核心思想是首先将实体和关系表示为某个连续向量空间下的嵌入，然后利用某个得分函数预测知识图谱中事实存在的概率。接下来主要详细描述在本文中利用到的几种知识图谱表示学习算法。在之后的描述中，知识图谱三元组表示为，对应的嵌入表示为。

2013年提出的TransE[1]方法是基于翻译的知识表示学习方法的起源，它的核心思想在1.2国内外研究现状小节中已经进行了阐述。TransE将关系视为在嵌入空间内的翻译，如果三元组事实存在，那么头实体的嵌入加上关系的嵌入应该靠近尾实体的嵌入，即公式：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

这一思想主要来源于知识图谱中各个事实组成的层级结构。在树状结构下面，某个节点应该是处在最靠近它的父节点的位置，兄弟节点之间应该距离较近，位置类似，孙子节点相比子节点应该距离更远。这样在连续的向量空间下，新的嵌入组成的结构还是保留了原来知识图谱中的层级结构。

TransE的训练和评估过程也是后续很多方法跟随的经典方法。首先TransE会首先将所有的实体和关系都表示为相同维度大小的向量，然后在训练过程中，对于每一个实际存在的事实，TransE会进行通过替换头实体、关系、尾实体获得实际不存在的负样本，然后对于所有的三元组，计算和之间的距离。向量之间距离的衡量可以是平方和：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

TransE期望正样本计算得到的距离小，负样本计算得到的距离大。

TransE的思想很简单也符合直觉，非常容易实现，参数少，能够很容易的和其它的方法结合起来。TransE的负采样训练方法和评估策略也成为了之后众多知识表示学习方法的训练方式和评估标准。但是TransE方法也存在它的缺点，最重要的一点是它难以很好的学习一对多，多对一和多对多的关系。比如张三导演了三部不同的电影，导演这个关系是一个一对多的关系。利用TransE的方法，尽管这三部电影都是完全不同的实体，TransE会把三部不同的电影都学习为相近的嵌入，很难区分它们之间的区别。为了解决这个问题，才有之后的一系列基于翻译的方法的提出。

RESCAL[16]方法是基于张量分解的经典方法，它首先根据不同的关系把整个知识图谱表示为三维张量。张量的值为0表示该三元组不存在，为1表示该三元组存在。之后，所有的实体被表示为一维向量，关系被表示为二维矩阵。然后对于三元组计算根据下面的公式该事实存在的概率：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

为了降低RESCAL方法的复杂度，DistMult[17]将简化为了对角矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

HolE[18]方法提出使用循环相关（Circular Correlation）的方式学习知识图谱表示的方法，循环相关操作可以看做是对于张量积的压缩，具体操作为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

循环相关操作具体定义是：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

ConvE[25]模型是使用卷积神经网络的经典表示学习方法。它首先通过直接堆积实体和的一维的嵌入进行转换为2维的形式：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

之后进行卷积操作：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

然后变回一维矩阵并且过一个全连接层：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

最后和目标实体的嵌入相乘，就得到了得分：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

### 2.2.2 基于图神经网络的知识图谱表示学习算法

R-GCN[30]是第一个将图神经网络拓展到知识图谱领域的方法。R-GCN的主要思想是“一个实体很多的信息隐藏在它的邻居内”。它设计了一个编码器，可以和其它作为解码器的张量分解模型结合。具体而言，R-GCN使用GCN[32]方法作为编码器，针对链接预测任务使用DistMult[17]方法作为解码器，对于实体分类任务则直接使用Softmax函数。

一般的GCN的形式可以定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

可以是神经网络结构，也可以是简单的线性转换，。

基于以上的原理，R-GCN设计了如下的传播层：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

公式中的可以为，即某个关系r的邻居的数量。

最后，对于链接预测任务，R-GCN使用DistMult方法作为解码器。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

训练使用的损失函数为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

对于实体分类任务在R-GCN传播层之后，经过一个全连接层，计算Softmax得分：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

在R-GCN的基础上，有多个基于GCN的方法出现。SACN[31]方法将GCN扩展至带权值的多个子图聚合的形式；VR-GCN[33]结合TransE的思想将不同方向的信息进行不同的翻译处理；TransGCN[34]设计了实数域和复数域下两种基于翻译的思想的编码器；COMPGCN[3]泛化了R-GCN，SACN，VR-GCN等方法，提出了基于组合的多方向图卷积神经网络。

除此之外还有一部分基于图神经网络的方法融合了注意力机制。KBGAT[51]融合图注意力机制，是第一个将图注意力网络应用到知识图谱领域的方法。

KBGAT认对于一个以为顶点的三元组，应该能够传播给一个消息：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

之后融合注意力机制：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

聚合所有的邻居信息：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

KBGAT还使用了多头注意力机制：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

基于负采样的损失函数：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

A2N[52]注意力的计算依赖于具体预测对象；LAN[53]先是基于统计的方法计算关系之间的影响程度，之后计算自注意力权值；Graph2Seq[54]将相同方向的消息融合，之后将不同方向的消息与中心实体融合，融合的方式是拼接之后经过全连接层；RGHAT[55]使用分组的方式对邻居的注意力进行了细分。ReInceptionE[56]使用了Inception的思想设计了新的构造消息的方式，并且采用注意力机制进行探索。

基于图神经网络的表示学习方法，其优点是充分利用了图的结构，往往能够取得更好的效果，但是存在以下问题：一是过度参数化，往往需要采取多重的防过拟合措施来取得合理的结果，这也导致了一般的网络深度在1至3层，再多的层数会导致效果的下降；二是难以适用于大规模的知识图谱，训练所需的计算成本和时间成本都非常高。

## 2.3 超网络

超网络（Hypernetwork），是指一个外部的神经网络的输出是另一个神经网络（也叫做主网络）的参数。超网络的思想首先起源于图像识别领域，比如2016年，研究者针对短时天气预报任务，根据前面时序的图像数据动态生成卷积核，在当前的天气图像上进行卷积操作[57]。类似的算法还包括DFN[58]算法、SRCNN[59]算法等。在2017年，有研究者将超网络拓展到卷积神经网络和循环神经网络上[60]。其定义的生成卷积核的方式为两个线性变换：

|  |  |  |
| --- | --- | --- |
|  |  | (2.47) |

其中的是用来为第层产生卷积核所用的权值向量，是第层的第个输入通道的卷积核。作者还额外提出了HyperRNN，实现了超网络引入到RNN中，具体的示意图如图2所示，该图来自于作者原论文[60]。通过实验作者发现其在使用更少的参数情况下，在多个不同的预测任务下都达到了不错的训练结果。此外还有研究人员尝试把超网络应用到网络结构搜索[61]、3D场景表示[62]等领域。HypER[63]是首个把超网络引入到知识图谱表示学习领域的模型，它利用超网络生成一维卷积核，然后在一维的实体向量上进行卷积操作。

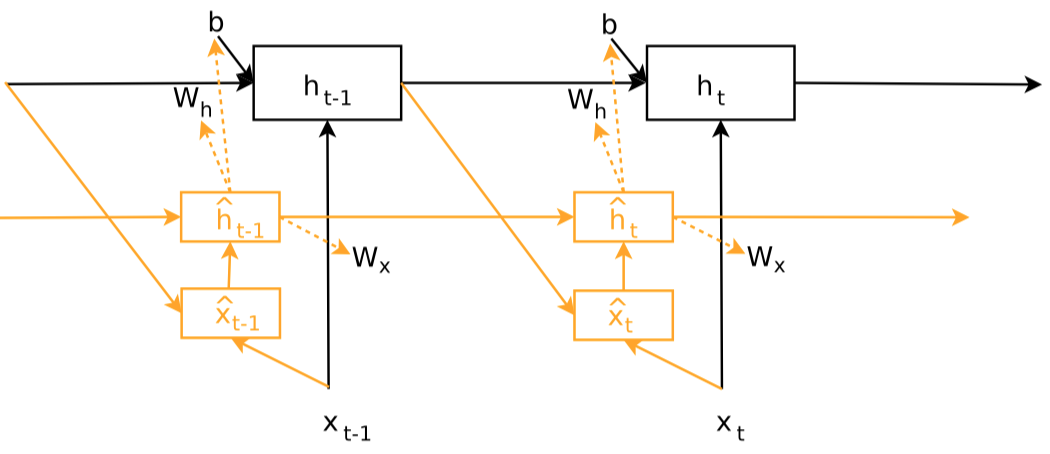


图2 HyperRNN示意图

最近也开始有研究者尝试把超网络用在图神经网络领域，Nachmani等人使用一个MLP来产生图神经网络的参数[64]；之后，他们又发现如果把消息传递中的初始消息和当前消息同时作为输入到超网络中，能够提升模型的性能[65]。LGNN[66]方法使用超网络为图中的每一个顶点都生成一个权值矩阵，实现了图神经网络卷积核参数的局部化。总体而言，目前对于超网络在图神经网络上的研究还不多。

## 2.4 本章小结

本章主要介绍相关理论基础。首先介绍了图神经网络的基本概念和异质图神经网络的目前研究概况；其次，描述了几种传统的知识图谱表示学习算法，还对图神经网络在知识图谱领域的应用进行了详细说明；最后，介绍了超网络的一般概念和其最新的发展情况。

# 第三章 基于关系感知的图注意力网络

## 3.1 现有问题分析

目前应用在知识图谱领域上的图神经网络对关系的建模能力不足。现有的图神经网络方法会首先将实体与关系表示为嵌入向量，然后将实体嵌入与关系嵌入进行某种组合操作，例如向量元素积、向量拼接等，之后作为共同的输入进行图神经网络的卷积操作。本文对于目前的基于图神经网络的知识表示学习算法所采用的消息函数进行了调查，结果如表2所示。

表2 目前图神经网络知识表示学习算法所采用的消息函数

|  |  |
| --- | --- |
| 方法 | 获取邻居信息方式 |
| R-GCN[30] |  |
| Graph2Seq[54] | or |
| SACN[31] |  |
| VR-GCN[33] |  |
| COMPGCN[3] |  |
| RGHAT[55] |  |
| ReInceptionE[56] |  |

知识图谱中的关系蕴含了丰富的语义信息。比如在著名的Freebase知识图谱中，实体来源与人名、性别、地区、职业、电影等各不相关的领域；对于关系，存在表示人的出生地、人的性别、人的学历、城市属于哪个国家等各种不同类型。将这样的异质性对应在图结构上，以图3为例。实体彼得·杰克逊是电影《指环王》的导演，同时他也是《指环王》的编剧和演员。对于中心实体《指环王》来说，彼得·杰克逊在链接不同的关系导演、编剧、演员时，他代表的语义各不相同。同样，编剧和演员也有很多其他的实体，比如弗兰·威尔士是编剧，伊恩·麦凯伦也是演员。即不同的关系能够反映同一实体的不同语义信息，同一关系下的不同实体应该共享某种特定的语义。

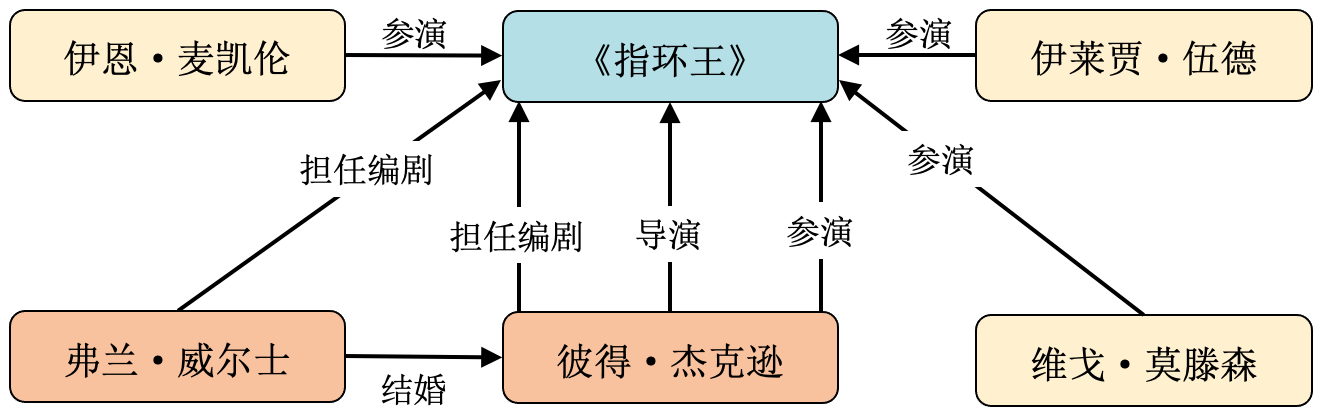


图3 多关系知识图谱示例

但是大多数目前在知识图谱上利用图神经网络学习知识表示的算法没有充分考虑关系带来的差异性。从表2可以看出，除了R-GCN外的方法，当同一实体链接到不同关系时，都是经过固定的网络参数，无法捕获关系特定的特征。对于关系的建模方法主要是在模型初始化时，赋予关系与实体一个具有相同维度的嵌入向量。这种做法忽略了知识图谱当中存在的一个重要现象，知识图谱中关系的数量要远远小于实体数量。例如在FB15k-237[67]数据集当中，实体的平均度有18.7，但是关系的平均度达到了1148！如果将关系和实体都看作是一个图上的顶点的话，往往一个关系拥有远比一个实体多的邻居，在整个图上拥有更重要的地位。在这种情况下，如果将关系学习为和实体拥有相同维度的嵌入，会直接限制模型对于关系的表达能力。R-GCN虽然从理论上能够建模更多的关系信息，但它为每个关系都定义了单独的权值矩阵，这就导致R-GCN出现了过度参数化的问题。另外，在R-GCN中，没有考虑如何处理关系嵌入，这导致R-GCN的泛用性收到了限制。

在另一方面，本文对常用的消息聚合方法进行了调研，发现一般的聚合方法忽略了关系的方向性。主流的消息聚合可以分为两个过程：单层的消息聚合以及多层的层间信息聚合。单层的消息聚合决定了如何混合不同的邻居信息；多层的层间信息聚合决定了如何选取不同阶的邻居信息。

目前在图神经网络领域，聚合方法主要包括基于池化操作、基于注意力以及基于门机制的方法等。而在基于图神经网络的知识表示学习算法中，主流的两种聚合操作是基于基于池化操作的聚合以及基于注意力的聚合。

（1）基于池化操作的聚合

池化操作（Pooling）的核心思想是选择重要特征，忽略次要特征。它能够减小输入特征的大小，提取重要的核心特征，减低噪音，从而加速模型的收敛，提升模型的性能。池化操作最常见的领域是图像处理领域。而在图神经网络中，池化操作一般为对于一维向量的操作，比如最大池化（Max Pooling）、均值池化（Average Pooling）等。最常用的操作是均值池化，比如在GCN中，就使用了提前计算好的固定正则项进行均值池化，取得了较好的效果。另外由于池化操作无需引入额外的参数，计算较为简单，因此得到了广泛的应用。

（2）基于注意力的聚合

在图神经网络领域，基于注意力的聚合起源于GAT[46]方法。GAT方法使用单层的多头线性注意力层进行邻居间的信息聚合，相比较与GCN取得了较大的效果提升。GAT的方法已经被应用到了多个领域，在知识图谱表示学习领域中，KBGAT[51]是首个将GAT应用到学习知识表示的算法；SACN[31]为每个关系都学习一个独立的权重系数；RGHAT[55]考虑到不同关系应该具有不同的权重，设计了一个两层的注意力网络。

使用基于注意力的聚合方法往往表现出更好的性能。但是在这些方法中没有考虑知识三元组中关系的方向问题。以图4为例，三元组<北京，位于，中国>和三元组<张三，居住在，北京>，对于实体北京来说，它的两个邻居实体：中国和张三，分别属于实体北京的尾实体和头实体。这种指向/方向实际上隐性地包含一种从属信息，比如北京属于中国，张三居住于北京。即关系的方向性是有实际的含义的。但是目前基于注意力的聚合方法忽略了这种关系的方向性。

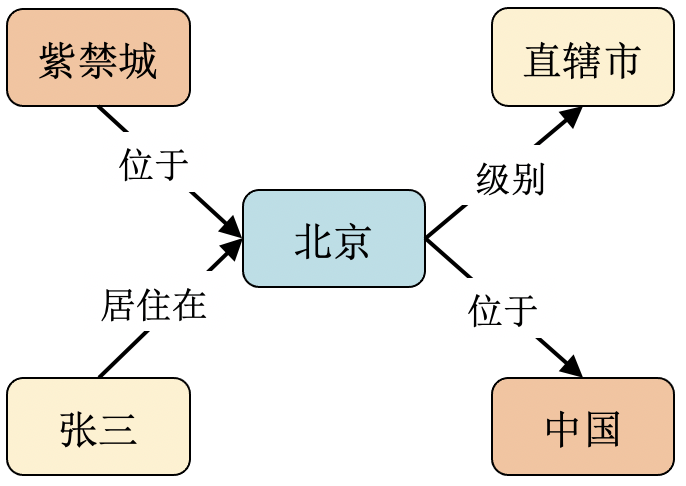


图4 关系的方向性示例

为了解决上述问题，本文设计了一种基于关系感知的图注意力网络（Relation Aware Graph ATtention network，RAGAT），首先在消息构造阶段，在之前的四种消息构造方法上，引入关系特定参数，讨论实现关系感知对于不同的图神经网络消息构造方法能否带来效果提升；其次，在消息聚合阶段，考虑消息的方向性，设计了一种基于注意力的多方向聚合方法。下面进行详细说明。

## 3.2 RAGAT总体结构

### 3.2.1 符号定义

为了清晰地说明论文提出的RAGAT方法的细节，本小节给出了关键的数学符号定义，如表3所示。整个知识图谱表示为，表示实体集合，表示关系集合，是所有边的集合。边存在于，说明关系存在于实体到实体。

表3 RAGAT中的符号定义

|  |  |
| --- | --- |
| 符号 | 描述 |
| ， | 单个邻居实体、邻居关系 |
|  | 中心实体 |
| ，， | 三元组分别对应的嵌入表示 |
|  | 实体-关系融合函数 |
|  | 消息函数 |
|  | 实体-关系融合后的表示 |
|  | 全局通用参数 |
|  | 全局通用权值矩阵 |
|  | 关系感知实体-关系融合函数 |
|  | 关系感知实体-关系融合后的表示 |
|  | 关系特定参数 |
|  | 关系特定对角矩阵 |
|  | 传递的邻居消息 |
|  | 以输入参数构造对角矩阵 |
|  | 哈德曼乘法运算 |
|  | 循环相关运算 |
|  | 向量拼接运算 |
|  | 实体的所有邻居集合 |
|  | 初始的关系集合 |
|  | 逆关系集合 |
|  | 自循环关系 |
|  | 属于出边的关系集合 |
|  | 属于入边的关系集合 |
|  | 自循环边的关系集合 |
|  | 方向感知权值矩阵 |
| ，， | 出边、入边以及自循环边 |
|  | 注意力参数 |
|  | 邻居的注意力权重 |
|  | 实体更新后的表示 |
|  | 多头子空间的数量 |
|  | 获取L阶邻居信息后实体的表示 |
|  | LeakyReLU激活函数 |
|  | Softmax归一化函数 |
|  | 邻居的注意力权重 |
|  | 实体更新后的表示 |

表3 RAGAT中的符号定义（续）

|  |  |
| --- | --- |
| 符号 | 描述 |
|  | 邻居信息聚合后的嵌入表示 |
|  | 自注意力层 |
|  | 多头聚合函数 |
|  | 关系转换矩阵 |
|  | 卷积运算 |
|  | 卷积核 |
|  | 张量展开为一维向量 |
|  | 三元组存在的概率 |
|  | 个随机特征排列 |
|  | 棋盘形式特征重组 |
|  | 循环卷积运算 |
| ， | 激活函数 |
|  | Sigmoid激活函数 |
|  | 损失函数 |

