Mixed Geometry Message and Trainable Convolutional Attention Network for Knowledge Graph Completion

Bin Shang[1,2], Yinliang Zhao[1,2*], Jun Liu[1,2], Di Wang[3*]

[1]Shaanxi Provincial Key Laboratory of Big Data Knowledge Engineering,
School of Computer Science and Technology, Xi'an Jiaotong University, China
[2]National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, China
[3]School of Computer Science and Technology, Xidian University, China
binshang0329@163.com, {zhaoy, liukeen}@xjtu.edu.cn, wangdi@xidian.edu.cn

Abstract

Knowledge graph completion (KGC) aims to study the embedding representation to solve the incompleteness of knowledge graphs (KGs). Recently, graph convolutional networks (GCNs) and graph attention networks (GATs) have been widely used in KGC tasks by capturing neighbor information of entities. However, Both GCNs and GATs based KGC models have their limitations, and the best method is to analyze the neighbors of each entity (pre-validating), while this process is prohibitively expensive. Furthermore, the representation quality of the embeddings can affect the aggregation of neighbor information (message passing). To address the above limitations, we propose a novel knowledge graph completion model with mixed geometry message and trainable convolutional attention network named MGTCA. Concretely, the mixed geometry message function generates rich neighbor message by integrating spatially information in the hyperbolic space, hypersphere space and Euclidean space jointly. To complete the autonomous switching of graph neural networks (GNNs) and eliminate the necessity of pre-validating the local structure of KGs, a trainable convolutional attention network is proposed by comprising three types of GNNs in one trainable formulation. Furthermore, a mixed geometry scoring function is proposed, which calculates scores of triples by novel prediction function and similarity function based on different geometric spaces. Extensive experiments on three standard datasets confirm the effectiveness of our innovations, and the performance of MGTCA is significantly improved compared to the state-of-the-art approaches.

Introduction

Knowledge graphs (KGs) represent real-world data as fact triples (head entity, relation, tail entity), which have shown great research value and application prospect. KGs are widely used in many downstream tasks, such as question answering (Kaiser, Saha Roy, and Weikum 2021), dialogue generation (Keizer et al. 2017), semantic search (Xiong, Power, and Callan 2017), and recommender systems (Wang et al. 2021b). Even though the scale of many public KGs is noticeably large such as Yago3 (Mahdisoltani, Biega, and Suchanek 2013) and Freebase (Bollacker et al. 2008), they are still confronted with incompleteness because there are many missing relations among them. Therefore, knowledge graph completion (KGC) has attracted extensive attention and attempts to automatically find out missing facts. Knowledge graph embedding (KGE) is an effective solution for KGC task, and many of them have been proposed such as (Bordes et al. 2013; Yang et al. 2015; Dettmers et al. 2018; Vashishth et al. 2020a; Li et al. 2022; Ge et al. 2023). KGE approaches aim to embed entities and relations into a low-dimensional vector space and define scoring functions to assess the plausibility of triples for link prediction. Although these methods are simple and efficient, they are significantly reliant on the pre-defined scoring function and rather challenging to encode structural information about an entity into a single vector (Dai et al. 2022).

In order to capture the intrinsic graph structure of KGs, graph neural networks (GNNs) (Gilmer et al. 2017; Javaloy et al. 2023) have been used for KGC task. GNNs based KGC models learn the hidden representation of each entity by aggregating its corresponding local neighbors' information (Dai et al. 2022; Wang et al. 2023). Recently, many studies tend to model KGs by diverse types of GNNs such as graph convolutional networks (GCNs) (Kipf and Welling 2017) based models R-GCN (Schlichtkrull et al. 2018), CompGCN (Vashishth et al. 2020b), and LTE-ConvE (Zhang et al. 2022); graph attention networks (GATs) (Veličković et al. 2018) based models MR-GAT (Dai et al. 2022), GreenKGC (Wang et al. 2023) and Ae2KGR (Shang et al. 2023b). Although these approaches have shown promising performance, they still suffer from several evident limitations as follows: (i) **Data dependence**. Both GCNs and GATs based KGC models have their strengths and limitations because they are data sensitive, which results in the problem of data dependence. GCNs based KGC approaches fully summarize neighbor messages to endow the central entity with sufficient structural information, while they tend to stack redundant information when there are various neighbors. GATs based KGC approaches introduces non-uniform score to each neighbors and can reduce the stacking of redundant information, while they tend to focus on certain neighbor entities and weaken the structural information. Therefore, the local structure of each entity can influence the performance of GCNs and GATs. The best method is to analyze the neighbors of each entity before selecting GCNs or GATs (pre-validating), while this

*Corresponding authors.

process is prohibitively expensive. (ii) **Message limitation**. The message function in GNNs can be affected by representation quality of embeddings, which results in the problem of message limitation. The message functions (MFs) are used to generate neighbor information and is a crucial component for the GNNs based KGC methods (Nathani et al. 2019; Dai et al. 2022). Existing MFs are designed only in Euclidean space (zero curvature), which cannot fully capture the intrinsic structural information of KGs and may lead to insufficient neighbor message. Therefore, exploring a new message function can help aggregate local information for GNNs and improve the representation quality of entity embeddings.

To address the above issues, in this paper, we propose a **M**ixed **G**eometry message and **T**rainable **C**onvolutional **A**ttention network based knowledge graph completion model named MGTCA. In order to deal with the problem of **message limitation**, MGTCA introduces a mixed geometry message function (MGMF), which captures spatially information in the hyperbolic space (negative curvature), hypersphere space (positive curvature) and Euclidean space (zero curvature) jointly. In addition, MGMF integrates these information into message through geometric mapping and linear transformation. In order to deal with the problem of **data dependence**, MGTCA presents a trainable convolutional attention network (TCAN), which comprises different types of GNNs in one trainable formulation. TCAN aims to eliminate the necessity of pre-validating the local structure of KGs, complete the autonomous switching of GNNs types, and learn the amount of attention required for each local structure. Furthermore, to calculate scores of triples, we propose a mixed geometry scoring function with novel prediction function and similarity function based on the three geometric spaces. Our contributions are summarized as follows:

- We propose a mixed geometry message function to generate rich neighbor message by integrating spatially information in the hyperbolic space, hypersphere space and Euclidean space jointly. To the best of our knowledge, we are the first to explore to generate mixed geometric message in GNNs based KGC methods.
- We propose a trainable convolutional attention network to complete the autonomous switching of GNNs types and learn the amount of attention required for each local structure by comprising different types of GNNs in one trainable formulation. To the best of our knowledge, we are the first to explore the autonomous switching of GNNs types in KGC task.
- We propose a mixed geometry scoring function to calculate scores of triples by novel prediction function and similarity function based on three geometric spaces.
- We conduct extensive experiments on three benchmark datasets. The results show that MGTCA achieves state-of-the-art performance compared to existing models.

