HIGH TECHNOLOGY LETTERS | Vol. 30 No. 3 | Sep. 2024 | pp. 271-279

doi:10. 3772/j. issn. 1006-6748. 2024. 03. 006

Hyperbolic hierarchical graph attention network for knowledge graph completion①

XU Hao (许 浩), CHEN Shudong②, QI Donglin, TONG Da, YU Yong, CHEN Shuai
(Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, P. R. China)
(University of Chinese Academy of Sciences, Beijing 100049, P. R. China)

Abstract

Utilizing graph neural networks for knowledge embedding to accomplish the task of knowledge graph completion (KGC) has become an important research area in knowledge graph completion. However, the number of nodes in the knowledge graph increases exponentially with the depth of the tree, whereas the distances of nodes in Euclidean space are second-order polynomial distances, whereby knowledge embedding using graph neural networks in Euclidean space will not represent the distances between nodes well. This paper introduces a novel approach called hyperbolic hierarchical graph attention network (H2GAT) to rectify this limitation. Firstly, the paper conducts knowledge representation in the hyperbolic space, effectively mitigating the issue of exponential growth of nodes with tree depth and consequent information loss. Secondly, it introduces a hierarchical graph attention mechanism specifically designed for the hyperbolic space, allowing for enhanced capture of the network structure inherent in the knowledge graph. Finally, the efficacy of the proposed H2GAT model is evaluated on benchmark datasets, namely WN18RR and FB15K-237, thereby validating its effectiveness. The H2GAT model achieved 0.445, 0.515, and 0.586 in the Hits@1, Hits@3 and Hits@10 metrics respectively on the WN18RR dataset and 0.243, 0.367 and 0.518 on the FB15K-237 dataset. By incorporating hyperbolic space embedding and hierarchical graph attention, the H2GAT model successfully addresses the limitations of existing hyperbolic knowledge embedding models, exhibiting its competence in knowledge graph completion tasks.

Key words: hyperbolic space, link prediction, knowledge graph embedding, knowledge graph completion (KGC)

0 Introduction

Knowledge graphs are semantic networks that reveal the relations between entities. Many large scale knowledge graphs have been constructed, such as Freebase[1], DBpedia[2] and YAGO3[3], which have been widely used in many downstream tasks such as recommendation systems[4], information retrieval[5] and question answering (QA)[6].

However, most knowledge graphs face the challenge of incompleteness, such as missing entities in triples and the existence of incorrect triples. Several related approaches have been proposed to address these issues in knowledge graph completion, which include translation-based models (TransE[7], TransD[8]), semantic-based models (Dismult[9], Complex[10]), and convolutional neural networks (CNN)-based models (ConvE[11], JointE[12]). In recent years, to leverage the connectivity structure information inherent in knowledge graphs, some researchers[13-14] have introduced graph neural networks into the task of knowledge graph completion to capture the connectivity structure information. KBGAT[15] combines graph neural networks with attention mechanisms to assign different weights to different relations. Subsequent work such as DisenceGAT[16] and EIGAT[17] aim to improve upon KBGAT by focusing on local neighborhood information. However, most existing graph neural networks (GNN)-based methods for knowledge graph completion (KGC) exhibit the following two limitations. Firstly, since the number of nodes in a tree structure grows exponentially with depth, representing complex knowledge graph structures in traditional Euclidean space using polynomial distance measures may result in significant information loss. Secondly, many previous studies frequent-

① Supported by the Beijing Municipal Science and Technology Program (No. Z231100001323004).
② To whom correspondence should be addressed. E-mail: chenshudong@ime.ac.cn.
Received on Oct. 26, 2023

ly emphasize capturing local neighborhood relations while disregarding the explicit consideration of the hierarchical organization within the graph. Consequently, they may fail to effectively capture high level dependencies and structural patterns present in the graph data.