### 3.2.2 网络结构

本节提出知识图谱关系感知图注意力网络RAGAT，整体结构如图5所示。从整体来看，RAGAT主要包括三个模块：数据预处理部分、基于图神经网络的编码器与基于卷积神经网络的解码器。

在数据预处理部分，主要负责对知识图谱的原始文本形式进行处理，将其转换为适合模型处理的输入形式。具体而言，有两方面的任务：实体与关系编码和关系拓展。实体与关系编码指从原始形式三元组文本中识别关系和实体，并且赋予全局唯一的标识ID。关系拓展指为了更充分的利用知识图谱的关系信息，为每个关系都新建关联一个逆关系。

与COMPGCN[3]，R-GCN[30]等以前的工作一样，本文同样将关系拓展为初始关系、逆关系和自循环关系。逆关系是人为新增的初始关系的反方向关系。因此，新的关系集合为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是逆关系集合，是自循环关系集合。新的边集合为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

编码器用于编码知识三元组，输入知识图谱，输出实体与关系的表示集合，属于本文研究的核心部分，主要负责在知识图谱形成的异质网络下，学习知识图谱实体和关系的嵌入表示。学习实体和关系表示的目的在于将原来没有意义的ID编码，转化为能够预测图结构的多维数值形式，学习到的表示能够用作更多的下游任务，比如知识图谱补全、实体识别、知识图谱对齐、推荐系统等众多领域。编码器可进一步细分为消息构造层和消息聚合层，消息构造层使用基于关系感知的邻居消息构造方法，对单个知识图谱三元组进行表示学习；消息聚合层使用基于注意力的多方向邻居信息聚合方法。

解码器用于针对特定的预测任务，使用上一部分的学习算法得到的知识表示，评估预测任务下的各项性能指标。在本研究中，使用在学术界最常见的链路预测任务作为预测目标，使用ConvE[25]和InteractE[29]两种不同的解码器评估学习算法性能。编码器加解码器一起实现端到端的模型训练。

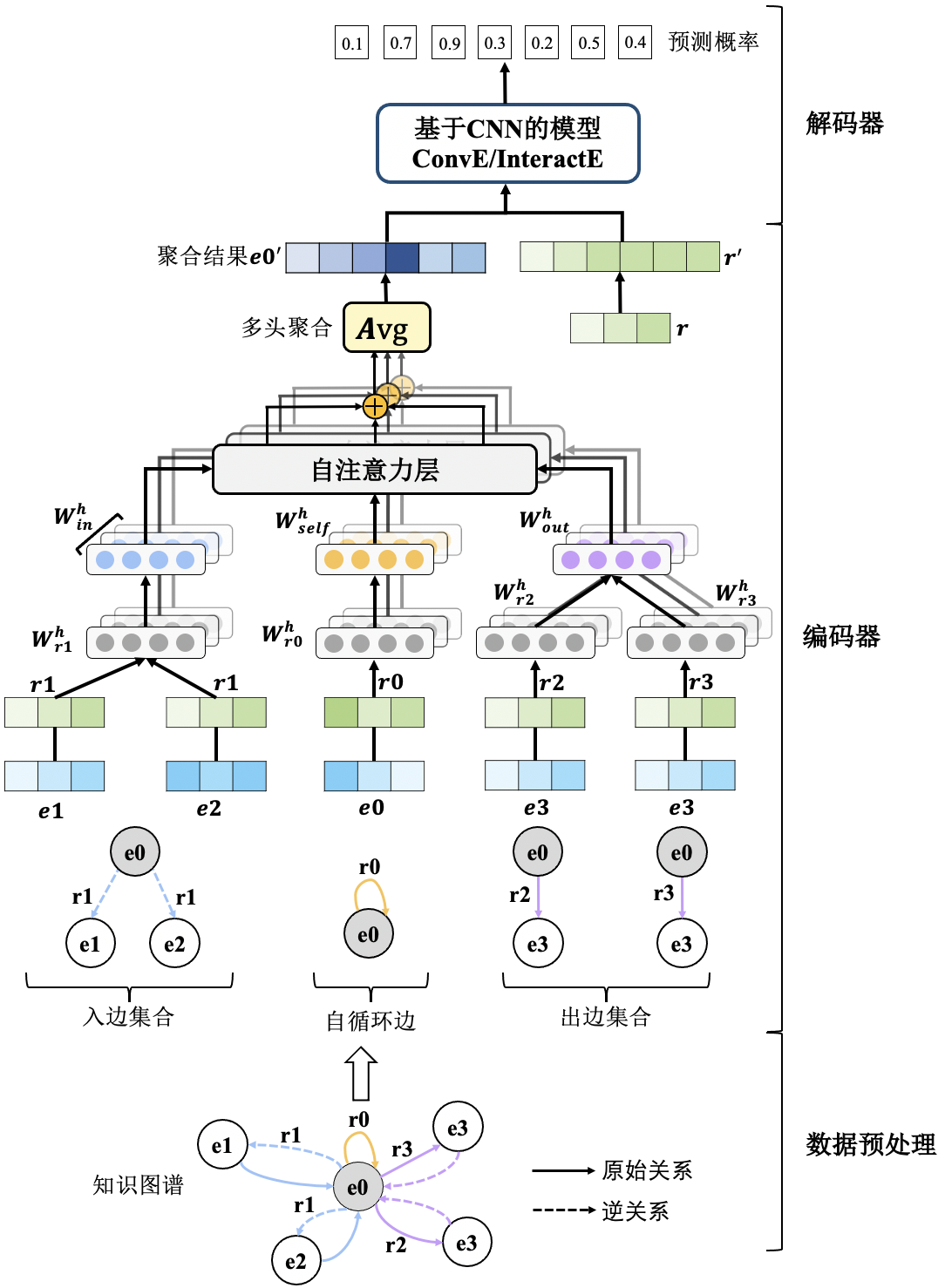


图5 RAGAT整体结构

## 3.2 RAGAT关键步骤

### 3.2.1 基于关系感知的消息构造

本文设计了一种新的基于关系感知的邻居信息构造函数，用来对单个邻居信息进行编码。如图6所示。

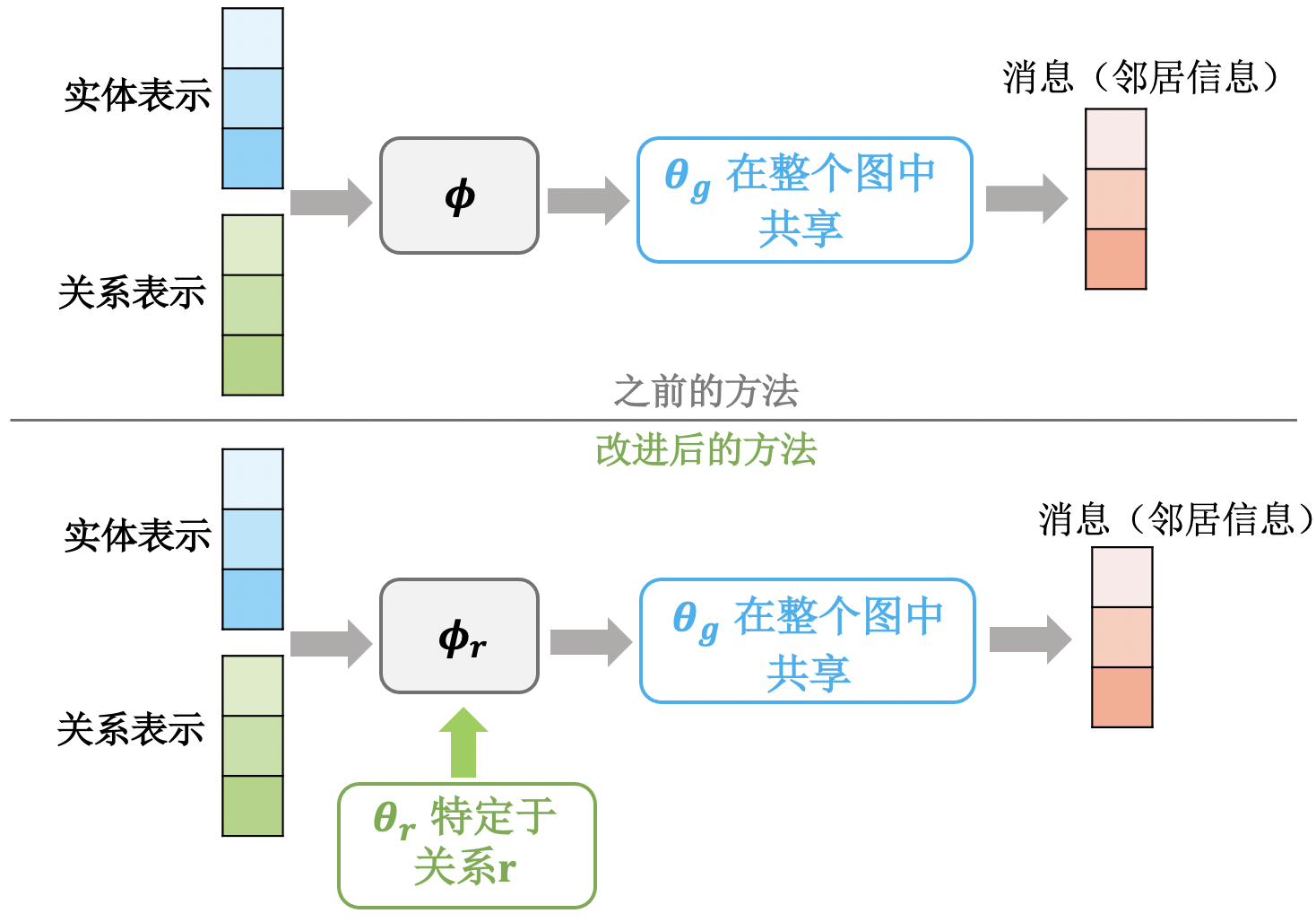


图6 基于关系感知的消息构造函数

首先，本文使用一种泛化的统一形式概括在表2中调研的现有的各类消息构造函数。假设实体和关系相连作为中心实体的邻居，经过实体-关系合并函数融合关系表示与实体的表示。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示实体-关系融合后的向量输出。之后，作为输入，经过网络参数，得到最终的消息：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

就表示邻居实体经过关系传递给实体的信息。参数在整个图上共享。在通常情况下都是一个权值矩阵，表示对于输入进行线性的投影。为了不同关系能够反映实体的不同方面，引入关系特定的参数。因此，在本文中获取消息的方法变为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

通过上面的公式，实际上为每个关系都构造了不同的消息函数，同时能够用来捕获关系之间的通用特征。

最直接定义的形式是假设其为一个权值矩阵，用来导出不同关系的独有特征。但是，类似于R-GCN的弊端，这样的形式可能会导致过度参数化。为了保证各个关系拥有自己的参数，同时保持较低的计算复杂度，在本研究中考虑限制为对角矩阵。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，，。实际上，可以被拓展到更多的形式，比如卷积核。但是考虑到模型训练的复杂度以及RAGAT的目的主要是探究实现关系感知对于消息构造函数的影响，因此，在本文中的定义都是对角矩阵。

为了捕获统一关系下的独有特征，对相应的实体进行转换操作：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

接下来，为了融合关系的表示，在前人工作的基础上，本文设计了五种不同的消息构造函数。

（1）基于翻译的消息函数

在知识表示学习的经典算法TransE[1]中，知识图谱中的关系被视为实体到实体的翻译，即假设头实体加上关系等于尾实体，这一转换操作被称作翻译（Translation）。同时，基于翻译的思想在VR-GCN[33]，TransGCN[34]等基于图神经网络的知识表示学习算法中也得到了应用，具体定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

为了在这一基础上学习关系特定的特征，本文使用替换。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

这一改变使得关系和实体之间能够拥有更丰富的交互，在中的每个元素，和相乘然后减去。

（2）基于向量乘法的消息函数

受到DistMult[17]的启发，基于向量相乘的方法在COMPGCN[3]中采用：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，表示哈德曼乘法。和前面对基于翻译的消息函数的改进方法一样，使用替换。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

（3）基于循环相关的消息函数

为了捕获实体与关系之间的特征交互，一种直接的办法是使用张量乘法，利用实体和关系的向量嵌入表示，构造实体-关系的二维矩阵。但是这种办法会急剧的增加参数量以及计算复杂度，因此，在HolE[18]中使用循环相关方法压缩实体-关系的二维矩阵，得到向量化的表示形式。基于循环相关的消息函数在COMPGCN[3]中也得到了采用。在本研究中，和前面的改进类似：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，表示循环相关操作。

（4）基于向量拼接的消息函数

很多基于图神经网络的知识表示学习算法直接拼接实体嵌入和关系嵌入作为神经网络的输入，例如A2N[52]，Graph2Seq[54]，KBGAT[51]等。拼接这一操作会扩大权值矩阵的维度，导致网络参数量变为前面三种消息函数的两倍。类似的，定义基于函数拼接的消息函数如下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

（5）基于交叉交互的消息函数

前面四种方法都是直接利用改进前人工作使用的消息函数，在本研究中，受到CrossE[21]方法的启发，提出了一种新的基于交叉交互的消息构造函数。CrossE认为实体和关系之中存在交叉交互（crossover interaction）。在CrossE中对于实体和关系分别学习交互嵌入（interaction embedding）：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是关系对实体的交互，是实体对关系的交互。之后，联合两种交互嵌入：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

如果把替换为，得到下面的形式：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

因为在本文中被限制为对角矩阵，因此可以CrossE的方法可以无缝的融合进来。同时，如果把看做是对角矩阵，而不是简单的向量的话，那么可以从新的角度来解释CrossE。

CrossE的问题之一在于在原来的方法中没有具体解释意义，与的区别也很模糊。在CrossE中，判断链路存在概率的得分函数定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

如果将看做是对角矩阵，，那么类似于TransR的思路，将和投影至关系空间（Relation Space）中。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

被用作进行关系特定的线性转换，而被用来对进行缩放操作。之后，作为CrossE中的交互嵌入。激活函数用来确保目标实体与交互嵌入拥有相似的分布。最后使用向量内积计算交互嵌入与目标实体之间的相似性，作为链路预测的概率。整个过程如图7所示。

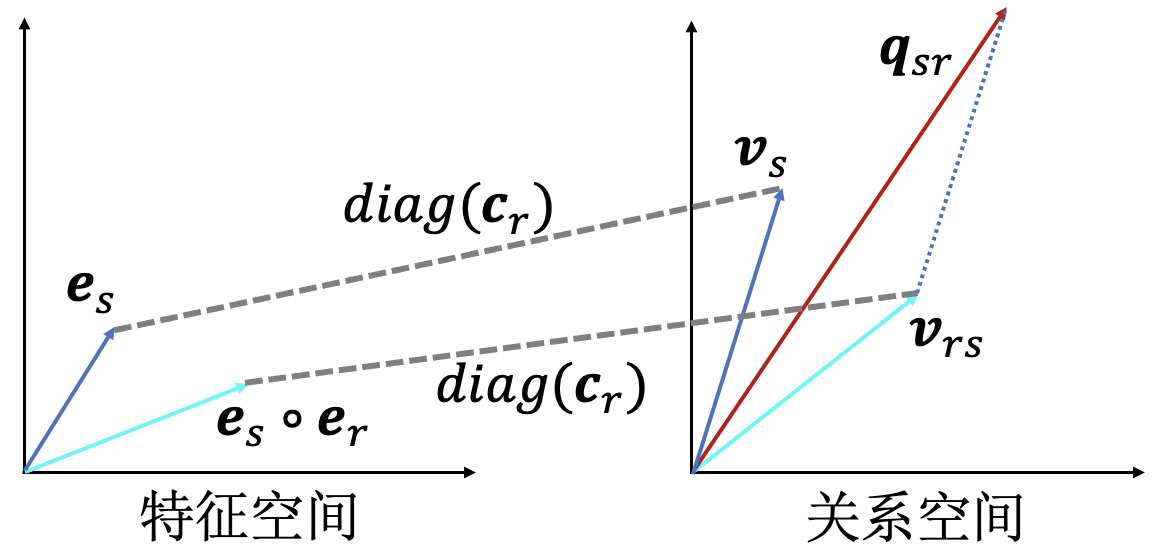


图7 针对CrossE的一种代数角度解释

RAGAT设计的所有五种关系感知函数，总结在表4中。

表4 五种关系感知函数变体

|  |  |  |
| --- | --- | --- |
| 关系感知方法变体 | 对应编码器缩写 | 数学形式 |
| 基于翻译 | Trans-GAT |  |
| 基于向量乘法 | Mult-GAT |  |
| 基于循环相关 | Corr-GAT |  |
| 基于向量拼接 | Concat-GAT |  |
| 基于交叉交互 | Cross-GAT |  |

其中，基于翻译、基于向量乘法、基于循环相关、基于向量拼接这四种方法属于在之前基于图神经网络的知识表示学习方法中得到应用的方法，使用这四种方法目的在于证明引入关系感知参数的作用。基于交叉交互的方法属于本文提出的新方法，它无缝的融合了CrossE方法中提出的实体与关系之间的交叉交互概念，更加灵活地处理和。

### 3.2.2 基于注意力的多方向消息聚合

为了实现对于关系方向的感知，本文首先对来源于不同方向的关系进行了区分，即将关系划分为入边、出边以及自循环边三个方向组，提取三个不同方向组的共有特征；然后基于多头注意力进行邻居信息聚合。入边和出边将当前层的邻居信息划分为了两个不同的子图；自循环边将上一层的邻居信息和当前层的邻居信息进行了划分。该方案考虑到关系的方向带来的异质性，同时能够自适应地选择在全局意义上更重要的邻居信息。

在一般情况下，传统的GCN方法使用均值池化操作，不区分邻居关系的方向：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，表示邻居三元组编码后的信息，代表实体的所有邻居,表示正则项，通常情况下。上面的公式可以进一步写为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

在这种情况下，由于存在自循环边，实体自身的信息通过自循环边传递到下一层。固定的权值矩阵会混合邻居信息以及自身信息。实际上，邻居信息与自身信息在很多情况下不处于同一分布，比如在知识图谱中，实体可能代表某个人，而他的邻居实体常常是代表人的各类属性，比如性别、出生地、职业等。这种情况下使用混合邻居信息与自身信息在直觉上是不合理的。同样，根据在前面的讨论，关系的方向性无法通过一个全局共享的权值实现感知。

本文提出的基于注意力的多方向邻居信息聚合方法如图8所示。首先，对于实体进行基于关系方向的采样，根据关系的方向将不同的关系划分到入边、出边和自循环边三个分组集合中。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

为了感知关系的方向特性，对属于，，三个集合中的关系，引入方向特定的参数。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，，，分别表示出边、入边以及自循环边。在出边集合中，中心实体处于三元组的头实体位置，；入边集合中，实体位于尾实体，；自循环边集合中，头实体和尾实体都是实体，。通过，模型能够学习来自三个不同方向的邻居信息。