Related Work

Non-Euclidean KGC Models

Modeling KGs in non-Euclidean spaces has attracted considerable attention, which can capture the complex structures of KGs by specific geometric space and improve the representation quality of embeddings. ManifoldE (Xiao, Huang, and Zhu 2016a) expands pointwise modeling in the translation based principle to manifoldwise space (e.g., hypersphere space). MuRP (Balazevic, Allen, and Hospedales 2019) learns KG embeddings in hyperbolic space to capture the hierarchical structure in the KG. RotH (Chami et al. 2020) introduces the hyperbolic geometry on the basis of the rotation. Meng's (Meng et al. 2019) proposes a spherical generative model and learns word and paragraph embeddings jointly. These works model the KG in only one geometric space, which can not capture the complex spatial structure of KGs. Recently, in order to make full use of the advantages of each geometric space, M^2 GNN (Wang et al. 2021a) constructs a generic graph neural network framework to model multi-relational KG. HBE (Pan and Wang 2021) fine-tunes the operator and fix model in polar coordinate system to embed KGs. GIE (Cao et al. 2022) is proposed to embrace semantic matching between entities and satisfy the key of relational representation learning.

Euclidean KGC Models

Euclidean KGC Models capture the information of KGs and prediction missing facts in Euclidean space. Generally, existing Euclidean KGC models can be divided into four groups: (i) *Translation-based models* consider the relations as translation between head and tail entities and design scoring function based on distances, such as TransE (Bordes et al. 2013), TransH (Wang et al. 2014), TransR (Lin et al. 2015), TransG (Xiao, Huang, and Zhu 2016b), RotatE (Sun et al. 2019), RotatE-IAS (Yang et al. 2022), HousE (Li et al. 2022), and CompoundE (Ge et al. 2023). (ii) *Semantic matching models* design scoring function by similarity matching of vector or matrix, such as RESCAL (Nickel, Tresp, and Kriegel 2011), DistMult (Yang et al. 2015), ComplEx (Trouillon et al. 2016), TuckER (Balažević, Allen, and Hospedales 2019), and HAKE (Zhang et al. 2020). (iii) *Convolutional neural networks (CNNs) based Models* employ multi-layer CNNs to generate more expressive embeddings, such as ConvKB (Nguyen et al. 2018), ConvE (Dettmers et al. 2018), and InteractE (Vashishth et al. 2020a). (iv) *Graph neural networks (GNNs) based models* utilize GNNs to update the embeddings of entities and relations based on the structural information of the knowledge graph, such as R-GCN (Schlichtkrull et al. 2018), KBGAT (Nathani et al. 2019), CompGCN (Vashishth et al. 2020b), ATTH (Chami et al. 2020), HittER (Chen et al. 2021), Rot-Pro (Song, Luo, and Huang 2021), SE-GNN (Li et al. 2022), LTE-ConvE (Zhang et al. 2022), MRGAT (Dai et al. 2022), HADC (Shang et al. 2023a), ConKGC (Shang et al. 2023c), and GreenKGC (Wang et al. 2023).

Although the aforementioned GNNs based models have achieved satisfactory performance, they use a single type of GNNs to learn embeddings, which will degrade the representation quality of embeddings because both GCNs and GATs have their limitations when aggregating neighbor information. Furthermore, the message function of them are in Euclidean space, which cannot fully capture the intrinsic structural information of KGs.

Preliminaries

Geometric Space

Geometric spaces are distinguished according to the value of the curvature. Specifically, the curvature c is negative for hyperbolic space $\mathbb{H}$, positive for hypersphere space $\mathbb{S}$, and zero for Euclidean space $\mathbb{E}$. The Poincaré ball is a popular model for describing the geometric space in mathematical language (Nickel and Kiela 2017; Chami et al. 2020; Xiao et al. 2022), which has basic mathematical operations (e.g., addition, multiplication) and provides closed-form expressions for many basic objects such as distance and angle (Ganea, Bécigneul, and Hofmann 2018). The principled generalizations of basic operations in hypersphere space are similar to operations in hyperbolic space, except that the curvature $c > 0$. Therefore, here we only introduce the operation of hyperbolic space, the operation of hypersphere space can be obtained by analogy.

The hyperbolic space can be formalized as an approximated vectorial structure by the framework of gyrovector space (Ungar 2008). For two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}_c^d$ in the hyperbolic space, the Möbius addition (Ganea, Bécigneul, and Hofmann 2018) is used as the vector addition in $\mathbb{H}_c^d$:

$$\boldsymbol{x} \oplus_c \boldsymbol{y} = \frac{(1 + 2c\boldsymbol{x}\boldsymbol{y} + c||\boldsymbol{y}||^2)\boldsymbol{x} + (1 - c||\boldsymbol{x}||^2)\boldsymbol{y}}{1 + 2c\boldsymbol{x}\boldsymbol{y} + c^2||\boldsymbol{x}||^2||\boldsymbol{y}||^2}, \quad (1)$$

then the distance between these two points is measured along a geodesic (shortest path between them) as follows:

$$d_c(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{\sqrt{c}} \text{arctanh}(\sqrt{c}|| - \boldsymbol{x} \oplus_c \boldsymbol{y}||). \quad (2)$$