To address the above two problems, this paper proposes a new hierarchical attention model based on non-Euclidean distance. Specifically, for the first problem, some work has been proposed to embed knowledge graphs in hyperbolic space, such as HS-KGCN[18], which constructed a graph convolution model in hyperbolic space, but they did not consider the hierarchical structure in knowledge graphs. Inspired by the exponential growth of distance in hyperbolic space, this paper constructs a graph attention model in imitation of the hyperbolic distance function, and specifically, the model calculates the attention scores based on the different entities corresponding to the relations to obtain the vector representation of the relations, and calculates the attention scores for the different relations to obtain the vector representation of the central nodes.

In summary, the main contributions of this paper are summarized as follows.

(1) Different from most existing KGC models, the hyperbolic distance called geodesic in hyperbolic space is used instead of the original Euclidean distance to solve the distortion problem caused by embedding knowledge graphs in Euclidean space.

(2) This paper proposes a hierarchical multi-relation graph attention network design that incorporates an attention mechanism to effectively aggregate relations and entities. This approach allows for the fusion of adjacent features, thereby enhancing interpretability.

(3) For the link prediction task, H2GAT is tested on two datasets, WN18RR and FB15K-237, and the results show that H2GAT outperforms the previous link prediction model in all metrics.

1 Related work

In recent years, many work has used knowledge embedding methods to study link prediction tasks, which can be classified into Euclidean space, Complex space and Hyperbolic space depending on the embedding space.

1.1 Euclidean space

Most of the previous approaches for tasks of link prediction using knowledge embedding methods have predominantly focused on Euclidean space and can be categorized based on the methods that they employed namely distance-based, tensor-decomposition-based, CNN-based, and GNN-based methods.

1.1.1 Translation-based models

TransE[7] is widely regarded as the most classical translational model in the field. TransE embeds entities and relations so that they can follow $h + r = t$, where (h, r, t) forms a triple in knowledge graphs, h represents the embedding of head entities, r represents the embedding of relations, and t represents the embedding of tail entities. However, there are limitations in TransE when dealing with reflexive/one-to-many/many-to-one/many-to-many relations. As a result, several of its extensions have been proposed to deal with the problem. TransH[19] projects the entity embeddings into a hyperplane by the norm vector of the hyperplane to overcome the shortcomings of the TransE, while maintaining the same computational complexity. On the other hand, TransR[20] takes a different approach by mapping entities into a specific relation space, which allows different entities to have different vector representations in different relation spaces.

1.1.2 Tensor-decomposition-based models

Rescal[21] is the most classical matrix decomposition model, which encodes entities as vectors and relations as a three dimensional matrix, and uses a bilinear function to score the triple. To address the problem of large computational effort of the Rescal model, Dismult[9] improves on Rescal by replacing the original relation matrix with a diagonal matrix, thus greatly reducing the training parameters.

1.1.3 CNN-based models

ConvE[11] is the first model to apply CNN models in link prediction tasks. ConvE uses 2D convolution layers over entity embeddings and relation embeddings and multiple layers of nonlinear features to predict missing relations or entities in knowledge graphs. Different from ConvE, ConvKB[22] uses a 3-column matrix to represent each triple (head entity, relation, tail entity) which is then passed through a convolutional layer where multiple filters operate on it to generate various feature maps. These feature maps are subsequently concatenated to form a single feature vector which is used to compute the scores of triples.

1.1.4 GNN-based models

Relational graph convolutional networks (RGCN)[23], vectorized relational graph convolutional networks (VRGCN)[24], and structure-aware convolutional network (SACN)[13] are early models that introduce graph convolutional networks (GCN) into link prediction tasks to learn representations of entities and relations in knowledge graphs. KBGAT is the first model to utilize graph attention mechanisms to learn representations of entities and relations in knowledge graphs. EIGAT[17] further enhances KBGAT by incorporating a global attention mechanism to learn entity embedding. DisenceGAT[16], on the other hand, recognizes that existing knowledge graph embedding methods are insufficient in accurately capturing complex relations. To address this problem, they design macroscopic and microscopic resolution mechanisms and introduce them into the graph attention network to learn adaptive representations of entities. Building upon existing GAT research, H2GAT employs a hierarchical graph attention network structure in the hyperbolic space to obtain more fine-grained entity embedding representations.