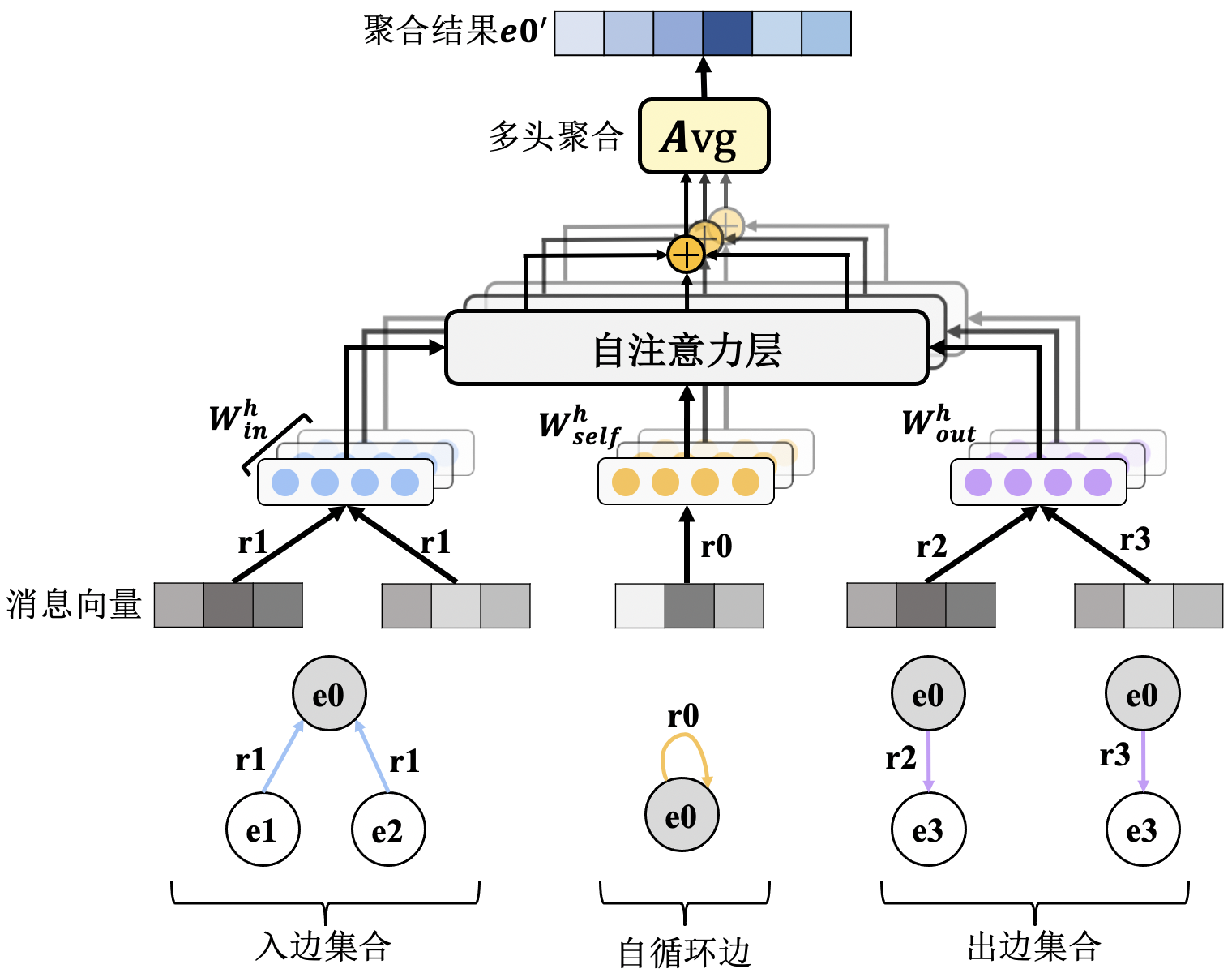
**

图8 基于注意力的多方向消息聚合函数

考虑到不同的邻居三元组对于中心实体的重要性是不同的，比如<张三，职业，篮球队员>和<张三，国籍，中国>，张三的职业是篮球队员这条信息比张三的国籍是中国更加具有差异性，更能够体现张三的特征。而传统GCN的方法使用了提前定义好的正则项进行均值池化操作，比如邻居数量的倒数，这样的固定值很明显无法适应性地学习邻居信息的重要性。

本文采用单层多头自注意力机制计算邻居信息的重要性，该方法首次在GAT中采用，随后由KBGAT[51]应用到知识图谱领域。使用该方法主要是考虑到控制方法的复杂度，使用单层注意力计算复杂度较低，与在Transformer[68]中采用的Query，Key，Value自注意力方法比较起来，参数量少，同时无需考虑使用残差等结构来加速模型的收敛。另外，该方法在多个不同的模型中都得到了应用，便于在实验中进行比较。

是单个邻居编码后的消息，首先将其输入到单层注意力层计算绝对权重：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，作为注意力向量，衡量邻居信息在全局意义上的重要性。表示实体通过关系传递给实体的消息的重要性系数。不随着中心实体而改变。因此，由于实体具有多个不同的邻居，需要衡量在局部的多个邻居消息中的相对重要程度。为了获取相对权重，使用函数归一化：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示在实体和实体之间相连的关系集合。可以看到，通过上面的操作，。此时，利用相对注意力系数聚合邻居信息。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

为了保证计算结果的稳定同时提升注意力，多头注意力机制在Transformer和GAT等利用自注意力的方法中都得到采用。本文同样使用多头注意力进一步提升模型性能。利用多头机制将不同方向的消息从多个子空间中进行学习：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

代表第h个子空间，在不同子空间中学习相似但不相同的。之后，为了控制参数量，使用均值聚合多个子空间下的消息表示：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

重复执行聚合公式次，每次聚合都能够让获得新的高阶邻居信息，最终成为获取阶邻居的表示。

之后，根据不同的关系与中心实体的相对方向，将关系划分到入、出、自循环三个分组。在每个方向分组下都具有独立的全局矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示多个方向邻居信息聚合后的嵌入。是基于单层注意力计算得到的相对注意力系数，具体计算方法为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

横跨不同方向的不同关系，因此，可以用来衡量各个邻居信息在全局意义下的重要程度。之后，采用多头机制加速模型的收敛：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

是获取了第阶邻居信息后的实体的表示，为了获取更高阶的邻居信息，只需要重复执行聚合函数即可，直至获得阶的邻居信息为止，最终得到。

通过组合前面的两种关键设计，RAGAT将关系进行了更细粒度的层级划分，如图9所示。不同关系拥有独立的关系特定参数，属于第一层；之后，不同的关系划分到不同的方向，在每个方向下有独立的方向参数，由在此方向下的所有关系共享，属于第二层。经过这两层的消息传递，得到最终的聚合结果。

前面描述了基于实体的消息聚合，为了保证最终学习的知识表示都能够拥有相同维度，对关系嵌入进行转换：

|  |  |  |
| --- | --- | --- |
|  |  | (3.31) |

是可学习的权值矩阵，用来将投影到存在的空间中去

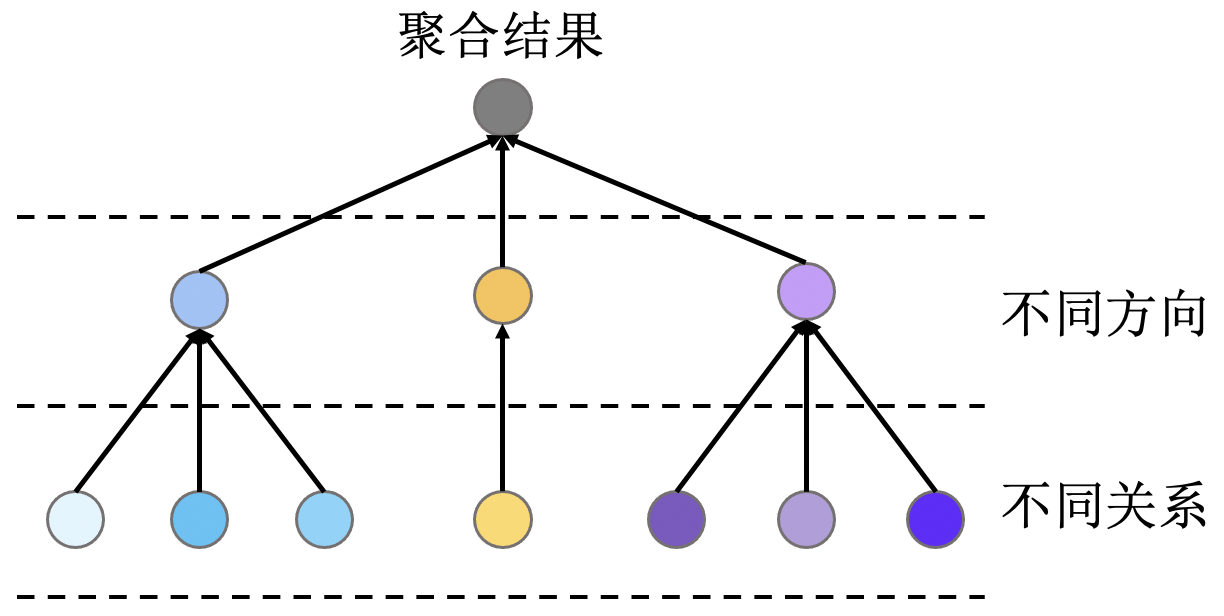


图9 RAGAT中关系的分层结构

### 3.2.3 得分函数的选择

为了综合评估加入关系感知参数的效果，RAGAT使用不同的得分函数（解码器）在链路预测任务下进行评估。链路预测是指使用学习得到的表示预测知识图谱中某个三元组存在的概率。在确定使用的得分函数后，由于是端到端的训练方法，得分函数的选择直接关系到模型的采样方法、训练技术、损失函数乃至于防过拟合策略的选择等。常见的得分函数包括基于翻译的方法比如TransE[1]，基于矩阵分解的方法DistMult[17]，以及基于卷积神经网络的方法ConvE[25]等。由于基于卷积神经网络的方法往往都表现出更好的性能，因此ConvE方法也成为了最常见的选择之一。为了对比实验的公平，本文同样采用ConvE方法作为得分函数。此外，由于ConvE本身在特征交互、卷积方法等存在限制，本文还采用最新提出的ConvE的改进方法InteractE[29]作为另一种解码器，进一步提升模型性能。

ConvE方法输入为单个的知识图谱三元组，输出为预测该三元组是否存在的概率大小。首先，ConvE重塑实体和关系的向量化嵌入表示，为二维张量；之后，ConvE在此二维张量上进行标准卷积操作，卷积操作导出的特征最后经过单层神经网络输出预测概率。

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是卷积操作，表示卷积核，表示将张量展开为一维向量，用来将导出的特征投影至与目标实体相同的空间中，最后使用向量内积求出预测概率，最后经过Sigmoid激活函数，求出。

InteractE从三个方面增强ConvE的表达能力：特征排列（Feature Permutation）、棋盘状特征重组（Checkered Feature Reshaping）以及循环卷积（Circular Convolution），具体结构如图10所示。

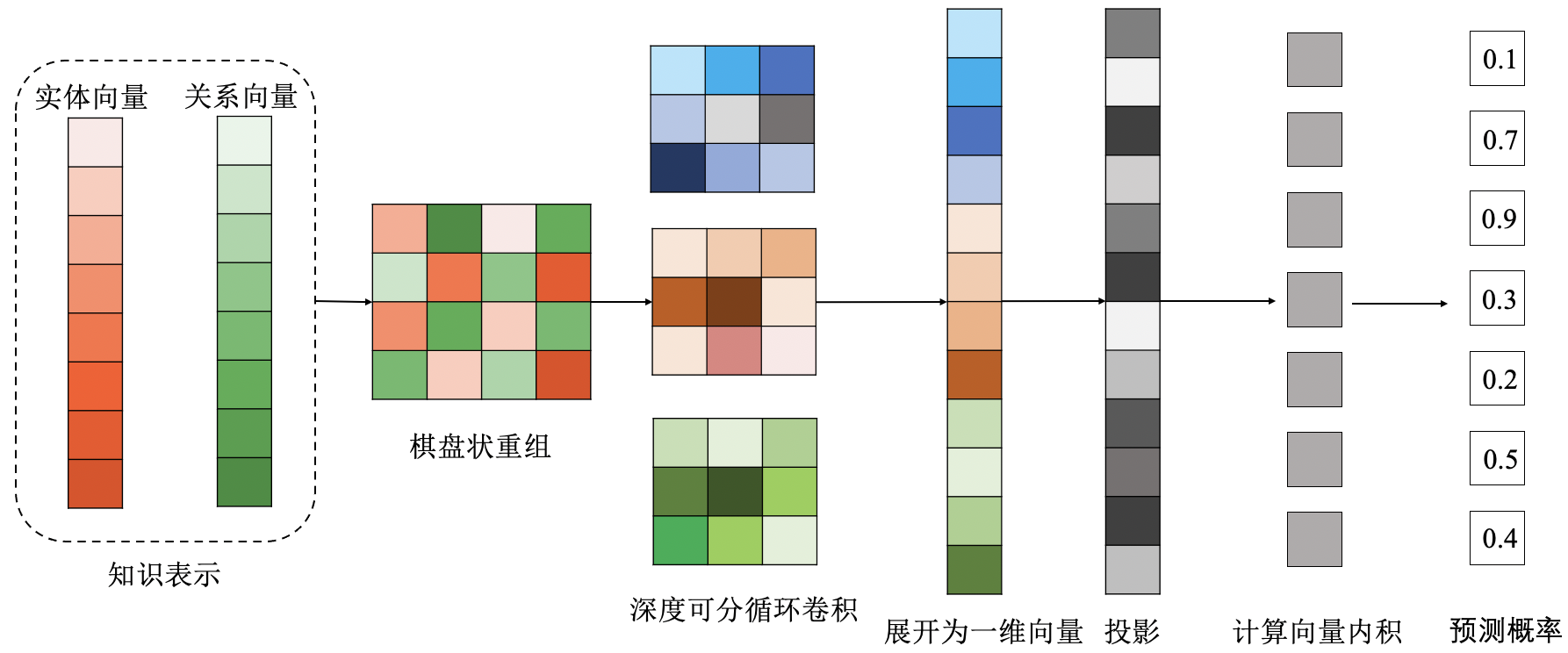


图10 InteractE解码器结构

对于输入，首先进行个随机特征排列：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是对的向量维度进行的第次重新随机排列，是对的向量维度进行的第次重新随机排列。之后，为了增加与之间的特征交互，将这两个向量重组为棋盘状的二维张量：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是表示对个特征排列的棋盘方式的重组。最后，使用循环卷积操作导出特征，并且经过单层神经网络输出预测概率：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示循环卷积操作。激活函数与选择为函数和函数。

RAGAT使用最常见的交叉熵损失函数作为优化目标，同时，由于ConvE方法和InteractE方法属于one-vs-all的对抗全图的方法，因此无需设计额外的负采样过程，从另一个角度讲，负采样为正确的目标实体以外的所有实体。同时，使用标签平滑（Label Smoothing）提升优化效果。最终采用的损失函数定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

表示第个三元组，表示真实的第个三元组是否存在的标签，是模型的输出。

## 3.3 本章小结

本章详细介绍了RAGAT的整体设计。本章首先说明前常用的图神经网络模型中存在的问题；随后，给出本章所用的数学符号定义；之后，阐明论文提出的RAGAT总体结构和符号定义；最后，详细说明RAGAT关键步骤的设计细节，包括：（1）基于关系感知的消息构造函数，通过引入关系感知参数，改进了目前存在四种不同的消息构造函数，同时基于CrossE提出了一种新的消息构造函数，并且为CrossE提供了代数角度的解释；通过将关系特定的参数限制在对角矩阵降低模型的计算复杂度；（2）基于注意力的多方向消息聚合方法，能够学习目前大多数模型没有考虑的关系方向问题，将关系划分到入边、出边和自循环边三个方向集合，各自拥有独立的通用参数，最后结合多头注意力机制进一步提升模型效果；（3）选择ConvE和InteractE两种不同的得分函数评估模型效果。

# 第四章 异质知识图谱图神经网络

## 4.1 现有问题分析

RAGAT方法为了增强图神经网络对于关系语义的捕获能力，而额外训练了关系感知参数。但是引入的关系参数受到知识图谱中关系数量的限制，为了计算简便而定义为对角矩阵，这样的做法事实上限制了RAGAT模型的建模能力，特别是对于知识图谱中高频率出现的关系而言，难以充分表达其语义信息。如果能够将关系感知参数拓展到更多形式，例如卷积核等，或许能够带来更多的性能提升。

但是增加关系感知参数带来的一个严重问题是参数量会随着知识图谱中关系数量以及关系感知参数本身的形式而急剧增加，会引入过多不必要的网络参数，最终导致模型过于复杂而难以训练。实际上本文经过调研发现，同样的问题也存在于异质图神经网络领域当中。

异质图神经网络HGNN是近期出现在图神经网络领域中新的研究热点，这些新模型通常依赖于设计复杂的异质信息聚合策略，大多在拥有边类型数量较少的异质图上取得了更好的结果，但是复杂的模型设计导致它们无法直接应用到可能拥有成千上百的关系类型的知识图谱中。

异质图神经网络建模异质性的一般策略为：（1）对于不同类型（异质）的点/边，每个类型独立定义异质参数，通常为权值矩阵；（2）在此基础上，采用更多复杂的建模策略，例如异质图注意力（HGAT[69]）、图结构学习（HGSL[50]）、融合节点内容（HetGNN[48]）、类型组合（HetSANN[70]）等。这样的方法比原始的图神经网络（GCN[32]、GAT[46]等）更能够充分学习异质图中蕴含的信息，表现出了更好的性能。

更加复杂的建模策略带来的后果之一是限制了这些模型的应用范围，特别是它们往往只适合于处理边类型数量较少的图。表5是对近两年出现的异质图神经网络采用的数据集以及边类型数量的调研。从表中可以看到，大多数模型实验采用的数据集的边类型数量集中在2-5种。而知识图谱是一种异质性很强的图，往往包含来自不同领域的实体，可能拥有成百上千的关系类型。在这种情况下，异质图神经网络就无法直接地应用到知识图谱上。

表5 异质图神经网络采用的数据集

|  |  |
| --- | --- |
| **HGNN模型** | **数据集（边类型数量）** |
| HeteGCN[71] (2021) | 20NG (4), MR (4), R8 (4), R52(4) |
| HGSL[50] (2021) | DBLP (4), ACM (4), Yelp (6) |
| SliCE [72](2021) | Amazon (2), Twitter (4), Healthcare (4) |
| MR-GCN [73](2020) | ACM (2), IMDB (2), Reuters (4) |
| MAGNN[74] (2020) | IMDB (2), DBLP (3), Last.fm (3) |
| HGT[49] (2020) | CS (5), Med (5), OAG (5) |
| HetSANN[70] (2020) | IMDB (2), DBLP (3), AMiner (3) |

以MAGNN[74]方法为例，示意图11来自于MAGNN原文，MAGNN方法基于元路径（meta path）处理异质图，它会首先根据不同的图构建出不同的元路径，例如在一个关于电影的异质图（IMDB）上，可以构建出电影-导演-电影、电影-演员-电影、导演-电影-演员-电影-导演等元路径，一个图上的顶点可以通过这些元路径链接到不同的邻居顶点。MAGNN首先在单个元路径下，通过了不同的消息构造函数捕获单个邻居信息，然后在单个元路径下的所有消息之间基于多头注意力进行聚合；不同元路径之间再通过注意力进行聚合。这种做法在图边类型数量较少的情况下能够实现细粒度建模，但是如果在一个拥有100个关系的知识图谱中，即使只考虑两跳的路径，也最后会出现一共10100个元路径（）。考虑到不同元路径下都有自己独立的网络参数，MAGNN方法从模型复杂度的角度来看，显然是无法直接用在知识图谱领域上的。