The practical computations (addition, multiplication, etc) in the hyperbolic space are often implemented using the tangent space. For $\boldsymbol{x} \in \mathbb{H}_c^d$, the associated tangent space $\mathcal{T}_x\mathbb{H}_c^d$ is a d-dimensional Euclidean space. Exponential map and logarithmic map can achieve the mutual transformation between the local hyperbolic space and the tangent space of the point. The logarithmic map $\log_{\boldsymbol{x}}^c$ transforms the point to the tangent space, and the exponential map $\exp_{\boldsymbol{x}}^c$ transforms back to the hyperbolic space. Specifically, these two maps have more appealing forms when $\boldsymbol{x} = \boldsymbol{0}$, namely for: $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{0}}\mathbb{H}_c^d \setminus \{\boldsymbol{0}\}, \boldsymbol{y} \in \mathbb{H}_c^d \setminus \{\boldsymbol{0}\}$:

$$\exp_{\boldsymbol{0}}^c(\boldsymbol{v}) = \tanh(\sqrt{c}||\boldsymbol{v}||)\frac{\boldsymbol{v}}{\sqrt{c}||\boldsymbol{v}||}, \quad (3)$$

$$\log_{\boldsymbol{0}}^c(\boldsymbol{y}) = \text{artanh}(\sqrt{c}||\boldsymbol{y}||)\frac{\boldsymbol{y}}{\sqrt{c}||\boldsymbol{y}||}. \quad (4)$$

Furthermore, the multiplication in hyperbolic space can be defined by Möbius scalar multiplication between vectors $\boldsymbol{r} \in \mathbb{E}$ and $\boldsymbol{x} \in \mathbb{H}_c^d$:

$$\boldsymbol{r} \otimes_c \boldsymbol{x} = \frac{1}{\sqrt{c}} \tanh\left(\boldsymbol{r}\tanh^{-1}(\sqrt{c}||\boldsymbol{x}||)\right) \frac{\boldsymbol{x}}{||\boldsymbol{x}||}, \quad (5)$$

Knowledge Graph Completion

A knowledge graph ($\mathcal{G}$) can be formulated as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the set of entities (nodes) and relations (edges), respectively. $\mathcal{T} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$ is triple set in $\mathcal{G}$, and $r \in \mathcal{R}$ is the relation between entity h and t. KGC approaches first project entities $h \in \mathcal{E}$ onto entity embedding matrix $\{\boldsymbol{h} \in \boldsymbol{E} \mid \boldsymbol{E} \in \mathbb{R}^{|\mathcal{E}| \times d}\}$ and relations $r \in \mathcal{R}$ onto relation embedding matrix $\{\boldsymbol{r} \in \boldsymbol{R} \mid \boldsymbol{R} \in \mathbb{R}^{|\mathcal{R}| \times d}\}$, where $|\mathcal{E}|$ and $|\mathcal{R}|$ represent the total number of entities and relations respectively, d is the embedding dimension. The link prediction task aims to predict the tail entity t for a query $(h, r, ?)$ or head entity for a query $(?, r, t)$. Such a goal is achieved by designing and learning a scoring function $\Phi(h, r, t) = \xi(\varphi(\boldsymbol{h}, \boldsymbol{r}), \boldsymbol{t})$. $\varphi(\boldsymbol{h}, \boldsymbol{r})$ is prediction function, which predicts the tail entity embedding $\boldsymbol{t}'$. $\xi(\boldsymbol{t}', \boldsymbol{t})$ is similarity function, which measures the similarity between the predicted tail entity embedding $\boldsymbol{t}'$ and the true tail entity embedding $\boldsymbol{t}$. The scoring function also directly affects the model performance. Furthermore, The goal of the optimization is to score a correct triple higher than incorrect triples.

Methodology

In this section, we show the formal description and implementation details of our proposed model MGTCA. We start by introducing the mixed geometry message function. Then we describe the trainable convolutional attention network. Next, we present the mixed geometry scoring function. In the end, we provide loss function. The overall framework of MGTCA is shown in Figure 1.

Generally, the whole model contains L layers, the input to l-th layer (l = 1,...,L) are two embedding sets: (1) the output entity embedding matrix $\boldsymbol{E}^{l-1} = \left\{\boldsymbol{e}_1^{l-1}, \boldsymbol{e}_2^{l-1}, ..., \boldsymbol{e}_{|\mathcal{E}|}^{l-1}\right\} \in \mathbb{R}^{|\mathcal{E}| \times d}$ from (l-1)-th layer, where $|\mathcal{E}|$ is the number of entities, and d is the dimension of embeddings. (2) the output relation embedding matrix $\boldsymbol{R}^{l-1} = \left\{\boldsymbol{r}_1^{l-1}, \boldsymbol{r}_2^{l-1}, ..., \boldsymbol{r}_{|\mathcal{R}|}^{l-1}\right\} \in \mathbb{R}^{|\mathcal{R}| \times d}$ from (l-1)-th layer, where $|\mathcal{R}|$ is the number of relations. The l-th layer then produces the corresponding new output embedding matrices (of potentially different cardinality), $\boldsymbol{E}^l \in \mathbb{R}^{|\mathcal{E}| \times d}$ and $\boldsymbol{R}^l \in \mathbb{R}^{|\mathcal{R}| \times d}$. Specifically, we describe the l-th layer of our model.

Mixed Geometry Message Function

Graph neural networks (GNNs) (Gilmer et al. 2017) have been widely used in knowledge graph completion tasks, which can update the embeddings of entities by aggregating the information from their neighbors based on message function (MF). In this way, the entity embeddings can obtain structural information. The message function is used to generate neighbor information and is a crucial component for the GNNs based KGC methods (Nathani et al. 2019; Dai et al. 2022). Recently, many MF for KGC task have been proposed and achieved satisfactory results in the Euclidean space (zero curvature). However, KGs usually contain rich structural information and they cannot be captured in the Euclidean space, which leads to insufficient neighbor information passed by MF. To alleviate this problem, we propose a mixed geometry message function to integrate spatially information in diverse geometric spaces (hyperbolic, hypersphere and Euclidean spaces).