1.2 Complex space

In recent years, knowledge embedding models based on complex space have also achieved promising results in link prediction tasks. Complex[10] is the first model to introduce complex-valued vectors and utilize the Hermitian dot product to calculate triple scores. RotatE[25] maps entities and relations to complex space and defines relations as rotations between the head and tail entities. ComplexGCN[26] extends the standard GCN to the complex space by establishing a complex graph convolutional model.

1.3 Hyperbolic space

Although the performance of existing knowledge embedding models for tasks of link prediction has significantly improved, they are limited by the spaces in which they are embedded, making it difficult to capture deep-level information in knowledge graphs. Addiionally, as the depth increases, the number of nodes grows exponentially while the distance in Euclidean space only grows linearly, resulting in distortions in existing entity and relation representations. In response to this, hyperbolic networks have emerged. Models such as Murp[27] and HyperKG[28] capture the hierarchical structure and deep-level information of heterogeneous graphs in hyperbolic space. HSKGCN[18] proposes a hyperbolic graph neural network embedding method, initializing relation and entity embeddings in hyperbolic space. Information aggregation is performed in the tangent space of hyperbolic space to capture the local neighborhood information and structural features of relations and entities in knowledge graphs.

2 Hyper graph attention network

This section will introduce hyperbolic hierarchical graph attention network. Firstly, this section will discuss the mathematical knowledge required for graph embedding in the hyperbolic space. Then, this section will present the hyperbolic hierarchical graph attention network. Finally, this section will describe the decoder model that calculates triple scores in the task of link prediction.

2.1 Hyperbolic space

Hyperbolic space is a space with a constant negative curvature, where the curvature is negative at any point in the space. Common models of hyperbolic space include the Lorentz model, the Klein model, the Hemisphere model, the Poincare ball model, and the Poincare half-space model[29]. In this paper, encoder adopts the Poincare ball model, which is the most widely used hyperbolic space model.

The definition of addition in hyperbolic space is given as

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

where, $\oplus_c$ is called Mobius addition, and when $c = 0$, $x \oplus y = x + y$ is the addition in Euclidean space.

The definition of distance in hyperbolic space is termed as geodesic which differs from the distance defined in the Euclidean space and is determined by the following equation.

$$d_c(x,y) = (2/\sqrt{c})\tanh^{-1}(\sqrt{c}\,\|-x \oplus_c y\|) \quad (2)$$

where, x and y are two points on the hyperbolic space and $d_c(x,y)$ is the distance in the hyperbolic space between them.

The exponential map function exp_0^c primarily maps vectors from Euclidean space to the hyperbolic space where c is the curvature in hyperbolic space. On the other hand, the logarithmic map function log_0^c maps vectors from the hyperbolic space back to the Euclidean space. These map functions are essential for converting between representations in the Hyperbolic space and the Euclidean space and are given by

$$log_0^c(y) = \tanh^{-1}(\sqrt{c}\,\|y\|)\frac{y}{\sqrt{c}\,\|y\|} \quad (3)$$

$$exp_0^c(v) = \tanh(\sqrt{c}\,\|v\|)\frac{v}{\sqrt{c}\,\|v\|} \quad (4)$$

2.2 Encoder

Fig. 1 shows the framework of the H2GAT model. H2GAT follows an encoder-decoder framework. First, the embeddings of entities and relations are initialized randomly on the hyperbolic space. Then the embeddings on the hyperbolic space are projected to the Euclidean space and are updated by the message-passing mechanisms in different layers. In the message-passing mechanisms, the entity-level attention and relation-level attention coefficients are combined to assign different weights to tail entities which are aggregated to update the embeddings of head entities. Next, embeddings from different layers are aggregated and projected back to Euclidean space. Finally, the encoder uses the residual connections on the hyperbolic space and outputs the entity embeddings to the decoder. The decoder is a knowledge graph completion model and this paper uses Murp as the decoder.