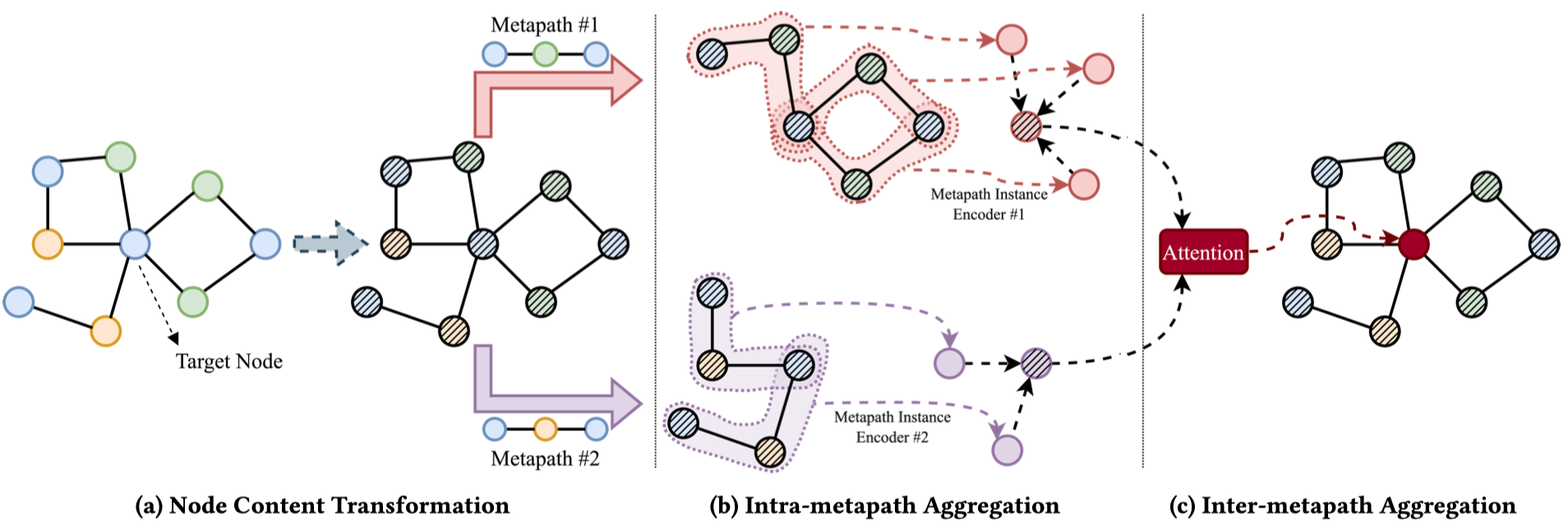


图11 MAGNN方法示意图

为了解决上述问题，本文提出了一种引入超网络（Hypernetwork）的异质知识图谱图神经网络（Heterogeneous Knowledge Graph Neural Network，HKGN），与RAGAT中通过训练额外的关系参数来增强对知识图谱的信息抽取能力不同，HKGN当中所有的参数都是由外部的超网络生成的。它一方面通过外部的超网络直接生成所需要的异质参数，将任意形式的参数都压缩为一维向量，从而大大减小了模型参数量；另一方面，在捕获邻居信息时，引入关系卷积层，通过增强邻居实体与邻居关系之间的特征交互，进一步提升了模型的性能。下面是HKGN的详细说明。

## 4.2 HKGN模型结构

HKGN模型同样符合一般的编码器+解码器的架构，核心区别是HKGN的编码器拥有一个异质图神经网络作为主网络，接收的输入是知识图谱中的实体嵌入，输出是更新后的实体嵌入；一个额外的网络作为超网络，接收的输入是知识图谱中的关系取值向量，输出是主网络运算所需的参数。

### 4.2.1 符号定义

为了方便介绍HKGN的设计思路与整体结构，表6说明了本章节用到的关键数学符号及其含义。

表6 HKGN中的符号定义

|  |  |
| --- | --- |
| 符号 | 描述 |
|  | 原始知识图谱 |
| ，， | 实体集合、关系集合、边集合 |
|  | 关系的逆关系 |
|  | 自循环边 |
|  | 逆关系边集合 |
|  | 拓展后的边集合 |
|  | 关系独有的权值向量 |
|  | 全局参数 |
|  | 超网络本身结构 |
|  | 异质参数 |
|  | 邻居实体向量 |
|  | 关系卷积核 |
|  | 二维卷积操作 |
|  | 激活函数 |
|  | 一维向量重组为二维矩阵 |
|  | 关系卷积操作的输出向量 |
|  | 线性转换矩阵 |
|  | 邻居消息向量 |
|  | 张量展开为一维向量 |
|  | 用于生成卷积核的全局参数 |
|  | 用于生成卷积核的关系向量 |

表6 HKGN中的符号定义（续）

|  |  |
| --- | --- |
| 符号 | 描述 |
|  | 一维向量重组为三维张量 |
|  | 张量乘法 |
|  | 用于生成矩阵的全局参数 |
|  | 用于生成矩阵的关系向量 |
|  | 中心实体在关系下的邻居实体集合 |
|  | 所有属于关系的向量 |
|  | 第层的实体向量 |
|  | 用于生成目标关系向量的全局参数 |
|  | 生成目标关系向量的全局参数 |
|  | 目标关系向量 |
|  | 层聚合以后的实体向量 |
|  | 损失函数 |
|  | 三元组的预测值 |
|  | 三元组的实际标签 |
|  | 向量拼接操作 |

与RAGAT一致，HKGN对于原始的知识图谱首先会拓展逆关系，然后增加自循环关系：

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

新增的逆关系以及自循环关系与原始关系在HKGN中都会经历相同的处理过程。

### 4.2.2超网络的引入

一般的异质图神经网络会首先定义具体的“类型”，比如在一个引文网络中，顶点可以分为论文（Paper）、作者（Author）、领域（Field）、机构（Institute）等类型，边可以分为引用（has\_citation\_to）、第几作者（is\_first/second/other\_author）等类型，如果进一步拓展，类型还可以定义为路径，比如论文-作者-论文、作者-论文-作者等。在知识图谱上，我们定义每一种关系都是不同的类型。

如果我们仿照异质图神经网络的一般策略，为不同的关系类型定义不同的参数，这会导致两方面的问题。（1）参数量会急剧的增加，增加的具体幅度与定义的参数类型有关，与一个知识图谱的关系数量成正比。图神经网络本身就是在一个非常稀疏的邻接矩阵上进行运算的，计算代价本就比较大，如果为所有的关系都定义了独立的参数，这会导致模型有大量的参数需要更新，计算复杂度增加。另外，不同的关系需要进行不同的运算，也为图神经网络的并行计算带来了新的困难。（2）知识图谱中存在长尾分布的现象，也就是说大部分关系只有少数的三元组，对于出现频率较少的关系，它们关联的异质参数很难得到较为充分的更新。

作者观察到，在知识图谱中，关系与关系之间并不是完全独立的。在TransCoRe[75]中，研究者把FB15k[1]数据集下使用TransE学习到的关系向量作为列向量组成一个关系矩阵，是向量的维度，是关系的数量，使用SVD进行分解，结果如所示图12，该图来源于作者原论文[75]。图中的横坐标维度是指选取的奇异值数量，纵坐标能量表示选取的奇异值与所有奇异值和的比值。通过结果可以看出，使用20%-30%的奇异值可以表达接近80%的能量（Energy），这说明了不同关系向量之间存在相关性，存在某种共通的低秩结构。

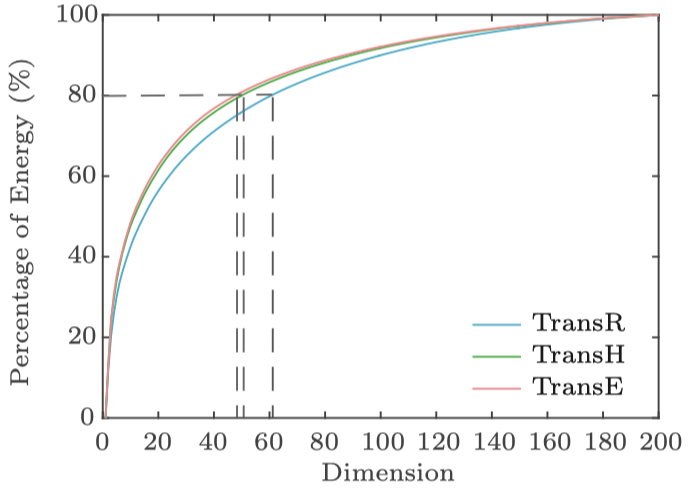


图12 使用SVD对关系矩阵进行分解

受到上面的启发，为了解决关系参数会随着关系数量而爆炸式增长的问题，本文考虑不再为不同的关系都引入完全独立的参数，而是想办法让不同关系之间能够共享信息，同时保留自身信息。具体而言，本文引入了一个外部的网络，叫做超网络（Hypernetwork），它的输出是另一个主网络（异质图神经网络）的参数。数学的定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是每个关系独有的权重向量，用来描述关系异质参数的内部结构信息，是指图神经网络第层；是Hypernetwork中所有关系共享的全局参数，用来显式地捕获关系之间全局的语义信息；是Hypernetwork本身的结构，可以是任意的映射函数，比如线性投影、多层感知器MLP、卷积神经网络CNN、甚至也可以是另外的图神经网络GNN；就是输出的异质参数，具体形式与主网络的模型设计有关。通过上面的定义，实现了不同关系之间的信息共享，无论所需要的异质参数是什么形式，对于关系来说，都被最后压缩为一维的向量，这样就达到了减小模型参数量的目的。

### 4.2.3 消息构造方法

基于上述的超网络，HKGN搭建了一个更加复杂的消息构造过程，具体如图13所示。对于邻居实体，HKGN首先引入了一个关系卷积层来显式地探索邻居实体和关系之间的特征交互：

|  |  |  |
| --- | --- | --- |
|  |  | (4.3) |

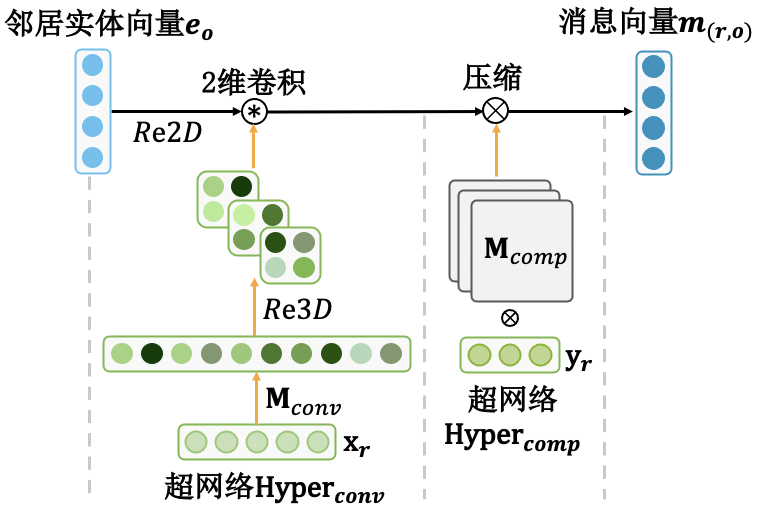


图13 引入超网络的消息构造方法

首先被重组为二维特征图，之后进行一个关系感知的卷积参数，其中是二维卷积核。是激活函数。异质图神经网络网络中常使用线性转换操作，如图14所示。对于来说，它要求的所有元素都参与最后的输出：

|  |  |  |
| --- | --- | --- |
|  |  | (4.4) |

在这种情况下，如果没有得到充分训练的话，不同的输出元素之间的差异就很小。即便是学习到了某个元素比较重要，也可能因为所有元素都会参与输出而降低了该元素的重要性。



图14 一般的线性转换操作

实际上对于输入的特征而言，有可能是只有部分特征比较重要，而另外的特征不重要。本文引入的关系卷积操作的输出单元只需要的部分输入，而不再是整个。能够和的不同区域交互得到最后的输出，增加了实体与关系之间的特征交互，如图15所示。与之前的直接线性转换相比，卷积操作显式地断开了输入单元与输出单元之间的链接，对于不同输出单元来说拥有了不同的输入元素，增大了之间差异。同时，二维的变换使得两个相距比较远的元素也可能位于同一个窗口内，增加了实体不同维度特征之间的交互。

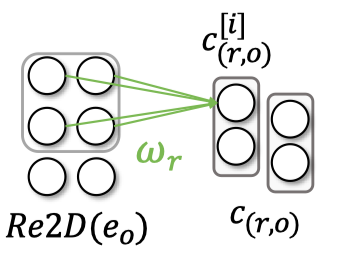


图15 关系卷积操作

为了计算效率，HKGN在一次捕获邻居信息的过程中只进行一次标准的二维卷积操作。在消息构造函数中加入额外的关系卷积层的灵感来源于使用卷积神经网络的知识图谱表示学习算法，由于HKGN的研究重点在于引入Hypernetwork使得能够在知识图谱上构建更加复杂的异质图神经网络，因此没有对卷积层的结构设计进行更深入的探究。事实上，作者注意到任何使用卷积神经网络的思想都可以尝试应用到HKGN当中，例如Inception网络、池化操作、循环卷积等。

接下来，由于卷积操作的输出是三维张量，为了能够让输出能够适用于接下来的网络结构，HKGN对进行了线性变换操作：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

三维张量首先被展开为一维向量，然后由矩阵转换为合适的维度。到目前为止，HKGN为每个关系都定义了权值矩阵和额外的卷积核，如果直接训练学习这么多的参数，可能出现严重的过度参数化问题。

因此，利用在前面小节中提出的超网络来直接生成所需要的参数。首先，对于卷积核，HKGN定义了一个简单的外部网络获取参数：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，，。是特定于关系的向量，用来描述参数中独属于关系的特征，在所有的关系中都共享，用来显示的捕获全局的语义信息，负责将向量转化为三维张量的卷积核。在实践中发现，即使将线性投影简单地替换为一个多层感知器MLP，也没有带来更好的效果，反而加大了训练的难度。一个简单的线性投影操作就已经能够取得较好的结果。

之后，对于权值矩阵，HKGN定义了另外一个外部超网络来获取参数：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，表示张量乘法操作，与的定义类似，是全局共享参数，是特定于关系的一维向量。

### 4.2.4 多关系图聚合方法

与RAGAT使用注意力来进行聚合的方法不同，为了说明引入超网络对于异质图神经网络的效果，HKGN直接使用简单的图卷积操作来实现消息聚合。对于中心实体来说，可能被不同的邻居三元组包围，位于同一邻居关系下的实体，可以看做是位于相同的关系图当中。因此，HKGN将整个知识图谱按照关系拆分为不同的关系子图，然后在不同的关系子图下分别进行图卷积操作。从单个中心实体的角度来看，实体的表示的更新方式为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，表示中心实体在关系的邻居实体集合。HKGN的整体结构如图16所示。

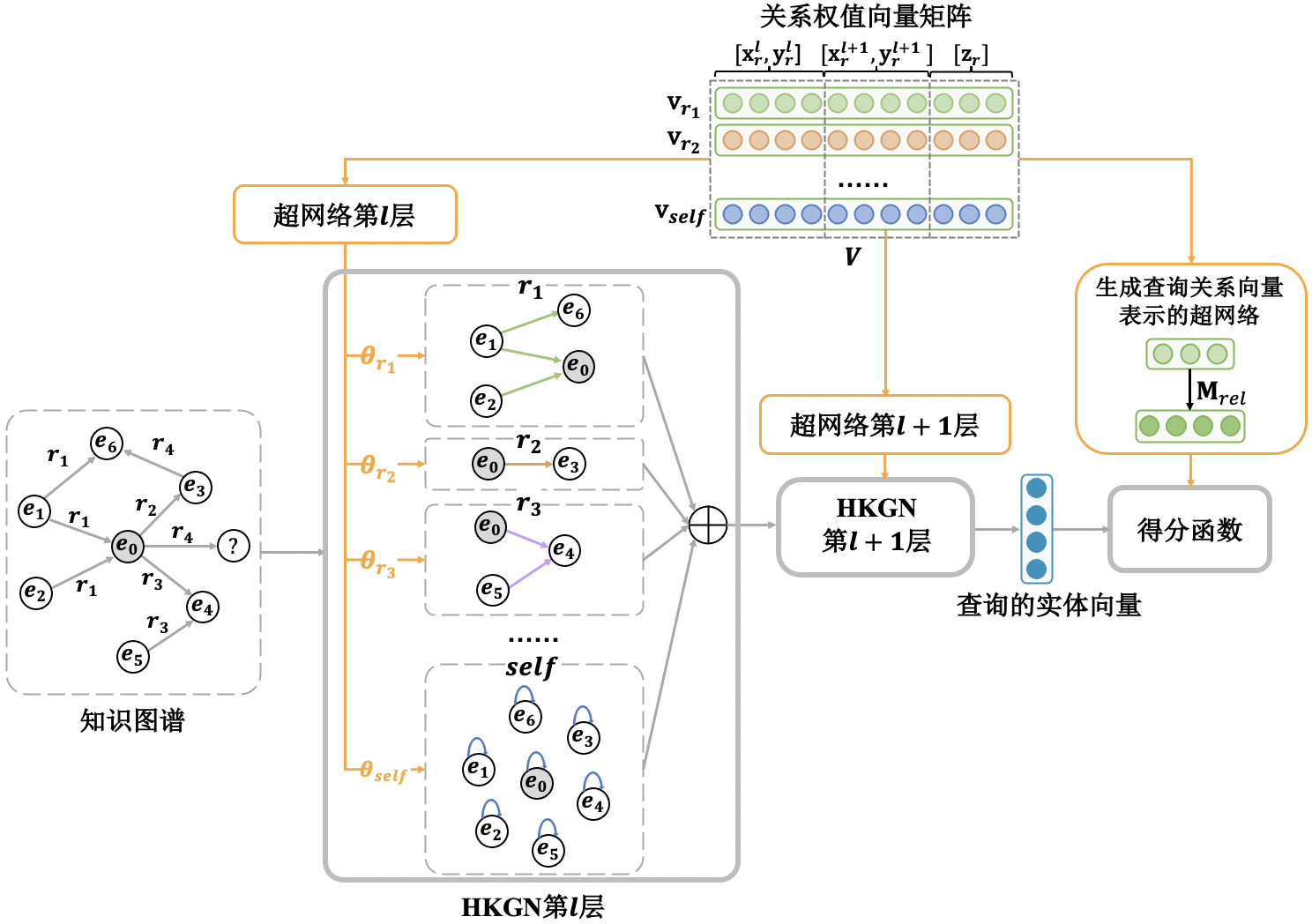


图16 HKGN整体结构

将Hypernetwork应用到MLP或者RNN中时，不同层之间共享相同的网络参数。但是在异质图神经网络中，由于不同阶邻居包含的信息是不同的，比如1阶邻居实体与中心实体的“联系”可能比2阶邻居实体更加紧密。因此与在MLP、CNN当中设计的Hypernetwork不同，在HKGN当中引入的Hypernetwork为了避免不同阶层之间可能存在的信息干扰，为不同图卷积层定义了独立的Hypernetwork。在HKGN中，每一个卷积层的Hypernetwork之间没有直接的联系。

因此，对于关系来说，相关的权重向量实际上是一个分段的向量：

|  |  |  |
| --- | --- | --- |
|  |  | (4.) |

其中的和是在第层的HKGN定义的超网络与。从中可以看出，是独立于实体向量的关系向量，主要用来反映模型在参数空间上关系可能包含的信息。如果模型有外部的特征需要输入，比如文本描述、预训练完毕的向量等，那么还需要学习新的关系向量用于保存外部的特征信息。到目前为止，HKGN的消息聚合过程可以写为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