Specifically, given a central entity h_i and its neighbor set

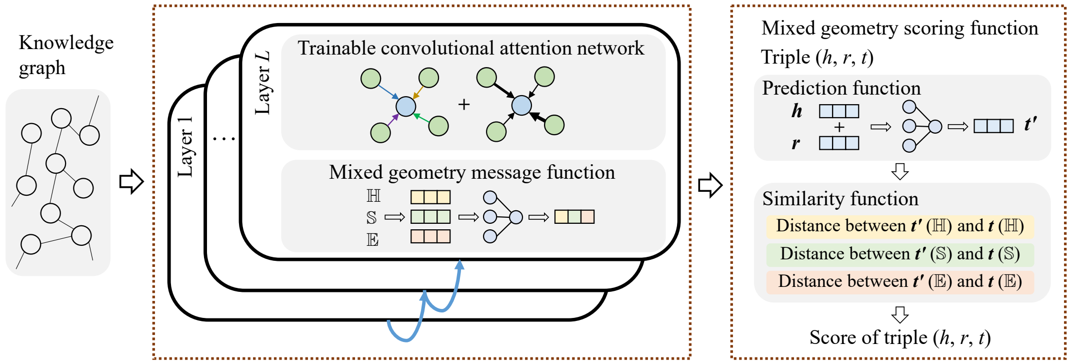

Figure 1: The overall framework of MGTCA. The embeddings are learned by multi layer trainable convolutional attention network with mixed geometry message function, and then are fed into mixed geometry scoring function for link prediction. $\mathbb{H}$, $\mathbb{S}$, and $\mathbb{E}$ represent hypersphere space, hypersphere space, and Euclidean space respectively.

$\mathcal{N}_i = \{(r_j, t_j) \mid (h_i, r_j, t_j) \in \mathcal{T}\}$, which denotes all neighbors of entity h_i. The messages in the three geometric spaces can be defined as follows:

$$\begin{aligned} \boldsymbol{E}_j^l &= \boldsymbol{r}_j^{l-1} \boldsymbol{t}_j^{l-1}, \\ \boldsymbol{H}_j^l &= \boldsymbol{r}_j^{l-1} \otimes_{c_1^l} \exp_{\mathbf{0}}^{c_1^l}(\boldsymbol{t}_j^{l-1}), \\ \boldsymbol{S}_j^l &= \boldsymbol{r}_j^{l-1} \otimes_{c_2^l} \exp_{\mathbf{0}}^{c_2^l}(\boldsymbol{t}_j^{l-1}), \end{aligned} \tag{6}$$

where $\boldsymbol{r}_j^{l-1}$ and $\boldsymbol{t}_j^{l-1}$ denote the embeddings of relation r_j and tail entity t_j in Euclidean space in (l-1)-th layer respectively, $c_1^l < 0$ and $c_2^l > 0$ are two trainable curvatures for hyperbolic and hypersphere spaces in l-th layer respectively, the operation $\otimes$ is related to Eq. (5) and $\exp_{\mathbf{0}}^c$ is the exponential map (Eq. (3)). Messages $\boldsymbol{E}_j^l$, $\boldsymbol{H}_j^l$, and $\boldsymbol{S}_j^l$ in l-th layer are from different geometric spaces, we combine them and define our mixed geometry message function as follows:

$$\phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}) = \boldsymbol{W}_m^l \left[\boldsymbol{E}_j^l \parallel \log_{\mathbf{0}}^{c_1^l}(\boldsymbol{H}_j^l) \parallel \log_{\mathbf{0}}^{c_2^l}(\boldsymbol{S}_j^l) \right], \tag{7}$$

where $\boldsymbol{W}_m^l \in \mathbb{R}^{d \times 3d}$ is a trainable transformation matrix, $\parallel$ denotes the concatenation of embeddings, $\log_{\mathbf{0}}^c$ is logarithmic map (Eq. (4)). It should be noted that the input and output of $\phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1})$ are in Euclidean space, but the output message contain rich spatially information from the three geometric spaces. In this way, we can improve embedding representation quality without burdening vector storage.

Trainable Convolutional Attention Network

Recently, many GNNs based KGC models have been proposed. Graph convolutional networks (GCNs) (Kipf and Welling 2017) and graph attention networks (GATs) (Veličković et al. 2018) are two important and widely used GNNs. For a given central entity h_i, the message passing function of GCNs for KGC task can be defined as follows:

$$\bar{\boldsymbol{h}}_i^l = \sigma(\widehat{\boldsymbol{h}}_i^l) \quad \text{where} \quad \widehat{\boldsymbol{h}}_i^l = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}), \tag{8}$$

where $\bar{\boldsymbol{h}}_i^l$ represents the generated embedding of entity h_i in l-th layer, $\sigma(\cdot)$ is an activation function, $\mathcal{N}_i$ denotes the neighbors of h_i, $\phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1})$ is our proposed mixed geometry message function (Eq. (7)). And the message passing function of GATs for KGC task is defined as follows:

$$\begin{aligned} &\bar{\boldsymbol{h}}_i^l = \sigma(\widehat{\boldsymbol{h}}_i^l) \quad \text{where} \quad \widehat{\boldsymbol{h}}_i^l = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^l \phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}), \\ &\alpha_{ij}^l = \frac{\exp(\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}))}{\sum_{k \in \mathcal{N}_i} \exp(\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_k^{l-1}, \boldsymbol{t}_k^{l-1}))}, \\ &\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}) = \\ &\text{LeakyRelu}(\boldsymbol{a}^{l\top}[\boldsymbol{W}_q^l \boldsymbol{h}_i^{l-1} \parallel \boldsymbol{W}_k^l \phi^l(\boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1})]), \end{aligned} \tag{9}$$

where $\boldsymbol{W}_q^l \in \mathbb{R}^{d_h \times d}$, and $\boldsymbol{W}_k^l \in \mathbb{R}^{d_h \times d}$ are trainable transformation matrices in l-th layer, d_h is the dimension of hidden embedding, $\boldsymbol{a}^l$ is the attention vector in l-th layer, $\parallel$ denotes the concatenation of two embeddings, and $\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1})$ is the attention function in l-th layer.