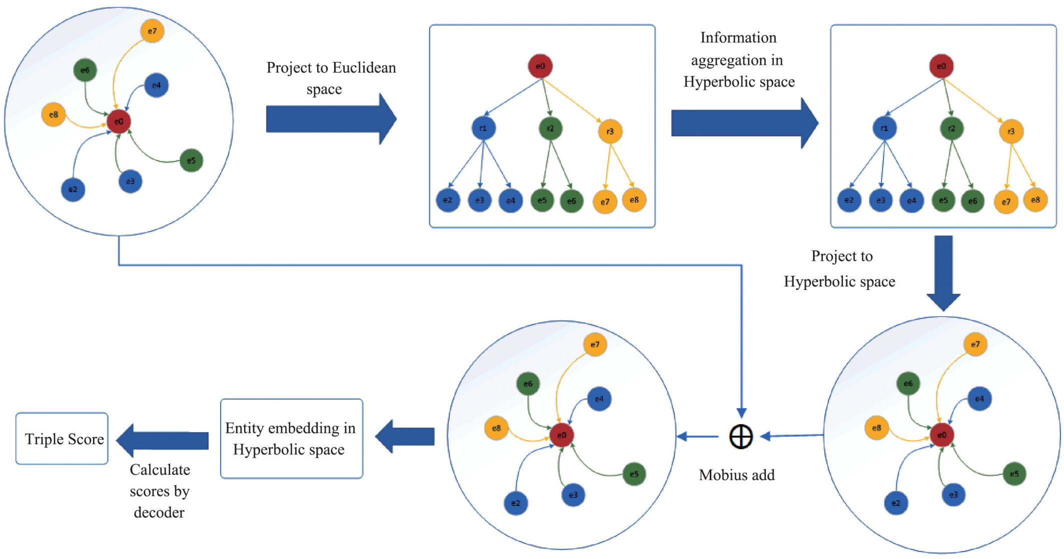

Fig. 1 The framework of H2GAT mode

Let e_i^l and r_i^l denote the embedding representations of the ith entity and ith relation, respectively, in the lth layer of the attention network, and let E_i^l and R_i^l represent the embedding representations of the entity and relation in the Poincare space. First, project the input randomly initialized hyperbolic space vector into Euclidean space as

$$e_i^0 = log_0^c(E_i^0) = \tanh^{-1}(\sqrt{c}\,\|E_i^0\|)\frac{E_i^0}{\sqrt{c}\,\|E_i^0\|} \quad (5)$$

For the lth layer of the hyperbolic hierarchical graph attention network, there are two inputs: E_p^l and R_p^l. First compute entity-level attention aggregation, where the entity-level attention believes that different tail entities should be assigned different attention weights even under the same relation. The tail entities within the same relation can be considered as a group, and the entity-level attention weights $\alpha_{r,t}$ are calculated as

$$e_{r,t}^l = \boldsymbol{W}_1[\boldsymbol{r}_i \| \boldsymbol{e}_t] \quad (6)$$

$$\alpha_{r,t} = \frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{a}_1 \cdot e_{r_i,t}^l))}{\sum_{k \in N_i^r} \exp(\mathrm{LeakyReLU}(\boldsymbol{a}_1 \cdot e_{r_i,k}^l))} \quad (7)$$

where, $\boldsymbol{W}_1$ and $\boldsymbol{a}_1$ denote the linear transformation matrices, $\boldsymbol{r}$ and $\boldsymbol{e}_t$ denote the embedding vectors of the relation and the tail entity, respectively, and the attention coefficient $\alpha_{r,t}$ denotes the weight that relation $\boldsymbol{r}$ assigned to the tail entity $\boldsymbol{e}_t$.