到目前为止，HKGN获取了来自不同关系邻居的信息，能够用于更新自身的表示。但是还需要将更新后的表示映射到具体的预测任务上，也就是还需要选择合适的解码器。由于解码器通常需要输入目标实体与目标关系的嵌入，而在HKGN中没有定义类似于一般图神经网络中的初始化的关系向量，如果直接拿和作为目标关系的嵌入，输入到解码器当中，可能造成信息的干扰。因此，定义在公式4.9中的是用在一个新的超网络，它用于产生之后在得分函数中需要的目标关系的向量表示，是一个简单的投影操作：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

回顾下目前在HKGN中定义的所有关系参数，~~通过~~引入超网络除去能够实现控制参数量的目的外，还能够带来三方面的优点：（1）关系向量与实体向量的解耦。关系向量不再需要与实体向量保持相同的维度以方便进行后续运算，根据需求的不同，关系向量可以拥有任意合适的维度，不再局限于实体向量的维度选择；（2）捕获关系之间的潜在语义联系。关系与关系之间通过实体相连在一起，超网络通过成为一个独立的网络，有能力实现对这种潜在联系的细粒度建模，例如把关系看做一个新的图，将超网络设计为一个新的图神经网络；同时，不同的超网络学习到的关系向量能够捕获在参数空间上不同的语义信息；（3）输出任意想要的关系参数。例如在模型初始化时，HKGN没有像一般的图神经网络模型为每个关系赋予初始化的嵌入向量便于之后的模型预测，但HKGN仍然可以通过简单的定义新的超网络直接输出想要的关系嵌入。

与RAGAT方法类似，为了能够预测知识图谱中缺失链接的概率，还需要一个得分函数作为解码器输出预测值。由于ConvE相比较于在RAGAT中采用的InteractE是更通用的解码器，因此为了公平比较，尽管在RAGAT中使用InteractE表示出了更好的性能，HKGN仍然选择采用ConvE作为得分函数，假设要预测三元组存在的概率，模型预测函数为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中,是表示经过层聚合以后获得的中心实体表示。HKGN采用交叉熵作为损失函数训练网络参数：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

其中，是预测三元组的实际标签，是预测值。

## 4.4 本章小结

本章对提出的异质知识图谱图神经网络HKGN的设计思路与整体结构进行了详细介绍。首先本章对RAGAT方法以及目前的异质图神经网络中存在的问题进行了说明；其次详细介绍了描述了HKGN的模型结构设计，主要包括：（1）HKGN中使用的数学符号及其含义；（2）将超网络引入到异质图神经网络中的设计思路与数学定义；（3）HKGN的消息构造方法，包括利用超网络获取所需参数和引入关系卷积层增加特征交互；（4） HKGN的多关系图聚合方法，包括超网络在不同图卷积层下的设计思路、超网络与得分函数的结合以及得分函数的选择。

# 第五章 实验与验证

## 5.1 实验设计

### 5.1.1 数据集介绍

为了验证本文提出的模型的有效性，本文在两个著名的知识图谱Freebase[41]和WordNet[42]上进行了实验验证。Freebase是一个描述现实世界的大型数据知识库，其中的数据条目都是使用结构化的形式组织，它在2010年被谷歌收购。Freebase在学术界的多个领域比如搜索、推荐等都得到了直接的应用。Freebase主要描述的内容包括人物、地区、职业、电影等现实世界的常识信息。WordNet是一个描述英文语义信息的字典，它没有按照一般的字典那样组织英文单词，而是把所有的单词组成一个单词网络。它包含的语义信息包括同义、反义、单词概念的上下层等多种单词语义关系。

由TransE[1]方法开始使用Freebase和WordNet的两个子集FB15k和WN18作为验证数据集，在之后的众多方法都是默认使用这两个数据集作为公开标准数据集。然而2018年由ConvE[25]方法指出，在这两个数据集中存在逆关系泄露问题。即在测试集中存在大量与训练集中的三元组一一对应的逆关系，因此ConvE[25]提出可以使用一个简单基于规则的方法就可以在这两个数据集上表现出良好的性能。进一步，ConvE[25]提议使用FB15k-237数据集和WN18RR数据集作为后续研究的标准数据集。FB15k-237和WN18RR是FB15k和WN18数据集去除逆关系之后的版本。本文因此基于FB15k-237数据集和WN18RR数据集进行实验验证。关于这两个公开数据集的统计信息如表7所示。其中，训练集用于训练模型参数，验证集用于超参数调优，测试集用于评估模型性能。

表7 数据集统计信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据集 | 实体数量 | 关系数量 | 训练集数量 | 验证集数量 | 测试集数量 |
| FB15k-237 | 14541 | 237 | 272115 | 17535 | 20466 |
| WN18RR | 40943 | 11 | 86835 | 3034 | 3134 |

### 5.1.2 评估策略

在获得了对于单个三元组是否存在的概率预测之后，需要一种评价策略来定量评估模型性能。最常见的链接预测任务评估策略起源于TransE[1]。链接预测任务实质是一个排序预测任务。TransE[1]的效果评估分为尾实体预测和头实体预测。尾实体预测是对于测试集上的三元组计算当固定时，预估所有的实体作为尾实体时的得分，之后按照得分从大到小排序，计算评估指标。头实体预测与尾实体预测类似，区别在于头实体预测是预测哪个实体适合作为三元组的头实体。

常见的评估指标包括平均排序（Mean Rank，MR），平均倒数排名（Mean Reciprocal Rank, MRR)和Top-K命中率（Hit@K）。下面使用表示所有候选实体数量，表示对于第个三元组预测的正确实体的排序位置。平均排序MR是指所有正确目标实体的平均排序位置，MR越小，正确实体排序越靠前，表示模型预测效果越好。MR的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

MR存在一个问题，那就是忽略了实际上我们对于不同位置的预测效果关注情况是不一样的，我们更加关注位置靠前的预测结果。比如，对于MR来说，如果存在5个预测结果从位置5提升到1，另外5个预测结果从位置1000下降到位置1005，计算出来的MR是一样的。但是很明显和从位置1000下降到位置1005比较起来，我们认为从位置5提升到1是更有意义的，这时的模型应该是更加理想的。

平均倒数排名MRR就解决了这一问题，MRR的计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

MRR使用排序结果的倒数作为预测的准确程度，这样排序越靠前的位置，预测准确度越高，对于MRR指标的贡献越大。与MR不同，MRR越大，表示模型效果越好。

Top-K命中率是计算最后的预测结果中，正确目标实体排序位置命中前K位置的概率，具体计算公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (.) |

对于Top-K命中率来说，结果越大，表示模型预测效果越好。在链路预测实验中，常用的评估指标为Top-1，Top-3和Top-10命中率（Hit@1，Hit@3，Hit@10）。

在进行效果评估的时候还存在另一个问题，以尾实体预测为例，当一定时，除了该三元组原来的正确外，还可能存在其它多个正确的尾实体，在这种情况下，这些正确的尾实体是否应该算作预测正确。这样的决策会影响到最终的指标计算。为了解决这个问题，TransE[1]分为了两种评估策略，未加工的评估策略（raw）和过滤的评估策略（filter），未加工的评估策略把其它正确的尾实体视作错误的预测，这种情况下的评估指标会降低；过滤的评估策略是把其它正确的尾实体过滤，不进入指标的计算当中。最常用的是过滤的评估策略（filter），本文同样采用该评估策略。

### 5.1.3 实验环境

本文实验所采用的实验环境如表8所示。在硬件环境方面，实验采用的处理器为Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz，内存128 G，显卡为GeForce RTX 3090；在软件环境方面，操作系统为Ubuntu，Python版本为3.8.5，使用1.7.0版本的Pytorch[76]作为深度学习框架。

表8 实验环境配置

|  |  |
| --- | --- |
| **环境** | **版本/内容** |
| 显卡 | GeForce RTX 3090 |
| 处理器 | Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz |
| 内存 | 128G |
| 操作系统 | Ubuntu 5.4.0-74 |
| Python | 3.8.5 |
| Pytorch | 1.7.0 |

## 5.2 RAGAT实验结果分析

### 5.2.1 超参数设置

RAGAT模型涉及到的超参数主要包括编码器消息构造函数的选择、图神经网络的层数、多头自注意力数量、初始表示的维度、输出维度、消息Dropout、分批次训练的批次大小、损失函数的学习率、标签平滑比例、解码器卷积核数量等设置。

经过在验证集上的超参数调优，RAGAT模型确定采用的超参数设置如表9所示。

### 5.2.2 对比方法

为了评估RAGAT方法的性能，根据模型的设计思路，主要对比以下的方法：

（1）基于翻译的模型，包括TransE[1]，VR-GCN[33]，TransE-GCN[34]和COMPGCN(Sub)[3]。

（2）基于向量乘法的模型，包括DistMult[17]和COMPGCN(Mult)[3]。

（3）基于循环相关的模型，包括HolE[18]和COMPGCN(Corr)[3]。

（4）基于向量拼接的模型，包括A2N[52]，KBGAT[51]。

（5）基于交叉交互的模型CrossE[21]。

表9 RAGAT超参数设置

|  |  |  |
| --- | --- | --- |
| **超参数** | **对于FB15k-237** | **对于WN18RR** |
| 消息构造函数 | 基于交叉交互 | 基于交叉交互 |
| 图神经网络层数 | 1 | 1 |
| 多头自注意力数量 | 2 | 1 |
| 初始维度 | 100 | 100 |
| 输出维度 | 200 | 200 |
| Dropout | 消息Dropout：0.4  解码器输入特征Dropout：0.4 | 消息Dropout：0.4  解码器输入特征Dropout：0.2 |
| 分批次训练的批次大小 | 1024 | 256 |
| 损失函数学习率 | 0.001 | 0.001 |
| 标签平滑比例 | 0.1 | 0.1 |
| 卷积核大小（InteractE） | 9 x 9 | 11 x 11 |
| 卷积核数量 | 200 x 1 | 200 x 4 |
| 随机排列数量 | 1 | 4 |

### 5.2.3 整体性能分析

表10展示了RAGAT与对比方法在FB15k-237和WN18RR两个数据集下的实验结果。表格中的对比模型的实验结果都是来自于之前发表的学术论文中提供的数据。其中，Rufﬁnelli等人最近对之前的浅层知识图谱表示学习算法从模型架构、训练策略等方面进行了大量的超参调优实验，他们发现如果如果选择合适的超参，很多浅层的方法也可以达到很好的性能[77]。在表10中，TransE和DistMult的结果直接选自Rufﬁnelli等人报告的结果[77]。KBGAT的数据是来自于Sun等人的论文[78]，在该论文中，作者指出KBGAT的评估方法存在评估不严谨，隐性地取优等问题，因此本文直接采用该论文中提供的KBGAT实验数据，而非KBGAT原始论文中的实验结果。从总体来看，RAGAT方法在FB15k-237数据集上，五个指标中的四个指标（MRR，Hit@1，Hit@3，Hit@10）取得了最好的结果；对于WN18RR数据集，在五个指标中的三个指标（MRR，Hit@1，Hit@3）取得了最好结果。这说明了RAGAT模型的整体效果。相比较于KGBAT，A2N这些同样使用了注意力机制的模型，RAGAT取得了更好的结果，这说明了总体上RAGAT关键设计的效果。同样，由于表10中RAGAT采用了CrossE提出的交叉交互概念，RAGAT和CrossE相比较起来取得了显著的提升，这说明了学习知识图谱图结构的有效性。

表10 RAGAT整体性能

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 模型 | FB15k-237 | | | | | WN18RR | | | | |
| MRR | MR | Hit@1 | Hit@3 | Hit@10 | MRR | MR | Hit@1 | Hit@3 | Hit@10 |
| TransE | 0.313 | - | 0.221 | 0.347 | 0.497 | 0.228 | - | 0.053 | 0.368 | 0.520 |
| DistMult | 0.343 | - | 0.250 | 0.378 | 0.531 | 0.452 | - | 0.413 | 0.466 | 0.530 |
| VR-GCN | 0.248 | - | 0.159 | 0.272 | 0.432 | - | - | - | - | - |
| CrossE | 0.299 | - | 0.211 | 0.331 | 0.474 | - | - | - | - | - |
| KBGAT | 0.157 | 270 | - | - | 0.331 | 0.412 | **1921** | - | - | 0.554 |
| TransE-GCN | 0.315 | - | 0.229 | 0.324 | 0.477 | 0.233 | - | 0.203 | 0.338 | 0.508 |
| A2N | 0.317 | - | 0.232 | 0.348 | 0.486 | 0.45 | - | 0.42 | 0.46 | 0.51 |
| COMPGCN(Sub) | 0.352 | 199 | - | - | 0.530 | - | - | - | - | - |
| COMPGCN(Mult) | 0.353 | 216 | - | - | 0.323 | - | - | - | - | - |
| COMPGCN(Corr) | 0.355 | 197 | 0.264 | 0.39 | 0.535 | 0.479 | 3533 | 0.443 | 0.494 | 0.546 |
| **RAGAT** | **0.365** | 199 | **0.273** | **0.401** | **0.547** | **0.489** | 2390 | **0.452** | **0.503** | 0.562 |

### 5.2.4 关键设计分析

针对本文提出的两个关键设计：关系感知消息构造方法和基于注意力的消息构造方法，本文进行了三方面的对比试验来验证关键设计的效果。（1）不同关系感知消息构造函数变体的对比实验，验证分析本文提出的不同消息构造函数变体之间的差异。（2）引入关系感知参数的对比实验，验证RAGAT模型引入关系特定参数实现关系感知的有效性。（3）消融实验，通过消融关系感知、方向感知等关键结构验证设计有效性。

（1）不同关系感知消息构造函数变体的对比实验

为了评估本文提出的五种消息构造函数变体的性能，在实验中使用了ConvE和InteractE两种模型作为解码器，在FB15k-237数据集上的实验结果总结在表11中。

表11 关系感知消息构造函数变体的实验结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 解码器(=Y) | ConvE | | | | | InteractE | | | | |
| 编码器X | MRR | MR | Hit@1 | Hit@3 | Hit@10 | MRR | MR | Hit@1 | Hit@3 | Hit@10 |
| Trans-GAT+Y | 0.363 | 163 | 0.272 | 0.397 | 0.545 | 0.363 | 163 | 0.271 | 0.398 | 0.546 |
| Mult-GAT+Y | 0.362 | 170 | 0.271 | 0.398 | 0.546 | 0.363 | 202 | 0.272 | 0.398 | 0.548 |
| Corr-GAT+Y | 0.362 | 191 | 0.271 | 0.395 | 0.544 | 0.364 | 192 | 0.272 | 0.400 | 0.547 |
| Concat-GAT+Y | 0.363 | 161 | 0.270 | 0.398 | 0.546 | 0.364 | 194 | 0.272 | 0.400 | 0.547 |
| Cross-GAT+Y | 0.363 | 194 | 0.271 | 0.397 | 0.544 | 0.365 | 199 | 0.273 | 0.401 | 0.547 |

在表11中的X+Y，X表示编码器，Y表示解码器。首先，从解码器的角度看，使用InteractE的方法比使用ConvE的方法表现出了更好的性能，在本文的实验过程中，ConvE和InteractE拥有相同的参数量、相同的输入、相同的训练策略，这样的实验结果说明了InteractE对于ConvE改进的有效性。其次，从编码器的角度看，总体而言，采用基于特征交互消息函数的编码器Cross-GAT取得了最好的结果。同时，Concat-GAT和Corr-GAT取得了次优的结果。这说明更加灵活的消息构造函数倾向于拥有更好的适应性，这一点与COMPGCN方法中对于消息构造函数的实验结论一致。同时，对比各项变体函数，可以看出来它们之间的差异已经相对不太明显，这说明了目前广泛存在于不同模型中，无参数的组合实体与关系性能受限，下一步的改进方向应该是考虑如何引入合适的额外参数提取实体和关系之间的联系。