GCNs based KGC models treat the neighbors of entities equally and can fully summarize neighbor messages to endow the central entity with sufficient structural information, while they may stack redundant information when there are various neighbors. GATs based KGC models introduces non-uniform score to each neighbors and can reduce the stacking of redundant information, while they tend to focus on certain neighbor entities and weaken the structural information. Based on the above observations, we can conclude that both GCNs and GATs based KGC approaches are sensitive to KG structures (properties of entity neighbors). The best method to deal this problem is to analyze the neighbors of each entity (pre-validating), while this process is prohibitively expensive. Therefore, we propose a knowledge graph convolutional attention network (KGCAT), which applies the convolutional operation to the attention function based on Eq. (9):

$$\begin{aligned}&\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}) = \\ &\text{LeakyRelu}(\boldsymbol{a}^{l\top}[\boldsymbol{W}_q^l \widetilde{\boldsymbol{h}}_i^l \parallel \boldsymbol{W}_k^l \phi^l(\boldsymbol{r}_j^{l-1}, \widetilde{\boldsymbol{t}}_j^l)]), \\ &\widetilde{\boldsymbol{h}}_i^l = \frac{1}{1+|\mathcal{N}_i|}\left(\boldsymbol{h}_i^{l-1} + \sum_{k\in\mathcal{N}_i} \phi^l(\boldsymbol{r}_k^{l-1}, \boldsymbol{t}_k^{l-1})\right),\end{aligned} \tag{10}$$

where $\widetilde{\boldsymbol{h}}_i^l$ and $\widetilde{\boldsymbol{t}}_j^l$ are the convolved embeddings of the central entity h_i and neighbor entity t_j in l-th layer respectively. Using the convolution operation before the attention mechanism can give the central entity sufficient structural information while avoiding redundant information stacking. But this method is a compromise, which may cover up the original advantages of GCNs and GATs. To this end, we propose a trainable convolutional attention network (TCAN), and its attention function is defined as follows:

$$\begin{aligned}&\psi^l(\boldsymbol{h}_i^{l-1}, \boldsymbol{r}_j^{l-1}, \boldsymbol{t}_j^{l-1}) = \\ &\alpha^l\text{LeakyRelu}(\boldsymbol{a}^{l\top}[\boldsymbol{W}_q^l \widetilde{\boldsymbol{h}}_i^l \parallel \boldsymbol{W}_k^l \phi^l(\boldsymbol{r}_j^{l-1}, \widetilde{\boldsymbol{t}}_j^l)]), \\ &\widetilde{\boldsymbol{h}}_i^l = \frac{\boldsymbol{h}_i^{l-1} + \beta^l \sum_{k\in\mathcal{N}_i} \phi^l(\boldsymbol{r}_k^{l-1}, \boldsymbol{t}_k^{l-1})}{1+\beta^l|\mathcal{N}_i|},\end{aligned} \tag{11}$$

where $\alpha^l, \beta^l \in [0, 1]$ are two trainable coefficients in l-th layer. According to these two coefficients, TCAN can be transformed into GCNs ($\alpha^l = 0$), GATs ($\alpha^l = 1$ and $\beta^l = 0$), and KGCAT ($\alpha^l = 1$ and $\beta^l = 1$). TCAN eliminates the necessity of pre-validating the local structure of KGs by comprising three types of GNNs in one trainable formulation. Furthermore, it completes the autonomous switching of GNNs types and can learn the amount of attention required for each local structure.

Our model contains L layers, each of them has a trainable convolutional attention network and mixed geometry message function. The output of the final layer is the generated embeddings of entities with rich structural information, which are fed into scoring function for link prediction.

Mixed Geometry Scoring Function

For a triple (h, r, t), the scoring function $\Phi(h, r, t) = \text{softmax}(\xi(\varphi(\boldsymbol{h}, \boldsymbol{r}), \boldsymbol{t}))$ is composed of prediction function $\boldsymbol{t}' = \varphi(\boldsymbol{h}, \boldsymbol{r})$ and similarity function $\xi(\boldsymbol{t}', \boldsymbol{t})$. In this work, we propose a novel prediction function as follows:

$$\boldsymbol{t}' = \varphi(\boldsymbol{h}, \boldsymbol{r}) = \boldsymbol{W}_p [\boldsymbol{h} \parallel \boldsymbol{r}] + \boldsymbol{b}_p, \tag{12}$$

where $\boldsymbol{W}_p \in \mathbb{R}^{d\times 2d}$ is a trainable transformation matrix, $\mathbf{b}_p$ is bias, $\boldsymbol{t}'$ is the predicted embedding of tail entity t. Considering that the embeddings contain the information from hyperbolic, hypersphere and Euclidean spaces, we propose a novel geometric similarity function as follows:

$$\begin{aligned}\xi(\boldsymbol{t}', \boldsymbol{t}) = d_0(\boldsymbol{t}', \boldsymbol{t}) + d_{c_1}(\exp_{\mathbf{0}}^{c_1}(\boldsymbol{t}'), \exp_{\mathbf{0}}^{c_1}(\boldsymbol{t})) + \\ d_{c_2}(\exp_{\mathbf{0}}^{c_2}(\boldsymbol{t}'), \exp_{\mathbf{0}}^{c_2}(\boldsymbol{t})),\end{aligned} \tag{13}$$

where $d_c(\cdot)$ is the distance function (Eq. (2)), $c_1 < 0$ and $c_2 > 0$ are two trainable curvatures for hyperbolic and hypersphere spaces. Finally, the softmax function is employed on the absolute score calculated by the similarity function to get the relative score of each triple.

Datasets	Entities	Relations	Train triples	Validation triples	Test triples
FB15k-237	14,541	237	272,115	17,535	20,466
YAGO3-10	123,182	37	1,079,040	5,000	5,000
WN18RR	40,943	11	86,835	3,034	3,134

Table 1: Statistics of the datasets used in this paper.