In the relation-level attention, different first-order relations are assigned different attention weights for the head entity h. The calculation of relation-level attention can be expressed by the following equation:

$$e_{h,r_i}^l = \boldsymbol{W}_2[\boldsymbol{e}_h \| \boldsymbol{r}_i] \quad (8)$$

$$\boldsymbol{\alpha}_{h,r_i} = \frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{a}_2 h\, e_{h,r_i}^l))}{\sum_{k \in N_h} \exp(\mathrm{LeakyReLU}(\boldsymbol{a}_2 h\, e_{h,r_k}^l))} \quad (9)$$

where, $\boldsymbol{W}_2$ and $\boldsymbol{a}_2$ denote the linear transformation matrices, $\boldsymbol{e}_h$ and $\boldsymbol{r}_i$ denote the embedding vectors of the head entity and the relation, respectively, and the attention coefficient $\boldsymbol{\alpha}_{h,r_i}$ denotes the weight that the head entity $\boldsymbol{e}_h$ assigned to the relation $\boldsymbol{r}_i$.

After obtaining the entity-level and relation-level attention coefficients, the message-passing can be performed in the graph network. The $l + 1$ layer embeddings of head entities are updated by aggregating the embeddings of the tail entities, which can be given as

$$e_h^{l+1} = \sum_{r_i \in N_h} \sum_{t_k \in N_h^{r_i}} \boldsymbol{\alpha}_{h,r_i} h\, \alpha_{r_i,t_k} h\, e_{t_k}^l \quad (10)$$

As used in KBGAT, in order to enhance the stability of the model training and to capture more structural features at different levels, this paper uses multi-head attention mechanism and average the final entity embeddings from multiple heads, which is given as

$$e^l = \frac{1}{K}\sum_{k=1}^{K} e_k^l \tag{11}$$

where, e_k^l means the entity embeddings of the kth head in the lth layer. Also, in order to minimize the information loss that occurs during graph message-passing, this paper also uses residual connections:

$$e' = e^l + e^0 \tag{12}$$

where, e^l means the output entity embeddings of the last layer and e^0 means the initial input entity embeddings to the model. After the encoding of entities and relations in Euclidean space is completed, they are then projected back to the hyperbolic space, thus completing the embedding representation of entities and relations in hyperbolic space.

$$E_i^l = exp_0^c(e_i^l) = \tanh(\sqrt{c}\,\|e_i^l\|)\frac{e_i^l}{\sqrt{c}\,\|e_i^l\|} \tag{13}$$

The training objective for the encoder is primarily inspired by TransE[7]. In TransE, the main principle is to satisfy $h + r \approx t$, where h and t represent the embeddings of head and tail entity, respectively, and r represents the embeddings of relations. By following this idea, the objective is to minimize the distance in hyperbolic space by learning the embeddings of entities and relations, which is given by

$$d_{hrt} = d_c(h \oplus r, t) \tag{14}$$

where, $\oplus$ is the Mobius addition and d_c is the distance in hyperbolic space.

Similar to KBGAT[15], this paper trains the encoder using the hinge loss function:

$$L(\Omega) = \sum_{t\in T}\sum_{t'\in T'}\max\{d_{hrt} - d_{hrt'} + \gamma, 0\} \tag{15}$$

where, γ is the margin hyperparameter, T is the set of valid triples, and T' is the set of invalid triples.

2.3 Decoder

This paper uses the embedding of entities and relationships obtained by the encoder in hyperbolic space to initialize the embedding of entities and relationships in the decoder to achieve better performance in the knowledge graph completion task. H2GAT uses Murp as decoder, which is a variation of the bilinear model in hyperbolic space. Unlike the direct use of inner product in Euclidean space to calculate the similarity between the head entity and the tail entity, Murp uses a relation matrix to transform the head entity, and the embeddings of tail entities is Mobius summed with the embeddings of relations obtained from H2GAT. Finally, the hyperbolic space distance is calculated to obtain the final score, and the score function is given as