（2）引入关系感知参数的对比实验

图17至图20展示了引入关系感知参数在FB15k-237上不同评估指标的实验结果。

在图中，展示了两种不同的RAGAT变体：

* X+Y：正常的RAGAT模型，具有关系感知参数。X是编码器，Y是解码器。
* X+Y-w/o ：移除关系感知参数之后的RAGAT模型。

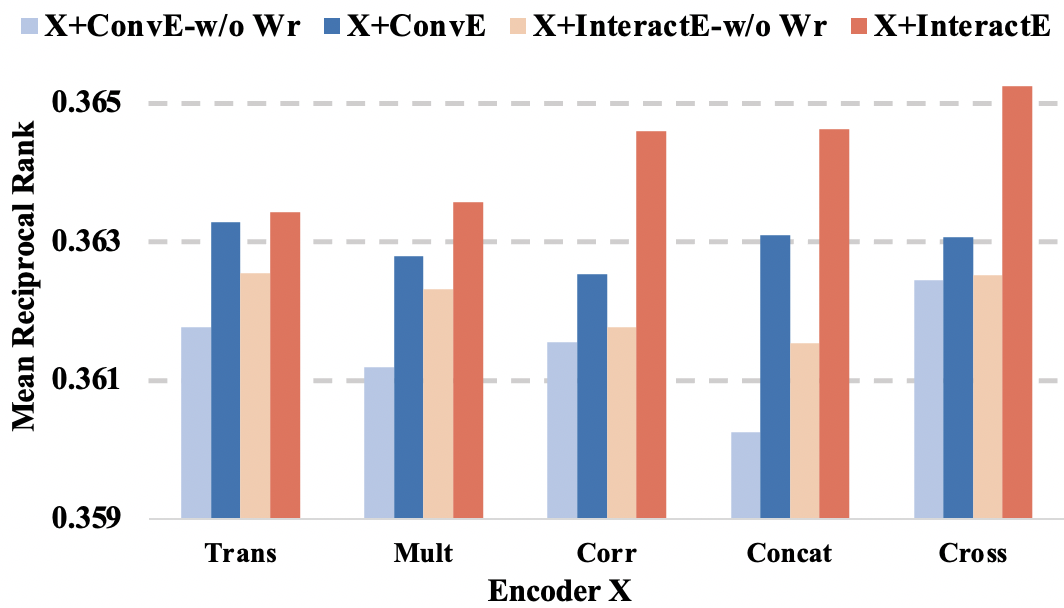


图17 引入关系感知参数在FB15k-237上的MRR结果

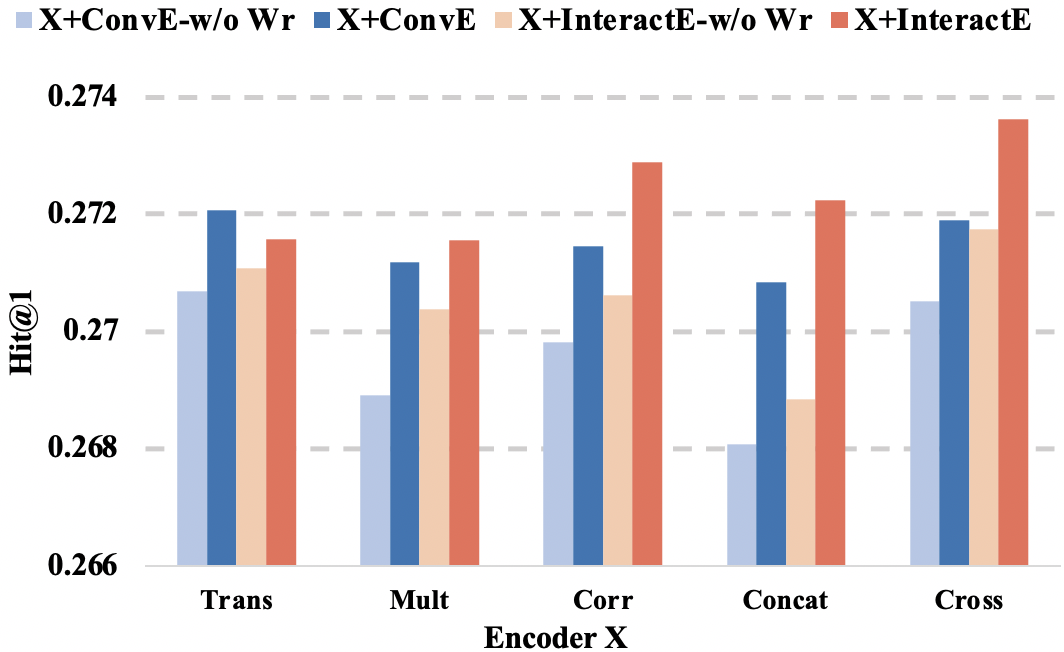


图18 引入关系感知参数在FB15k-237上的Hit@1结果

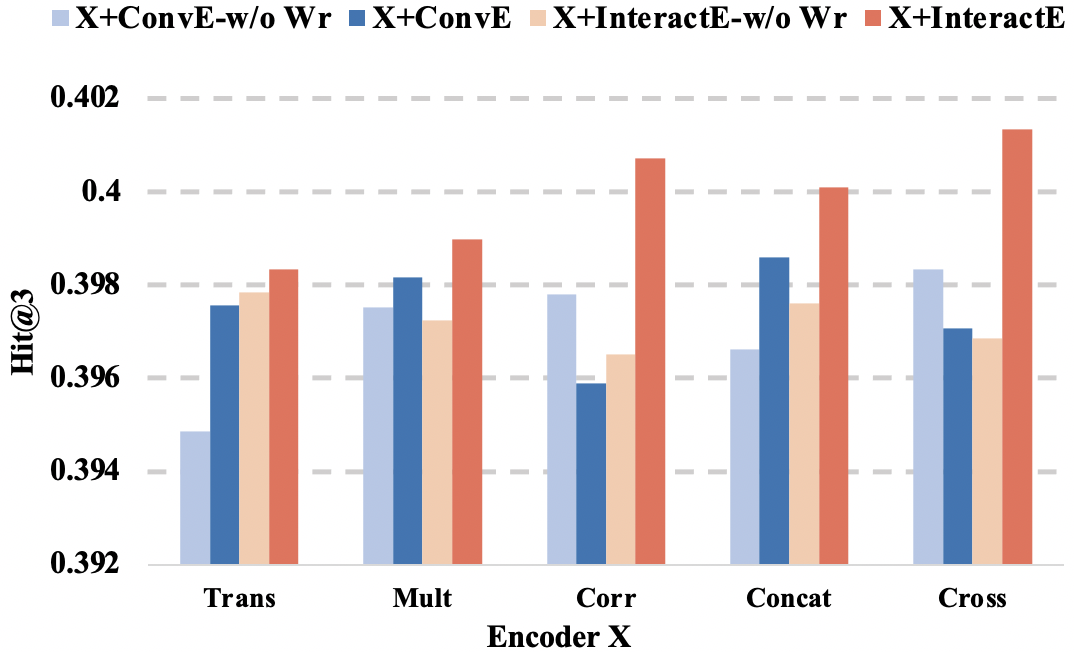


图19 引入关系感知参数在FB15k-237上的Hit@3结果

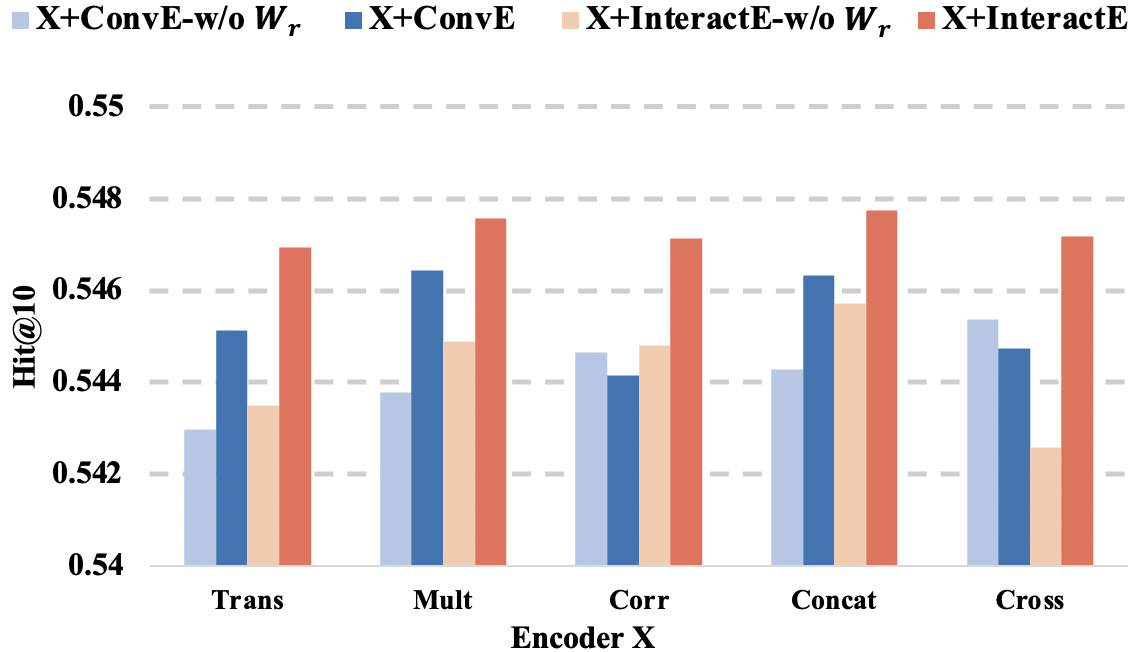


图20 引入关系感知参数在FB15k-237上的Hit@10结果

在移除关系感知参数之后，此时的消息构造函数对应于在之前各类基于图神经网络的知识表示学习算法中广泛应用的方法。通过实验结果可以看出，引入关系特定参数后，总是取得了更好的实验性能。这说明了本文提出的实现关系感知的关键设计的有效性。另外，可以观察到Corr和Concat这种相对更复杂的组合方法，在没有实现关系感知之前甚至效果比Trans和Mult这样简单的方法还要差。这说明可能对于复杂的组合方法来说，引入关系感知参数有可能获得更大的效果提升。

（3）消融实验

为了进一步验证各项关键设计的效果，本文基于表现出最好性能的解码器Cross-GAT上进行消融实验，实验结果如表12所示。通过实验可以看出来，如果没有关系感知或者没有关系的方向感知，都会造成模型的性能下降，这说明了在关系感知消息构造方法中实现关系感知的重要性；同时，如果在消息聚和时，没有实现对于关系的多方向感知，同样会出现模型效果降低。另外，针对本文提出的基于特征交叉交互的Cross消息构造函数，在没有实现关系对实体的交互和实体对关系的交互时，都会出现模型效果的下降，这验证了实现特征交叉的有效性。

表12 消融实验结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | MRR | MR | Hit@1 | Hit@3 | Hit@10 |
| RAGAT | 0.365 | 196 | 0.273 | 0.401 | 0.547 |
| 没有关系感知 | 0.362 | 168 | 0.271 | 0.396 | 0.542 |
| 没有方向感知 | 0.362 | 199 | 0.270 | 0.398 | 0.544 |
| 没有关系对实体的交互 | 0.363 | 201 | 0.272 | 0.398 | 0.548 |
| 没有实体对关系的交互 | 0.363 | 198 | 0.272 | 0.398 | 0.545 |
| 没有注意力机制 | 0.362 | 204 | 0.271 | 0.397 | 0.544 |

## 5.3 HKGN实验结果分析

### 5.3.1 超参数设置

HKGN模型涉及到的超参数主要包括不同超网络的关系向量的维度、关系卷积层的卷积核数量、卷积核大小、图神经网络的层数、输出维度、消息Dropout概率、分批次训练的批次大小、损失函数的学习率、标签平滑比例等超参数。经过在验证集上的超参数调优，本文确定采用的HKGN超参数设置如表13所示。

表13 HKGN超参设置

|  |  |  |
| --- | --- | --- |
| **超参数** | **对于FB15k-237** | **对于WN18RR** |
| 超网络  输入向量维度 | 100 | 100 |
| 超网络  输入向量维度 | 2 | 2 |
| 超网络  输入向量维度 | 100 | 100 |
| 关系卷积核大小 | 3x3 | 3x3 |
| 卷积核数量 | 32 | 32 |
| 图神经网络层数 | 2 | 1 |
| 初始维度 | 100 | 100 |
| 输出维度 | 200 | 200 |
| 消息Dropout概率 | 0.4 | 0.4 |
| 分批次训练的批次大小 | 1024 | 256 |
| 损失函数学习率 | 0.001 | 0.001 |
| 标签平滑比例 | 0.1 | 0.1 |

### 5.3.2 对比方法

为了证明HKGN方法的有效性，本文选择了不同分类的知识图谱表示学习方法中，最新出现的研究成果作为对比基准（Baselines）。主要选择的方法如下：

（1）浅层的知识图谱表示学习算法：具有较低的时间与计算复杂度，包括TransE[1]、DistMult[17]和DualE[79]（无类型约束先验）；

（2）基于卷积的知识图谱表示学习算法：包括ConvE[25]、ConvR[27]、HypER[63]和CoPER[80];

（3）基于图神经网络的知识图谱表示学习算法：包括R-GCN[30]、SACN[31]、VR-GCN[33]、A2N[52]、KBGAT[51]、COMPGCN[3]、HRAN[81]以及在第三章提出的RAGAT方法（使用ConvE解码器）；

特别的，在上述方法中有几个关键的对比模型与HKGN关联较为紧密，下面对它们进行详细说明。

（1）HypER[63]：第一个将超网络引入知识图谱表示学习领域的模型，使用超网络生成一维卷积核，然后在实体向量上进行卷积操作，最后经过线性转换得到预测值；

（2）CopER[80]：使用关系作为上下文向量，通过一个参数生成器获得主网络（ConvE[25]和MINERVA[82]）所需参数；虽然在论文中作者没有提及超网络的概念，但是在本质上和超网络是一样的。本文将CopER作为超网络在知识图谱表示学习领域最新的应用进展，将其作为关键的对比模型之一。

（3）HRAN[81]：异质图神经网络在知识图谱表示学习算法领域的最新进展，通过设计一种关系的异质注意力以及新的解码器，获得了更好的性能。

### 5.3.3 整体性能分析

表14展示了HKGN与基准模型的对比结果。在表14中，TransE、DistMult和ConvE的结果直接选自Rufﬁnelli等人经过大量调参后选择的最优结果[77]。Sun等人调查了在KBGAT中出现的不正确的评估策略以及测试集泄露的问题，因此，KBGAT的结果直接来自于Sun等人修复错误之后的新结果[78]。除此之外，本文作者在调研时发现，CopER在论文中报告的性能指标是只报告了尾实体预测场景下的结果，因此，本文基于CopER的源码（https://github.com/otiliastr/coper）重新进行了实验，获得了更公平的评估结果。从表14中可以观察到：（1）HKGN在两个数据集上，都在所有对比方法中取得了最好的结果，这证明了HKGN模型的整体性能。与HRAN这个最新出现的异质图神经网络方法对比起来，HKGN取得了更好的结果。这证明了HKGN在具有强异质性的知识图谱中学习丰富的语义信息和建模图结构的能力。（2）HKGN比HypER和CopER这两个利用超网络学习知识图谱表示的方法表现更好，这证明了图结构在链路预测任务中能够提供很重要的信息。

表14 HKGN整体性能

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | FB15k-237 | | | | | WN18RR | | | | |
| 算法 | MRR | MR | Hits@1 | Hits@3 | Hits@10 | MRR | MR | Hits@1 | Hits@3 | Hits@10 |
| TransE | 0.313 | - | 0.221 | 0.347 | 0.497 | 0.228 | - | 0.053 | 0.368 | 0.520 |
| DistMult | 0.343 | - | 0.250 | 0.378 | 0.531 | 0.452 | - | 0.413 | 0.466 | 0.530 |
| DualE | 0.330 | - | 0.237 | 0.363 | 0.518 | 0.482 | - | 0.440 | 0.500 | 0.561 |
| ConvE | 0.339 | 244 | 0.248 | 0.369 | 0.521 | 0.442 | 4187 | 0.411 | 0.451 | 0.504 |
| ConvR | 0.350 | - | 0.261 | 0.385 | 0.528 | 0.475 | - | 0.443 | 0.489 | 0.537 |
| HypER | 0.341 | 250 | 0.252 | 0.376 | 0.520 | 0.465 | 5798 | 0.436 | 0.477 | 0.522 |
| CoPER-ConvE | 0.320 | 390 | 0.234 | 0.351 | 0.491 | 0.442 | 5315 | 0.419 | 0.450 | 0.488 |
| R-GCN | 0.248 | - | 0.153 | 0.258 | 0.414 | - | - | - | - | - |
| SACN | 0.350 | - | 0.26 | 0.39 | 0.54 | 0.47 | - | 0.43 | 0.48 | 0.54 |
| VR-GCN | 0.248 | - | 0.159 | 0.272 | 0.432 | 0.45 | - | 0.42 | 0.46 | 0.51 |
| A2N | 0.317 | - | 0.232 | 0.348 | 0.486 | - | - | - | - | - |
| KBGAT | 0.157 | 270 | - | - | 0.331 | 0.412 | 1921 | - | - | 0.554 |
| COMPGCN | 0.355 | 197 | 0.264 | 0.39 | 0.535 | 0.479 | 3533 | 0.443 | 0.494 | 0.546 |
| HRAN | 0.355 | **156** | 0.263 | 0.39 | 0.541 | 0.479 | 2113 | **0.45** | 0.494 | 0.542 |
| RAGAT-ConvE | 0.363 | 194 | 0.271 | 0.397 | 0.544 | 0.481 | 2390 | 0.440 | 0.499 | 0.556 |
| HKGN | **0.365** | 171 | **0.272** | **0.402** | **0.552** | **0.487** | 2468 | 0.448 | **0.505** | **0.561** |

更进一步，本文探究了HKGN对于不同关系分类的预测效果。在实验中，对FB15k-237数据集，根据训练集中关系头/尾实体的对应数量进行分类：一对一（1-1）关系，一对多（1-N）关系，多对一（N-1）关系和多对多（N-N）关系，然后划分测试集，评估具体的实验结果。具体划分方法与在TransH[11]中采用的划分方法一致：对于每个关系，首先计算该关系的平均头实体对应的尾实体数量，计算平均尾实体对应的头实体数量。如果并且，关系就属于一对一（1-1）关系；如果并且，关系属于多对多（N-N）关系；如果并且，关系属于多对一（N-1）关系；如果并且，关系属于一对多（1-N）关系。对FB15k-237划分后的数据分布统计如表15所示。

表15 FB15k-237关系分类统计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 关系分类 | 一对一（1-1） | 一对多（1-N） | 多对一（N-1） | 多对多（N-N） |
| 关系数量 | 17 | 26 | 86 | 108 |
| 训练集边数量 | 4278 | 12536 | 50635 | 204666 |
| 验证集边数量 | 167 | 1043 | 3936 | 12389 |
| 测试集边数量 | 192 | 1293 | 4508 | 14473 |

根据划分后的关系分类评估的实验结果如表16所示。其中，COMPGCN以及CopER的实验数据是使用它们的开源代码重新评估得到的。通过实验可以看出来，HKGN更加擅长捕获1-N，N-1和N-N这样更复杂的关系，这可能是因为更复杂的关系可能拥有更丰富的图结构信息。同时可以看出，HKGN对于1-1关系的预测效果较差，这个可能是因为1-1关系自身的信息被来自其它关系的信息干扰了。这也说明了HKGN或许可能通过设计更灵活的结构让信息在关系之间流通，比如借助门机制、残差链接等结构，或许可以进一步提升模型的性能。但是在知识图谱中，往往多对多这样复杂关系数量更多，比如表15中统计的1-1关系边数量仅仅占全部边数量的1.5%左右，这也解释了为什么HKGN在总体上取得了更好的效果。

表16 HKGN关系分类的实验结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 对比模型 | | COMPGCN | | | HKGN | | | CopER | | |
| 预测任务 | 关系分类 | MRR | MR | Hit@10 | MRR | MR | Hit@10 | MRR | MR | Hit@10 |
| 头实体  预测 | 1-1 | **0.489** | 220 | **0.609** | 0.398 | **175** | 0.583 | 0.429 | 722 | 0.531 |
| 1-N | 0.477 | 82 | **0.668** | **0.484** | **75** | 0.664 | 0.438 | 346 | 0.607 |
| N-1 | 0.102 | 838 | 0.185 | **0.129** | **442** | **0.241** | 0.074 | 1055 | 0.138 |
| N-N | 0.277 | 175 | 0.482 | **0.291** | **158** | **0.498** | 0.244 | 312 | 0.432 |
| 尾实体  预测 | 1-1 | **0.449** | 281 | 0.578 | 0.401 | **200** | **0.598** | 0.425 | 634 | 0.52 |
| 1-N | 0.079 | 818 | 0.16 | **0.080** | **724** | **0.168** | 0.059 | 1379 | 0.113 |
| N-1 | 0.768 | **35** | 0.877 | **0.774** | 43 | **0.884** | 0.744 | 152 | 0.85 |
| N-N | 0.390 | 109 | 0.616 | **0.399** | **97** | **0.624** | 0.349 | 243 | 0.57 |
| 平均结果 | 1-1 | **0.469** | 251 | **0.593** | 0.400 | **188** | 0.591 | 0.427 | 678 | 0.526 |
| 1-N | 0.278 | 450 | 0.414 | **0.282** | **400** | **0.416** | 0.249 | 862 | 0.36 |
| N-1 | 0.435 | 437 | 0.531 | **0.452** | **242** | **0.562** | 0.409 | 604 | 0.494 |
| N-N | 0.334 | 142 | 0.549 | **0.345** | **128** | **0.561** | 0.297 | 278 | 0.501 |

### 5.3.4 关键设计分析

本文分别从关系卷积层的效果和超网络对于模型参数与性能的影响两个方面探究HKGN关键设计的有效性。

（1）关系卷积层的影响实验

为了探究关系卷积层对于模型性能的影响，本文首先通过消融实验，对比HKGN去除关系卷积层与没有去除关系卷积层的性能结果，结果如表17所示，表中的wo Conv.表示移除关系卷积层之后的模型。从表中可以看出，HKGN的关系卷积层在两个数据集上都起到了对模型性能的提升作用。

表17 HKGN无关系卷积层的实验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据集 | 模型 | MRR | MR | Hit@1 | Hit@3 | Hit@10 |
| FB15k-237 | Full HKGN | 0.365 | 191 | 0.272 | 0.402 | 0.552 |
| wo Conv. | 0.355 | 172 | 0.264 | 0.390 | 0.536 |
| WN18RR | Full HKGN | 0.487 | 2468 | 0.448 | 0.505 | 0.561 |
| wo Conv. | 0.459 | 2639 | 0.410 | 0.479 | 0.553 |

其次，为了更进一步的探究关系卷积层对不同实体的影响，本文将FB15k-237和WN18RR的实体按照度（邻居数量）划分到不同的区间内，统计结果如表18所示。具体划分过程是，统计训练集下每个实体的入边和出边的数量，然后依次将不同实体划分到不同度区间内，同时将测试集中头实体具有对应度区间的测试三元组也划分到相应的度区间中去。从表中可以看出来，两个数据集的实体度分布有很大的差异性，FB15k-237数据集的邻居数量更多，而WN18RR数据集的实体邻居更为稀疏。