Training and Optimization

Our objective is to minimize the Bernoulli negative log-likelihood, based on which the loss function of MGTCA is defined as follows:

$$\begin{aligned}\mathcal{L} = \sum_{(h,r,t)\in\mathcal{T}} -\frac{1}{N}\sum_{i=1}^{N}(y(h, r, t_i)\log(p_i) \\ + (1 - y(h, r, t_i))\log(1 - p_i)),\end{aligned} \tag{14}$$

where $y(h, r, t_i)$ is the label (1 or 0) of the triple (h, r, t_i), $p_i = \Phi(h, r, t_i)$ is the score calculated by scoring function, N denotes the number of candidates for the tail entity. We use Adam (Kingma and Ba 2015) as optimizer, and label smoothing (Szegedy et al. 2016), Dropout (Srivastava et al. 2014), Batch normalization (Ioffe and Szegedy 2015) to lessen overfitting.

Experiments

Experimental Setup

Datasets We evaluate our proposed model by three standard datasets: FB15k-237 (Toutanova et al. 2015), YAGO3-10 (Dettmers et al. 2018), and WN18RR (Dettmers et al. 2018). FB15k-237 is a subset of FB15k (Bordes et al. 2013), in which the inverse relations are removed. YAGO3-10 is a subset of YAGO3 (Mahdisoltani, Biega, and Suchanek 2013), which constitutes entities with at least 10 relations. WN18RR is a subset of WN18 (Bordes et al. 2013) and the main relation patterns are symmetry/antisymmetry and composition. The details of them are summarized in Table 1.

Evaluation Metrics Following previous work (Dettmers et al. 2018), our model is evaluated with link prediction task: ranking all entities to predict the tail entity in query $(h, r, ?)$ or the head entity in query $(?, r, t)$. We adopt four evaluation metrics: the average inverse rank of the test triples mean reciprocal rank (MRR), and the proportion of correct entities ranked in top k Hits@k ($k \in \{1, 3, 10\}$). We follow the standard evaluation protocol in the filtered setting: all true triples in the KG are filtered out during evaluation.

Baselines We compare results with the following SOTA models: Euclidean approaches TransE (Bordes et al. 2013), ConvE (Dettmers et al. 2018), RotatE (Sun et al. 2019), CompGCN (Vashishth et al. 2020b), HittER (Chen et al. 2021), LTE-ConvE (Zhang et al. 2022), RotatE-IAS (Yang et al. 2022), MRGAT (Dai et al. 2022), GreenKGC (Wang et al. 2023), and CompoundE (Ge et al. 2023). Non-Euclidean approaches MuRP (Balazevic, Allen, and Hospedales 2019), MuRS (Wang et al. 2021a), MuRMP (Wang et al. 2021a), HBE (Pan and Wang 2021), Rot-Pro (Song, Luo, and Huang 2021), and GIE (Cao et al. 2022).

Model	FB15k-237				YAGO3-10				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Euclidean approaches												
TransE	.294	-	-	.465	-	-	-	-	.226	-	-	.501
ConvE	.325	.237	.356	.501	.440	.350	.490	.620	.430	.400	.440	.520
RotatE	.338	.241	.375	.533	.495	.402	.550	.670	.476	.428	.492	.571
CompGCN	.355	.264	.390	.535	.489	.395	.500	.582	.479	.443	.494	.546
HittER	<u>.373</u>	<u>.279</u>	<u>.409</u>	<u>.558</u>	-	-	-	-	<u>.503</u>	.462	<u>.516</u>	<u>.584</u>
LTE-ConvE	.355	.264	.389	.535	-	-	-	-	.472	.437	.485	.544
RotatE-IAS	.339	.242	.374	.532	-	-	-	-	.483	<u>.467</u>	.502	.570
MRGAT	.358	.266	.386	.542	.552	.439	.561	.698	.481	.443	.501	.568
GreenKGC	.345	.265	.369	.507	.453	.361	.509	.629	.411	.367	.430	.491
CompoundE	.357	.264	.393	.545	-	-	-	-	.491	.450	.508	.576
Non-Euclidean approaches												
MuRP	.335	.243	.367	.518	.354	.249	.400	.567	.481	.440	.495	.566
MuRS	.338	.249	.373	.525	.351	.244	.382	.562	.454	.432	.482	.550
MuRMP	.345	.258	.385	.542	.358	.248	.389	.566	.473	.435	.485	.552
HBE	.336	.239	.372	.534	-	-	-	-	.488	.448	.502	.570
Rot-Pro	.344	.246	.383	.540	.542	.443	.596	.699	.457	.397	.482	.577
GIE	.362	.271	.401	.552	<u>.579</u>	<u>.505</u>	<u>.618</u>	<u>.709</u>	.491	.452	.505	.575
MGTCA	**.393**	**.291**	**.428**	**.583**	**.586**	**.514**	**.629**	**.721**	**.511**	**.475**	**.525**	**.593**

Table 2: Link prediction results of MRR and Hits@k on FB15k-237, YAGO3-10, and WN18RR datasets. The best score is in bold and second best score is underlined.

Implementation Details We set layer number $L = 5$, attention head number is 3, and dimension $d = 200$. Coefficients α^l and β^l $(l = 1, ..., L)$ are initially set to 0.5. For each dataset, the best performing hyper-parameters are found by grid search on the validation set. All experiments are performed on single NVIDIA GeForce RTX2080Ti GPU, and are implemented by the PyTorch framework.

Results on Link Prediction

Main Results Table 2 presents the link prediction results on FB15k-237, YAGO3-10, and WN18RR datasets. We strictly follow the experimental setting and data splitting of the previous work (Dettmers et al. 2018) and report the results in the original papers for some baselines. It is clear that our proposed model MGTCA achieves the best performance on the vast majority of datasets by comparing with existing state-of-the-art (SOTA) models. MGTCA improves the four evaluation metrics by 2%-3% compared to the SOTA results (underlined results) on the FB15k-237 dataset. Particularly, MRR and Hits@10 are improved from 0.373 to 0.393 and 0.558 to 0.583, respectively. On YAGO3-10 and WN18RR datasets, MGTCA yields a significant improvement for Hits@3 and Hits@10 compared with SOTA baselines. Furthermore, compared with GNNs based KGC models such as MRGAT and GreenKGC, MGTCA achieves definitive improvement on all datasets, which demonstrate that our proposed trainable convolutional attention network facilitates the exploration of local structures as well as the learning of embeddings. Finally, MGTCA outperforms existing non-Euclidean approaches, the reason is that MGTCA designs its unique mixed geometry message function and scoring function. These two functions integrate the spatially information from three geometric spaces for message passing and link prediction respectively.