$$\begin{aligned}\varphi_{MuRP}(h, \boldsymbol{r}, t) &= -d_B(E_h, E_t)^2 + b_h + b_t \\ &= -d_B(exp_0^c(\boldsymbol{R}\,log_0^c(E_h)), \\ &\quad E_t \oplus_c \boldsymbol{r}_h)^2 + b_h + b_t\end{aligned} \tag{16}$$

where, E_h and E_t are embeddings of the head entity h and the tail entity t in hyperbolic space, $\boldsymbol{r}_h$ is the translation vector of relation $\boldsymbol{r}$ in hyperbolic space and $\boldsymbol{R}$ is the diagonal relation matrix to transform the embeddings of the head entity in tangent space. The loss function of decoder is the soft-margin loss which is given by

$$\pounds(y, \varphi) = -\frac{1}{N}\sum_{i=1}^{N}(y_i\log(\varphi_i) + (1 - y_i)\log(1 - \varphi_i)) \tag{17}$$

where, φ_i is the score of the ith triple, y_i is the label of the ith triple that indicates whether the triple is positive or negative and N is the number of training samples.

3 Experiments

Link prediction primarily predicts the missing head or tail entity given an incomplete triple. In this section, this paper evaluates the performance of H2GAT on the task of link prediction.

3.1 Dataset

To verify the effectiveness of H2GAT model in link prediction tasks, this paper performs experiments on two datasets, FB15k-237[30] and WN18RR[31]. FB15k-237 and Win18RR are two widely used datasets in link prediction tasks and contain a large number of one-to-one, one-to-many, many-to-one, and many-to-many relations. Statistics on the number of entities and relations in the two datasets are shown in Table 1.

Table 1 Statistics of WN18RR and FB15K237

Dataset	Triples	Relation	Unique entity	Train triples	Valid triples	Test triples
WIN18RR	173 670	11	40 559	86 835	3 034	3 134
FB15K237	542 330	200	14 505	272 115	17 535	20 466

3.2 Baselines

To verify the validity of H2GAT, H2GAT is compared with some classical models, such as TransE[7] based on translation model, Dismult[9] based on tensor decomposition, ConvE[11] based on CNN, ATTH[32] and MuRP[27] based on embeddings in hyperbolic space, and some state-of-the-art GNN based models, such as SACN[13], CompGCN[14], KMAE[33], HS-KGCN[18], and ComplexGCN[26].

3.3 Experiment set

This paper employs the Riemannian Adam optimizer to train the encoder and decoder in hyperbolic space on a Nvidia RTX 3090Ti. The settimgs of experimental parameters are shown in Table 2.

Table 2 The settings of experimental parameters

Parameters	WN18RR
Number of heads in attention mechanism	4
Learning rate of encoder	0.000 5
Learning rate of decoder	50
The embedding size of entities and relations	64
The output embedding size of encoder	64
The ratio of positive to negative samples	0.1

For the encoder, the embedding size of entities and relations is set as 64 and the number of heads in the multi-head attention mechanism is set to 4. The learning rate is set as 0.000 5. For the decoder, the embedding size of entities and relations is also set as 64. The learning rate is set as 50. For both the encoder and decoder, L2 regularization with $\lambda = 0.0005$ is applied, and all training parameters are initialized randomly. To address the issue of subpar performance in predicting head entities in link prediction, this paper augments the original dataset by creating reverse triplets for each triplet. The ratio of positive to negative samples is chosen as 1/10.

During the testing phase, this paper performed entity replacements for the tail entities in the testing triples using all possible entities. This paper computed the scores for all replaced triples and subsequently ranked these candidate entities based on these scores. This paper evaluated H2GAT model using the following metrics: (1) *MRR* (mean reciprocal rank, the average reciprocal rank of all correct triples); (2) *Hits@n* (the proportion of correct triplets that are ranked within the top n positions).

3.4 Results and analysis

3.4.1 Analysis of model performance on link prediction

Table 3 and Table 4 present the experimental results of the proposed H2GAT model compared with other classical models in the link prediction task. By analyzing the above experimental results, the following conclusions can be drawn.