表18 数据集实体度分布的统计

|  |  |  |
| --- | --- | --- |
| FB15k-237数据集 | | |
| 实体度区间 | 实体数量 | 测试集三元组数量 |
| [0, 100) | 13839 | 16385 |
| [100, 200) | 496 | 2055 |
| [200, 300) | 76 | 525 |
| [300, 400) | 44 | 493 |
| [400, 5000) | 35 | 389 |
| [500, 1000) | 35 | 373 |
| [1000, max) | 16 | 246 |
| 总计 | 14541 | 20466 |
| WN18RR数据集 | | |
| 实体度区间 | 实体数量 | 测试集三元组数量 |
| [0, 10) | 38102 | 2595 |
| [10, 25) | 2497 | 417 |
| [25, 50) | 243 | 47 |
| [50, 100) | 65 | 29 |
| [100, 500) | 36 | 46 |
| 总计 | 40943 | 3134 |

图21和图22分别展示了FB15k-237和WN18RR数据集中，HKGN与移除关系卷积层后的Hit@10指标对比结果。

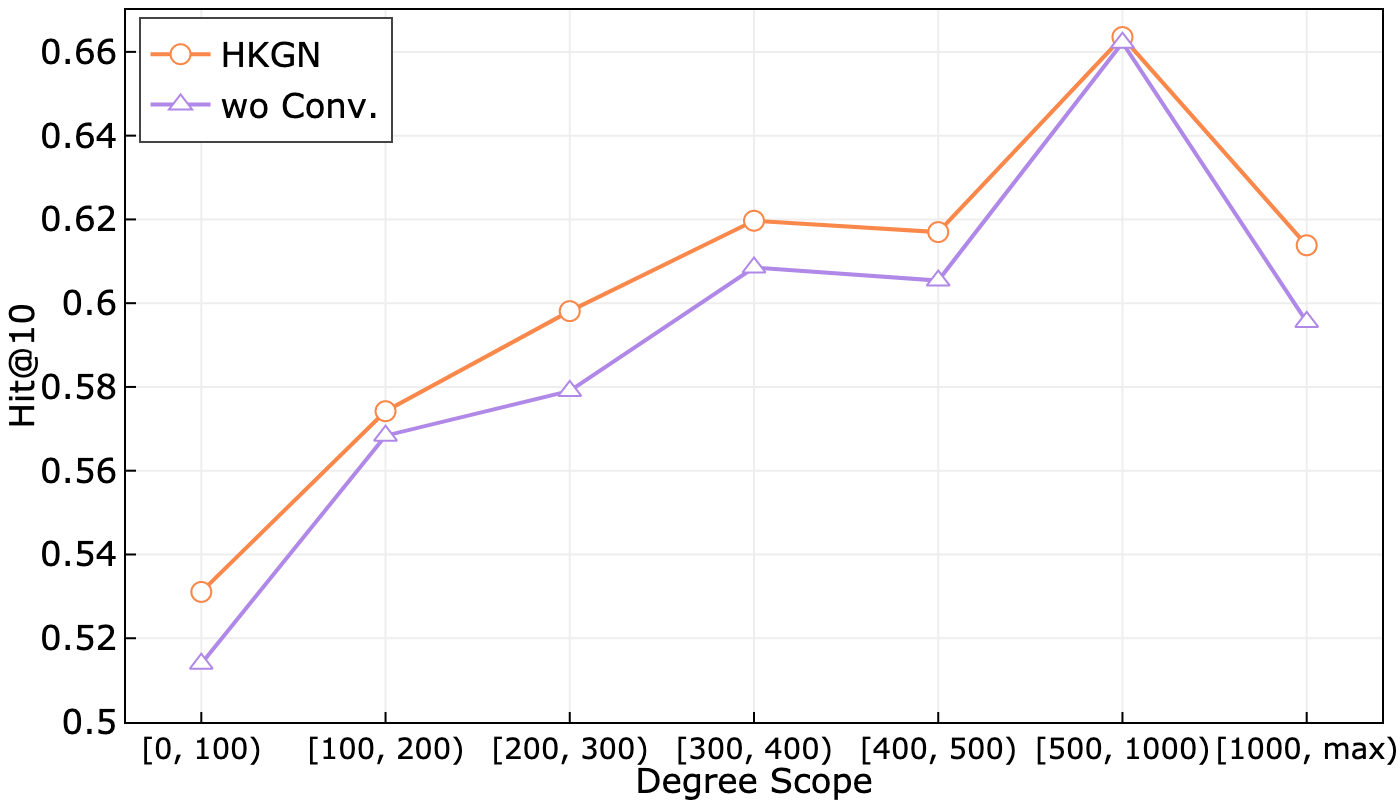


图21 关系卷积层对FB15k-237中不同度实体的影响

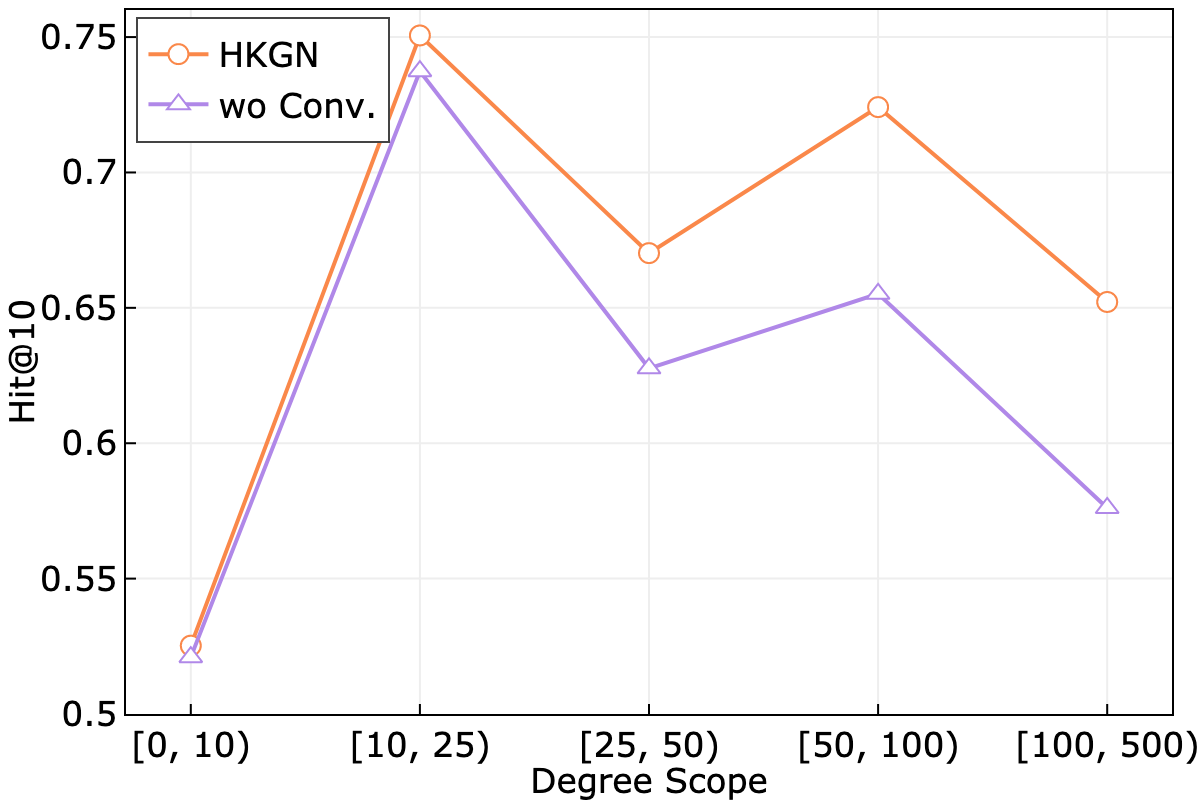


图22 关系卷积层对WN18RR中不同度实体的影响

从图中可以看出，首先，在实体度较小的时候，模型性能普遍比较差，随着实体度的增加，更多的邻居实体可以提供更丰富的信息，但也注意到，在FB15k-237数据集实体度超过1000的时候，WN18RR数据集实体度超过25的时候模型的效果出现了下降的现象，这可能是由于过多的邻居中反而存在过多的噪音，有用的邻居信息反而被干扰淹没了；其次，在移除关系卷积层后，模型在所有的实体区间内，都出现了效果的下降，这证明了关系卷积层在捕获邻居实体与邻居关系特征交互中的作用，验证了关键设计的有效性。同时需要注意到，这两个数据集的效果分布出现了比较大的差异，特别是在WN18RR中，波动很强烈，作者推测是因为在WN18RR中实体度较大的实体数量过少，导致评估结果很容易不稳定的情况。

（2）超网络的影响实验

为了探究超网络对于HKGN模型参数与性能的影响，本文对不同的超网络结构进行了消融实验，结果如表19所示。表19中的Hyper代表所有的超网络、代表关系卷积层的超网络、代表维度压缩层的超网络、代表生成目标关系向量的超网络、#param.表示模型的参数量。从表可以看出，在消融掉所有的超网络后，在两个数据集下都会出现模型性能的下降以及参数量的增加，这表明了超网络能够在控制由知识图谱中关系数量带来的异质参数量的同时，还能够为模型的性能带来提升。特别是在FB15k-237数据集中，消融掉超网络会使模型的参数量会急剧变为原来参数量的117倍，达到了近3.9亿的参数量。更值得注意的是，这仅仅是在实验环境中，用在FB15k-237这个仅仅有237种关系的数据集中需要的参数量，而实际中的大规模知识图谱会拥有远远比这多的关系数量。这进一步证明了将异质图神经网络网络应用到知识图谱中时，需要考虑模型复杂度的必要性，这也是本研究的意义所在。

表19 HKGN超网络的消融实验结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ablation | FB15K-237 | | | WN18RR | | |
| MRR | Hit@10 | #param. | MRR | Hit@10 | #param. |
| Full HKGN | 0.365 | 0.552 | 3.31M | 0.487 | 0.561 | 4.98M |
| wo Hyper | 0.358 | 0.544 | 389.54M | 0.483 | 0.558 | 13.52M |
|  | 0.362 | 0.549 | 3.38M | 0.485 | 0.557 | 4.96M |
|  | 0.36 | 0.545 | 389.46M | 0.487 | 0.557 | 13.58M |
|  | 0.364 | 0.552 | 3.32M | 0.484 | 0.557 | 4.94M |

### 5.3.5 超参数影响分析

本文对HKGN中各个超网络的输入关系向量的维度对于模型性能（MRR指标）的影响也进行了实验。图23表示的输入关系向量的维度在FB15k-237和WN18RR数据集上对性能的影响，图24表示的输入向量的维度对性能的影响，图25表示的输入向量的维度对模型性能的影响。从三张图中可以看出来，总体而言，输入的关系向量维度大小虽然会影响模型性能，但是变化的大小并不显著，MRR的变化范围集中在千分位上的波动，这证明模型对关系向量大小不敏感，证明了模型设计的超网络的鲁棒性。另一方面可以看出，基本上随着选择的维度，模型性能会出现先上升后下降的趋势。这是因为在关系向量维度还比较小的时候，模型的建模能力会受限，而当关系向量维度逐渐增加到一定程度后，模型整体的参数量和计算复杂度上升，性能下降。

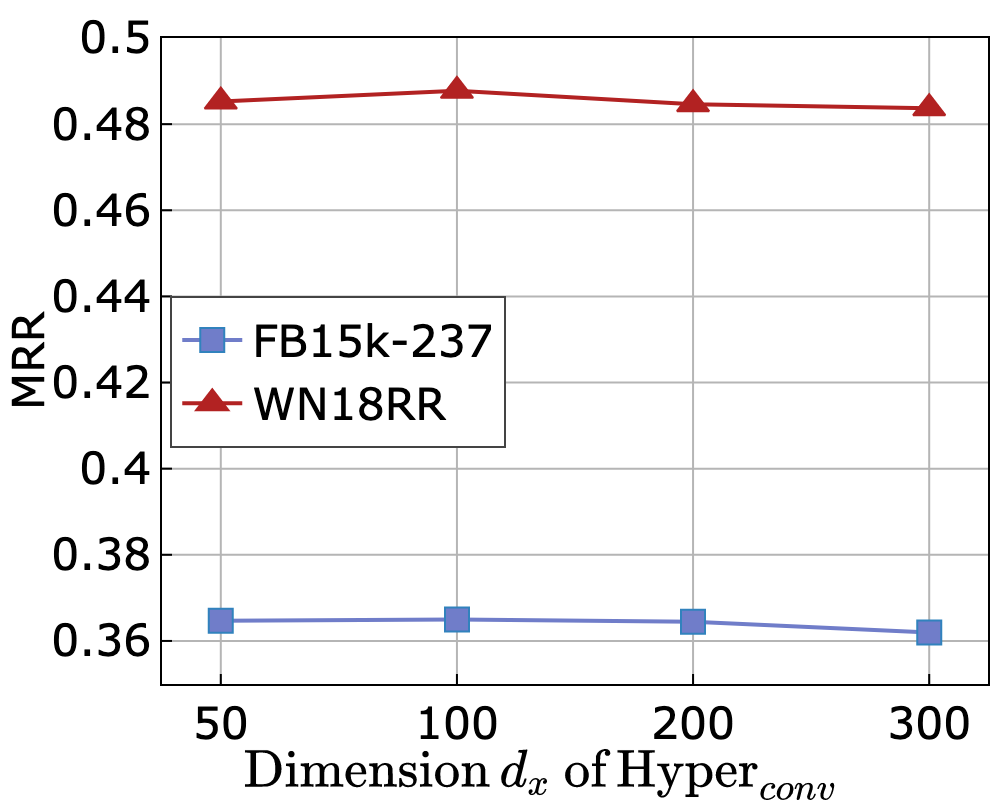


图23 超参数对模型性能的影响

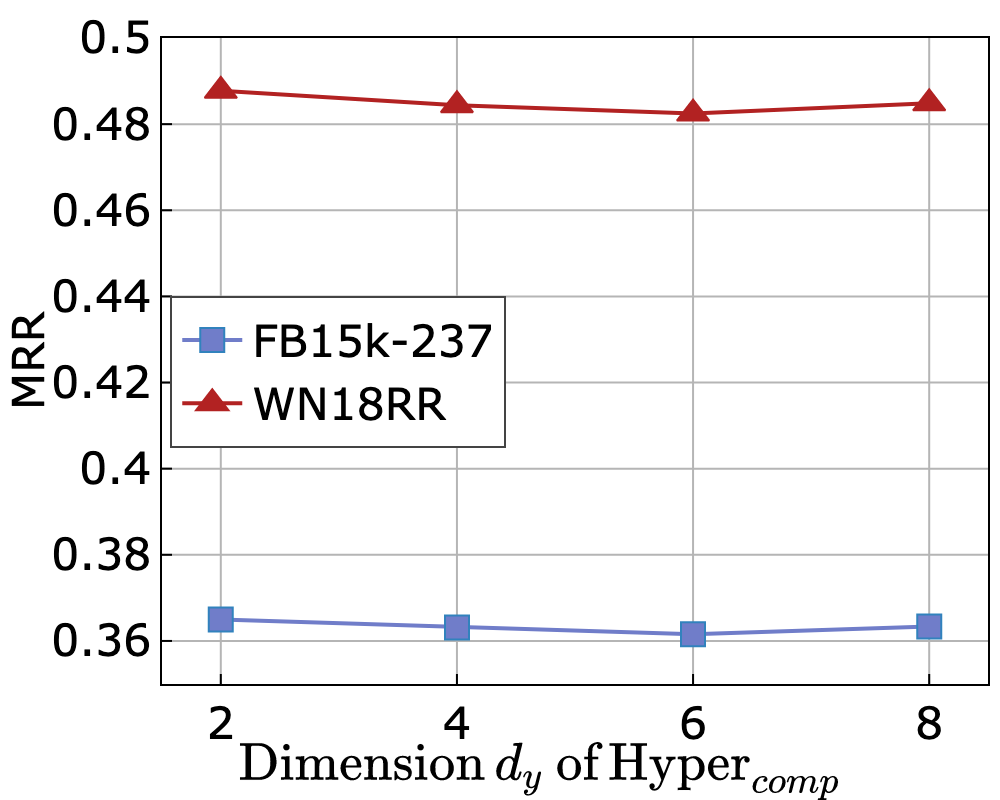
****

图24 超参数对模型性能的影响

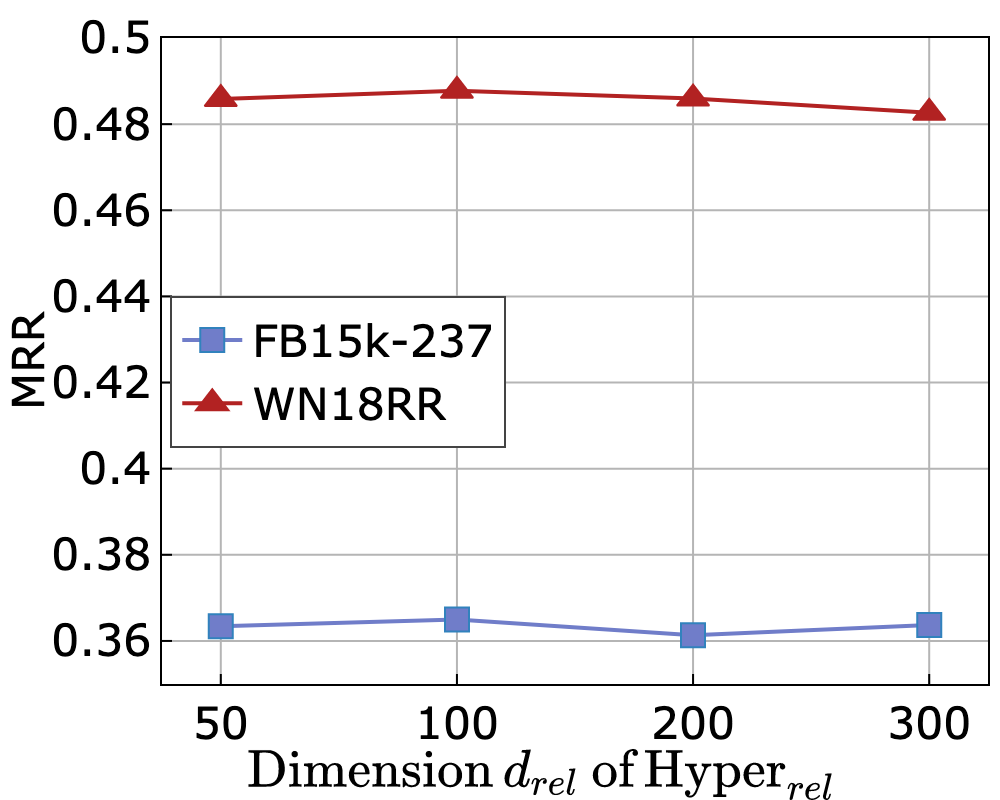


图25 超参数对模型性能的影响

## 5.4 本章小结

本章对于本文提出的两个模型RAGAT和HKGN的实验部分进行了介绍。首先，详细描述了实验的基础设计，包括数据集基本情况、模型的评估策略和实验环境；接下来，从整体性能分析和关键设计分析两个方面详细说明了对RAGAT模型的探究实验；最后，对于HKGN模型涉及的三方面实验：整体性能分析、关键设计分析和超参数影响分析进行了详细描述。