	1-1		1-N		N-1		N-N	
	MRR	H10	MRR	H10	MRR	H10	MRR	H10
TransE	.217	.407	.183	.399	.254	.381	.323	.510
ConvE	.195	.401	.212	.410	.271	.397	.352	.531
MRGAT	.178	.395	.237	.413	.294	.432	.371	.562
MGTCA	**.219**	**.411**	**.246**	**.421**	**.312**	**.447**	**.382**	**.572**

Table 3: Link prediction results of MRR and Hits@10 from different relation types on FB15k-237 dataset.

Analysis of Relations Generally, the number of relations is much smaller than the number of entities in KGs, so the same relation is usually connected to multiple entities, which leads to multiple relation types. Following (Bordes et al. 2013), relations can be classified into four categories: one-to-one (1-1), one-to-many (1-N), many-to-one (N-1), and many-to-many (N-N). In order to verify whether MGTCA can effectively deal with the challenge brought by different relation types, we classify the relations in FB15k-237 into the four types mentioned above. The link prediction results of them is shown in Table 3. It can be found that MGTCA is more advantageous for modeling KGs with various relation types. Specifically, MGTCA can better model complex relations such as 1-N, N-1, and N-N types, the reason is that the proposed mixed geometry message function is able to capture the interactions between entities and relations.

	FB15k-237				YAGO3-10				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
w/o MGMF	.377	.275	.411	.567	.568	.497	.610	.704	.496	.459	.509	.578
w/o $\mathbb{H}$	.380	.278	.413	.570	.570	.501	.613	.706	.499	.463	.511	.580
w/o $\mathbb{S}$	.381	.280	.414	.571	.571	.502	.612	.708	.501	.466	.513	.582
w/o $\mathbb{E}$	.380	.279	.412	.572	.571	.501	.613	.706	.501	.466	.513	.581
w/o MGSF	.380	.280	.414	.569	.572	.499	.611	.706	.499	.465	.510	.581
MGTCA-GCN	.381	.280	.415	.571	.574	.500	.612	.709	.501	.467	.512	.582
MGTCA-GAT	.384	.284	.417	.573	.577	.503	.616	.712	.503	.470	.516	.585
MGTCA-KGCAT	.387	.287	.421	.576	.580	.507	.620	.715	.506	.474	.520	.588
MGTCA	**.393**	**.291**	**.428**	**.583**	**.586**	**.514**	**.629**	**.721**	**.511**	**.475**	**.525**	**.593**

Table 4: Ablation study results on three datasets. w/o MGMF represents removing mixed geometry message function (MGMF) from MGTCA. w/o $\mathbb{H}$, w/o $\mathbb{S}$ and w/o $\mathbb{E}$ denote removing hyperbolic, hypersphere and Euclidean space respectively. w/o MGSF denotes removing mixed geometry scoring function (MGSF). MGTCA-M denotes that the proposed trainable convolutional attention network (TCAN) is replaced by M.

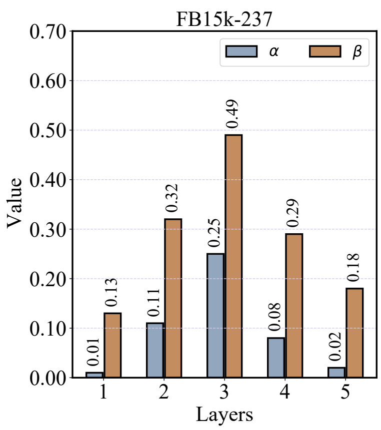

Figure 2: Values of α and β over layers.

Effect of α and β

The two coefficients α and β are important for our proposed trainable convolutional attention network (TCAN). TCAN can be transformed into GCNs ($\alpha = 0$), GATs ($\alpha = 1$ and $\beta = 0$), and KGCAT ($\alpha = 1$ and $\beta = 1$). The values of these two coefficients reflect the characteristics of each layer. Therefore, we observe their values in each layer on FB15k-237 dataset, and the experimental results are shown in Figure 2. It can be found that the value of α of the first layer and the last layer are close to 0, so both layers can be regarded as GCNs. The attention of the third layer plays the largest role, and this layer has been approximately transformed into KGCAT. The second and fourth layers are close to GATs. Overall, the results fully verify the advantage of MGTCA, that is, each layer of it can autonomously learn α and β to adjust its GNNs type.

Ablation Study

Table 4 shows the ablation study results of our proposed MGTCA on the three datasets, where we evaluate the innovations of our model to judge their contribution. The comparison results indicate that our proposed mixed geometry message function (MGMF), trainable convolutional attention network (TCAN) and mixed geometry scoring function (MGSF) are all valid, that is, removing any of them will make the model less effective. Specifically, we use the message function from MRGAT (Dai et al. 2022) for removing MGMF. The hyperbolic, hypersphere and Euclidean space can be removed directly from our model, and the scoring function is defined according to the rest spaces. For removing MGSF, we use the scoring function of ConvE (Dettmers et al. 2018) for link prediction. MGMF integrates the spatially information to generate rich neighbor message, TCAN comprises different types of GNNs in one trainable formulation, and MGSF designs novel prediction function and similarity function based on the three geometric spaces. These three innovations are important components of our model, and the ablation results have verify their contribution. Furthermore, removing any of geometric spaces in MGTCA leads to a decline in model performance, which demonstrate that they all contribute significantly to the message passing and link prediction.