# 总结与展望

随着2017年图卷积神经网络GCN的提出，图神经网络在各个领域大放异彩，受到了众多的关注，在知识图谱领域同样有很多融合图神经网络的算法被提出来，取得了较好的效果。本文就如何使图神经网络能够更充分地学习知识图谱中的关系语义这一问题进行了深入的探究，提出了两种新型的图神经网络模型：关系感知图注意力网络RAGAT和知识图谱异质图神经网络HKGN。RAGAT方法在使用图注意机制的统一架构下，通过为不同关系直接训练额外的参数提升了模型对关系语义的建模能力；但是RAGAT方法定义的关系参数是对角矩阵，如果将对角矩阵拓展为其它更复杂的参数形式又会引发过度参数化问题；为了解决这一问题，HKGN不再直接定义独立的关系参数，而是通过引入所有关系共享的超网络，动态生成所需的关系参数，从而较好的控制了模型参数量。

本文的主要工作内容包括以下几方面：

（1）从直接引入额外的关系特定参数的思想出发，提出了RAGAT表示学习算法。该方法在消息构造过程中，为了解决最近出现的方法无法建模知识图谱关系异质性的问题，提出了关系感知消息构造方法。该方法通过引入关系特定参数，改进目前已经出现的四种消息构造方法，同时提出了一种新的基于交叉交互的消息构造方法；在消息聚合过程中，针对目前基于注意力聚合方式的模型忽略了关系方向信息的缺点，提出了基于注意力的多方向消息聚合方法；同时为了公平评估，引入了ConvE和InteractE两种不同的卷积神经网络解码器。

（2）从关系参数生成的思想出发，提出了HKGN表示学习算法。该方法通过将超网络引入异质图神经网络，尝试解决关系类型数量过多会造成异质参数量过多的问题，能够将任意形式的异质参数都压缩为一维的向量。在此结构的基础上，在消息构造阶段，增加关系卷积层，实现了更细粒度的关系语义建模，提升了模型性能；在消息聚合阶段，通过聚合不同关系子图的消息，捕获更丰富的语义信息。HKGN整体结构的所有异质参数都是由超网络生成，从而让模型更加适合于边类型数量多的异质图。

（3）将本文提出的知识图谱表示学习算法与多种不同类型的表示学习算法进行对比，在FB15k-237和WN18RR公开数据集上进行实验，证明了总体结构设计的性能；同时，针对RAGAT和HKGN方法，分别设计了不同的对比实验，充分验证了关键设计的有效性。

本文基于图神经网络来学习知识图谱中的图结构信息，通过实验初步验证了模型的可行性。在未来的研究中还存在以下几方面的内容可以继续拓展：

（1）异质消息聚合机制。在本文中，RAGAT使用了GAT中的单层注意力机制，HKGN直接使用GCN的求和方法聚合。需要注意到，这两种方法的聚合机制对于异质性聚合粒度还比较粗，还可以借鉴目前出现的新的异质图注意力机制来提升模型性能，并且还可以考虑如何将超网络与异质图聚合聚合机制进行融合。

（2）基于图神经网络的多源信息融合。知识图谱除去图结构信息外，还包括文本信息、路径信息乃至于时序信息等丰富的特征信息，事实上，无论是单一的学习哪种信息都无法完全的挖掘知识图谱中的语义，因此如何基于图神经网络融合多种来源的信息是一个非常值得研究的问题。

# 参考文献

[1] Bordes A, Usunier N, García-Durán A, et al. Translating Embeddings for Modeling Multi-relational Data[C]// Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, United States: 2013: 2787–2795.

[2] Wang Q, Mao Z, Wang B, et al. Knowledge Graph Embedding: A Survey of Approaches and Applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2017, 29(12): 2724–2743.

[3] Vashishth S, Sanyal S, Nitin V, et al. Composition-based Multi-Relational Graph Convolutional Networks[C]// 8th International Conference on Learning Representations, Addis Ababa, Ethiopia: OpenReview.net, 2020.

[4] Wang H, Zhang F, Wang J, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems[C]// Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018: 417–426.

[5] Schneider E W. Course Modularization Applied: The Interface System and Its Implications for Sequence Control and Data Analysis[M]. 1973.

[6] Singhal A. Introducing the knowledge graph: things, not strings[EB/OL]. Official google blog. 2012-05. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

[7] Hogan A, Blomqvist E, Cochez M, et al. Knowledge Graphs[J]. arXiv:2003.02320 [cs], 2020.

[8] Krishnan A. Making search easier: How Amazon’s Product Graph is helping customers find products more easily[EB/OL]. 2018. https://blog.aboutamazon.com/innovation/making-search-easier.

[9] Pittman R J, Srivastava A, Hewavitharana S, et al. Cracking the Code on Conversational Commerce[EB/OL]. 2017. https://medium.com/@rjpittman/cracking-the-code-on-conversational-commerce-775b5172f312.

[10] Noy N, Gao Y, Jain A, et al. Industry-scale knowledge graphs: Lessons and challenges[J]. Communications of the ACM, 2019, 62: 36–43.

[11] Wang Z, Zhang J, Feng J, et al. Knowledge Graph Embedding by Translating on Hyperplanes[C]// Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada: AAAI Press, 2014: 1112–1119.

[12] Lin Y, Liu Z, Sun M, et al. Learning Entity and Relation Embeddings for Knowledge Graph Completion[C]// Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, USA: AAAI Press, 2015: 2181–2187.

[13] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C]// Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, Beijing, China: Association for Computational Linguistics, 2015, volume 1: Long papers: 687–696.

[14] Xiao H, Huang M, Hao Y, et al. Transa: An adaptive approach for knowledge graph embedding[J]. arXiv preprint arXiv:1509.05490, 2015.

[15] Zhang Z, Cai J, Zhang Y, et al. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction[C]// The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press, 2020: 3065–3072.

[16] Nickel M, Tresp V, Kriegel H-P. A Three-Way Model for Collective Learning on Multi-Relational Data[C]// Proceedings of the 28th International Conference on Machine Learning, Bellevue, Washington, USA: Omnipress, 2011: 809–816.

[17] Yang B, Yih W, He X, et al. Embedding Entities and Relations for Learning and Inference in Knowledge Bases[C]// 3rd International Conference on Learning Representations, San Diego, CA, USA: 2015.

[18] Nickel M, Rosasco L, Poggio T A. Holographic Embeddings of Knowledge Graphs[C]// Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA: AAAI Press, 2016: 1955–1961.

[19] Trouillon T, Welbl J, Riedel S, et al. Complex Embeddings for Simple Link Prediction[C]// Proceedings of the 33nd International Conference on Machine Learning, New York City, NY, USA: JMLR.org, 2016, 48: 2071–2080.

[20] Sun Z, Deng Z-H, Nie J-Y, et al. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space[C]// 7th International Conference on Learning Representations, New Orleans, LA, USA: OpenReview.net, 2019.

[21] Zhang W, Paudel B, Zhang W, et al. Interaction Embeddings for Prediction and Explanation in Knowledge Graphs[C]// Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia: ACM, 2019: 96–104.

[22] Glorot X, Bordes A, Weston J, et al. A Semantic Matching Energy Function for Learning with Multi-relational Data[C]// 1st International Conference on Learning Representations, Workshop Track Proceedings, Scottsdale, Arizona, USA: 2013, 94: 233–259.

[23] Socher R, Chen D, Manning C D, et al. Reasoning with Neural Tensor Networks for Knowledge Base Completion[C]// Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, United States: 2013: 926–934.

[24] Dong X, Gabrilovich E, Heitz G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C]// Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, New York, USA: Association for Computing Machinery, 2014: 601–610.

[25] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D Knowledge Graph Embeddings[C]// Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA,: AAAI Press, 2018: 1811–1818.

[26] Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, USA: Association for Computational Linguistics, 2018, Volume 2 (Short Papers): 327–333.

[27] Jiang X, Wang Q, Wang B. Adaptive Convolution for Multi-Relational Learning[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota: Association for Computational Linguistics, 2019: 978–987.

[28] Nguyen D Q, Vu T, Nguyen T D, et al. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, Volume 1 (Long and Short Papers): 2180–2189.

[29] Vashishth S, Sanyal S, Nitin V, et al. InteractE: Improving Convolution-Based Knowledge Graph Embeddings by Increasing Feature Interactions[C]// The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NK, USA: AAAI Press, 2020: 3009–3016.

[30] Schlichtkrull M S, Kipf T N, Bloem P, et al. Modeling Relational Data with Graph Convolutional Networks[C]// The Semantic Web - 15th International Conference, Heraklion, Crete, Greece: Springer, 2018, 10843: 593–607.

[31] Shang C, Tang Y, Huang J, et al. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion[C]// The Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA: AAAI Press, 2019, 33: 3060–3067.

[32] Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks[C]// 5th International Conference on Learning Representations, Toulon, France: OpenReview.net, 2017.

[33] Ye R, Li X, Fang Y, et al. A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment[C]// Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China: ijcai.org, 2019: 4135–4141.

[34] Cai L, Yan B, Mai G, et al. TransGCN: Coupling Transformation Assumptions with Graph Convolutional Networks for Link Prediction[C]// Proceedings of the 10th International Conference on Knowledge Capture, New York, NY, USA: ACM, 2019: 131–138.

[35] Guo S, Wang Q, Wang B, et al. Semantically Smooth Knowledge Graph Embedding[C]// Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China: Association for Computational Linguistics, 2015, Volume 1: Long Papers: 84–94.

[36] Lin Y, Liu Z, Luan H, et al. Modeling Relation Paths for Representation Learning of Knowledge Bases[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal: Association for Computational Linguistics, 2015: 705–714.

[37] Xie R, Liu Z, Jia J, et al. Representation Learning of Knowledge Graphs with Entity Descriptions[C]// Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA: AAAI Press, 2016: 2659–2665.

[38] Nickel M, Tresp V, Kriegel H-P. Factorizing YAGO: Scalable Machine Learning for Linked Data[C]// Proceedings of the 21st International Conference on World Wide Web, New York, NY, USA: Association for Computing Machinery, 2012: 271–280.

[39] Wang Q, Wang B, Guo L. Knowledge Base Completion Using Embeddings and Rules[C]// Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, 2015: 1859–1865.

[40] Jiang T, Liu T, Ge T, et al. Encoding Temporal Information for Time-Aware Link Prediction[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas: Association for Computational Linguistics, 2016: 2350–2354.

[41] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada: 2008: 1247–1250.

[42] Miller G A. WordNet: A lexical database for english[J]. Communications of the ACM, 1995, 38(11): 39–41.

[43] Bruna J, Zaremba W, Szlam A, et al. Spectral Networks and Locally Connected Networks on Graphs[C]// 2nd International Conference on Learning Representations, Banff, AB, Canada: 2014.

[44] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[C]// Advances in neural information processing systems, Curran Associates, Inc., 2016, 29.

[45] Gilmer J, Schoenholz S S, Riley P F, et al. Neural Message Passing for Quantum Chemistry[C]// Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia: PMLR, 2017, 70: 1263–1272.

[46] Velickovic P, Cucurull G, Casanova A, et al. Graph Attention Networks[C]// 6th International Conference on Learning Representations, Vancouver, BC, Canada: OpenReview.net, 2018.

[47] Wang X, Ji H, Shi C, et al. Heterogeneous Graph Attention Network[C]// The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, San Francisco, CA, USA: ACM, 2019: 2022–2032.

[48] Zhang C, Song D, Huang C, et al. Heterogeneous graph neural network[C]// Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery &amp; data mining, New York, NY, USA: Association for Computing Machinery, 2019: 793–803.

[49] Hu Z, Dong Y, Wang K, et al. Heterogeneous graph transformer[C]// Proceedings of the web conference 2020, New York, NY, USA: Association for Computing Machinery, 2020: 2704–2710.

[50] Zhao J, Wang X, Shi C, et al. Heterogeneous graph structure learning for graph neural networks[C]// Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, AAAI Press, 2021: 4697–4705.

[51] Nathani D, Chauhan J, Sharma C, et al. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs[C]// Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, 2019, Volume 1: Long Papers: 4710–4723.

[52] Bansal T, Juan D-C, Ravi S, et al. A2N: Attending to Neighbors for Knowledge Graph Inference[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy: Association for Computational Linguistics, 2019: 4387–4392.

[53] Wang P, Han J, Li C, et al. Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding[C]// The Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA: AAAI Press, 2019: 7152–7159.

[54] Li W, Zhang X, Wang Y, et al. Graph2Seq: Fusion Embedding Learning for Knowledge Graph Completion[J]. IEEE Access, 2019, 7: 157960–157971.

[55] Zhang Z, Zhuang F, Zhu H, et al. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion[C]// The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NK, USA: AAAI Press, 2020: 9612–9619.

[56] Xie Z, Zhou G, Liu J, et al. ReInceptionE: Relation-Aware Inception Network with Joint Local-Global Structural Information for Knowledge Graph Embedding[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online: Association for Computational Linguistics, 2020: 5929–5939.

[57] Klein B, Wolf L, Afek Y. A Dynamic Convolutional Layer for short rangeweather prediction[C]// 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 4840–4848.

[58] Jia X, De Brabandere B, Tuytelaars T, et al. Dynamic filter networks[C]// Advances in neural information processing systems, Curran Associates, Inc., 2016, 29.

[59] Riegler G, Schulter S, Rüther M, et al. Conditioned Regression Models for Non-blind Single Image Super-Resolution[C]// 2015 IEEE International Conference on Computer Vision (ICCV), 2015: 522–530.

[60] Ha D, Dai A, Le Q V. HyperNetworks[C]// 5th international conference on learning representations, OpenReview.net, 2017: toulon, france.

[61] Brock A, Lim T, Ritchie J M, et al. SMASH: One-shot model architecture search through HyperNetworks[C]// International conference on learning representations, 2018.

[62] Sitzmann V, Martel J, Bergman A, et al. Implicit Neural Representations with Periodic Activation Functions[C]// Advances in Neural Information Processing Systems, Curran Associates, Inc., 2020, 33: 7462–7473.

[63] Balažević I, Allen C, Hospedales T M. Hypernetwork knowledge graph embeddings[C]// Artificial neural networks and machine learning – ICANN 2019: Workshop and special sessions, Cham: Springer International Publishing, 2019: 553–565.

[64] Nachmani E, Wolf L. Hyper-graph-network decoders for block codes[C]// Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, december 8-14, 2019, vancouver, BC, canada, 2019: 2326–2336.

[65] Nachmani E, Wolf L. Molecule property prediction and classification with graph hypernetworks[J]. CoRR, 2020, abs/2002.00240.

[66] Liu Z, Fang Y, Liu C, et al. Node-wise localization of graph neural networks[C]// Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event / montreal, canada, ijcai.org, 2021: 1520–1526.

[67] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference[C]// Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China: 2015: 57–66.

[68] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C]// Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA: 2017: 5998–6008.

[69] Linmei H, Yang T, Shi C, et al. Heterogeneous graph attention networks for semi-supervised short text classification[C]// Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, 2019: 4821–4830.

[70] Hong H, Guo H, Lin Y, et al. An attention-based graph neural network for heterogeneous structural learning[C]// The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, new york, NY, USA, february 7-12, 2020, AAAI Press, 2020: 4132–4139.

[71] Ragesh R, Sellamanickam S, Iyer A, et al. HeteGCN: Heterogeneous graph convolutional networks for text classification[C]// the fourteenth ACM international conference on web search and data mining, virtual event, israel: ACM, 2021: 860–868.

[72] Wang P, Agarwal K, Ham C, et al. Self-supervised learning of contextual embeddings for link prediction in heterogeneous networks[C]// WWW ’21: The web conference 2021, virtual event / ljubljana, slovenia, april 19-23, 2021, ACM / IW3C2, 2021: 2946–2957.

[73] Huang Z, Li X, Ye Y, et al. MR-GCN: Multi-Relational Graph Convolutional Networks based on Generalized Tensor Product[C]// Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020: 1258–1264.

[74] Fu X, Zhang J, Meng Z, et al. Magnn: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding[C]// Proceedings of the web conference 2020, New York, NY, USA: Association for Computing Machinery, 2020: 2331–2341.

[75] Zhu J-Z, Jia Y-T, Xu J, et al. Modeling the Correlations of Relations for Knowledge Graph Embedding[J]. Journal of Computer Science and Technology, 2018, 33(2): 323–334.

[76] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[C]// Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, vancouver, BC, canada: 2019: 8024–8035.

[77] Ruffinelli D, Broscheit S, Gemulla R. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings[C]// 8th International Conference on Learning Representations, OpenReview.net, 2020.

[78] Sun Z, Vashishth S, Sanyal S, et al. A Re-evaluation of Knowledge Graph Completion Methods[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020: 5516–5522.

[79] Cao Z, Xu Q, Yang Z, et al. Dual quaternion knowledge graph embeddings[C]// Thirty-fifth AAAI conference on artificial intelligence, AAAI Press, 2021: 6894–6902.

[80] Stoica G, Stretcu O, Platanios E A, et al. Contextual Parameter Generation for Knowledge Graph Link Prediction[C]// The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press, 2020: 3000–3008.

[81] Li Z, Liu H, Zhang Z, et al. Learning Knowledge Graph Embedding With Heterogeneous Relation Attention Networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021: 1–13.

[82] Das R, Dhuliawala S, Zaheer M, et al. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning[C]// 6th international conference on learning representations, ICLR 2018, vancouver, BC, canada, april 30 - may 3, 2018, conference track proceedings, OpenReview.net, 2018.

# 攻读硕士期间取得的学术成果

[1] **Liu Xiyang**, Tan Huobin, Chen Qinghong, Lin Guangyan. RAGAT: Relation aware graph attention network for knowledge graph completion[J]. IEEE Access, 2021, 9: 20840–20849.

[2] Wang Ze, Lin Guangyan, Tan Huobin, Chen Qinghong and **Liu Xiyang**. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems[C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 219–228.

# 致谢

当我写下这句话时，我在北航又度过了匆匆的三年。如果说本科四年的经历是一种渐变的累积，让我从高中时的懵懂，逐渐开始转变为主动思考未来的人生道路；那么硕士这三年，就是我拥有了自己的想法，能够更加清晰而明确的追求某种自我认可。

感谢谭老师一直以来对我的指导和帮助，感谢谭老师和林老师提供的具有良好氛围的实验室环境，我非常庆幸自己在当初在大三时选择谭老师作为自己的导师，当我有困惑时，谭老师总会耐心的帮助我解决疑问，提供意见。事实上很多情况下，作为一个学生没有很多的信息来源，也没有经验，面对许多的问题不知道应该选择，在这种情况下能够获得老师的意见，无疑是一种幸运。

我同样感激实验室的其他伙伴，感谢王泽学长在我刚进入实验室时对我的带领和指导；感谢朱伯同同学和王子烈同学搭建的实验环境，特别是有很多实验技巧都是子烈指点的，让我受益良多；感谢陈清泓同学帮我修改小论文的语法。感谢所有实验室中帮助过我的伙伴，没有他们的工作，就没有完成这篇论文的基础。

我还想感激自己的家人，他们是我前进的动力，是我在这世界上唯一也最温暖的家。

另外我想感谢这个和平稳定的国家和社会。我常常思考，尽管作为学生平时会遭遇不如意的事情，撞见各样的难题，但是所有的这些问题是所有人都会有的吗？仅仅只需要考虑如何把事做好，不需要考虑衣食住行的基本需求，本身就是一种极大的幸运。

对于外人来说，看到的自己是一个个的标签，但对于个人来说，只有过程才是我之所以为我的根本。回想起三年发生的各种事情，我没有办法用或慷慨或激昂的语句来概括，最准确的总结可能是一个人努力的做了一点事，仅此而已。