Conclusion

In this paper, we propose a mixed geometry message and trainable convolutional attention network for knowledge graph completion named MGTCA. MGTCA introduces a mixed geometry message function to enrich the neighbor message by integrating the spatially information in the hyperbolic space (negative curvature), hypersphere space (positive curvature) and Euclidean space (zero curvature) jointly. To eliminate the necessity of pre-validating the local structure of KGs, complete the autonomous switching of GNNs types, and learn the amount of attention required for each local structure, MGTCA presents a trainable convolutional attention network (TCAN) by comprising different types of GNNs in one trainable formulation. Moreover, MGTCA designs a mixed geometry scoring function to calculate scores of triples by novel prediction function and similarity function based on the three geometric spaces. Empirical experimental evaluations on three well-established datasets show that MGTCA can achieve the state-of-the-art performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62137002, 62192781, and 62072354), the Fundamental Research Funds for the Central Universities (QTZX23084).

References

Balazevic, I.; Allen, C.; and Hospedales, T. 2019. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*, 32.

Balažević, I.; Allen, C.; and Hospedales, T. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5185–5194.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2022. Geometry interaction knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 5521–5529.

Chami, I.; Wolf, A.; Juan, D.-C.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6901–6914.

Chen, S.; Liu, X.; Gao, J.; Jiao, J.; Zhang, R.; and Ji, Y. 2021. HittER: Hierarchical Transformers for Knowledge Graph Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10395–10407.

Dai, G.; Wang, X.; Zou, X.; Liu, C.; and Cen, S. 2022. MRGAT: Multi-Relational Graph Attention Network for knowledge graph completion. *Neural Networks*, 154: 234–245.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Ganea, O.; Bécigneul, G.; and Hofmann, T. 2018. Hyperbolic neural networks. *Advances in neural information processing systems*, 31.

Ge, X.; Wang, Y. C.; Wang, B.; and Kuo, C.-C. J. 2023. Compounding Geometric Operations for Knowledge Graph Completion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6947–6965.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. PMLR.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.

Javaloy, A.; Martin, P. S.; Levi, A.; and Valera, I. 2023. Learnable Graph Convolutional Attention Networks. In *The Eleventh International Conference on Learning Representations*.

Kaiser, M.; Saha Roy, R.; and Weikum, G. 2021. Reinforcement learning from reformulations in conversational question answering over knowledge graphs. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 459–469.

Keizer, S.; Guhe, M.; Cuayáhuitl, H.; Efstathiou, I.; Engelbrecht, K.-P.; Dobre, M.; Lascarides, A.; Lemon, O.; et al. 2017. Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents. 480–484. EACL.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 1–15.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, R.; Cao, Y.; Zhu, Q.; Bi, G.; Fang, F.; Liu, Y.; and Li, Q. 2022. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 5781–5791.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Meng, Y.; Huang, J.; Wang, G.; Zhang, C.; Zhuang, H.; Kaplan, L.; and Han, J. 2019. Spherical text embedding. *Advances in neural information processing systems*, 32.

Nathani, D.; Chauhan, J.; Sharma, C.; and Kaul, M. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4710–4723.

Nguyen, T. D.; Nguyen, D. Q.; Phung, D.; et al. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 327–333.

Nickel, M.; and Kiela, D. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 809–816.

Pan, Z.; and Wang, P. 2021. Hyperbolic hierarchy-aware knowledge graph embedding for link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2941–2948.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Shang, B.; Zhao, Y.; Liu, J.; Liu, Y.; and Wang, C. 2023a. A contrastive knowledge graph embedding model with hierarchical attention and dynamic completion. *Neural Computing and Applications*, 35(20): 15005–15018.

Shang, B.; Zhao, Y.; Liu, Y.; and Wang, C. 2023b. Attention-based exploitation and exploration strategy for multi-hop knowledge graph reasoning. *Information Sciences*, 653: 119787.

Shang, B.; Zhao, Y.; Wang, D.; and Liu, J. 2023c. Relation-Aware Multi-Positive Contrastive Knowledge Graph Completion with Embedding Dimension Scaling. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 878–888.

Song, T.; Luo, J.; and Huang, L. 2021. Rot-pro: Modeling transitivity by projection in knowledge graph embedding. *Advances in Neural Information Processing Systems*, 34: 24695–24706.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; and Gamon, M. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1499–1509.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, 2071–2080. PMLR.

Ungar, A. A. 2008. *Analytic hyperbolic geometry and Albert Einstein's special theory of relativity*. World Scientific.

Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; and Talukdar, P. 2020a. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3009–3016.

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. 2020b. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, S.; Wei, X.; Nogueira dos Santos, C. N.; Wang, Z.; Nallapati, R.; Arnold, A.; Xiang, B.; Yu, P. S.; and Cruz, I. F. 2021a. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*, 1761–1771.

Wang, X.; Huang, T.; Wang, D.; Yuan, Y.; Liu, Z.; He, X.; and Chua, T.-S. 2021b. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, 878–887.

Wang, Y.-C.; Ge, X.; Wang, B.; and Kuo, C.-C. J. 2023. Greenkgc: A lightweight knowledge graph completion method.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Xiao, H.; Huang, M.; and Zhu, X. 2016a. From one point to a manifold: knowledge graph embedding for precise link prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 1315–1321.

Xiao, H.; Huang, M.; and Zhu, X. 2016b. TransG: A Generative Model for Knowledge Graph Embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2316–2325.

Xiao, H.; Liu, X.; Song, Y.; Wong, G. Y.; and See, S. 2022. Complex Hyperbolic Knowledge Graph Embeddings with Fast Fourier Transform. *arXiv preprint arXiv:2211.03635*.

Xiong, C.; Power, R.; and Callan, J. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, 1271–1279.

Yang, B.; Yih, S. W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*.

Yang, J.; Ying, X.; Shi, Y.; Tong, X.; Wang, R.; Chen, T.; and Xing, B. 2022. Knowledge graph embedding by adaptive limit scoring loss using dynamic weighting strategy. In *Findings of the Association for Computational Linguistics: ACL 2022*, 1153–1163.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3065–3072.

Zhang, Z.; Wang, J.; Ye, J.; and Wu, F. 2022. Rethinking graph convolutional networks in knowledge graph completion. In *Proceedings of the ACM Web Conference 2022*, 798–807.