Table 3 Link prediction on WN18RR

	WN18RR			
Model	*MRR*	*Hits*@1	*Hits*@3	*Hits*@10
TransE[7]	0.226	-	-	0.501
DisMult[9]	0.430	0.390	0.440	0.490
ConvE[11]	0.430	0.400	0.440	0.520
SACN[13]	0.540	0.430	0.480	0.540
MURP[27]	0.477	0.438	0.489	0.555
ATTH[32]	0.466	0.438	0.489	0.555
CompGCN[14]	0.479	0.443	0.494	0.546
KMAE[33]	0.537	0.415	0.465	0.524
HSKGCN[18]	0.478	0.435	0.496	0.557
ComplexGCN[26]	0.455	0.423	0.468	0.516
H2GAT	0.487	0.445	0.515	0.586

Firstly, comparing the H2GAT with the decoder only model MURP, H2GAT shows improvements over Murp of 0.004, 0.027, and 0.031 in the Hits@1, Hits@3, and Hits@10 metrics on the WN18RR dataset. Similarly, on the FB15K237 dataset, H2GAT shows improvements of 0.016, 0.021, and 0.025 in the Hits@1, Hits@3, and Hits@10 metrics, respectively. This result demonstrates the effectiveness of the encoder and the information learned by H2GAT in the hyperbolic space is valuable.

Secondly, compared with the HSKGCN which also utilizes MURP as the decoder for knowledge embeddings in the hyperbolic space, H2GAT outperforms HSKGCN on all datasets. Specifically, H2GAT shows improvements of 0.010, 0.019, and 0.029 in the Hits@1, Hits@3, and Hits@10 metrics on the WN18RR dataset and shows improvements of 0.003, 0.008, and 0.014 in the Hits@1, Hits@3, and Hits@10 metrics on the FB15K-237 dataset. This result indicates the superiority of the encoder of H2GAT and can be attributed to H2GAT's ability to capture hierarchical information of nodes and effectively utilize local neighborhood information of central nodes for embeddings. As a result, H2GAT provides more effective entity embeddings for the decoder model, contributing to its superior performance on the link prediction task.

Table 4 Link prediction on FB15K-237

FB15K-237				
Model	*MRR*	*Hits*@1	*Hits*@3	*Hits*@10
TransE[7]	0.294	-	-	0.465
DisMult[9]	0.241	0.155	0.263	0.419
ConvE[11]	0.325	0.237	0.356	0.501
SACN[13]	0.360	0.260	0.390	0.540
MURP[27]	0.324	0.227	0.346	0.506
ATTH[32]	0.324	0.236	0.354	0.501
CompGCN[14]	0.355	0.264	0.390	0.535
KMAE[33]	-	0.240	0.358	0.502
HSKGCN[18]	0.327	0.240	0.359	0.504
ComplexGCN[26]	0.338	0.245	0.371	0.524
H2GAT	0.331	0.243	0.367	0.518

Thirdly, as a graph attention model that is embedded in the hyperbolic space, H2GAT is capable of capturing neighborhood connectivity information of central nodes. This gives it an advantage over traditional knowledge embedding models such as TransE and ConvE. In comparison to other graph neural network models such as SACN, RGCN, and CompGCN, H2GAT benefits from embedding in the hyperbolic space, which avoids the information loss caused by the exponential growth of nodes with tree depth in Euclidean space. Consequently, H2GAT outperforms traditional graph neural network models as well.

3.4.2 Effect of super parameters of the model

This paper discusses the effect of learning rate of the H2GAT. With an embedding dimension of 64, this paper testes the effect of different learning rates (0.000 1, 0.000 5 and 0.001 0) on the model effectiveness, and the results are shown in Fig. 2. It can be seen that the model works best at a learning rate of 0.000 5, which indicates that the optimal solution cannot be obtained when the learning rate is too small or too large.

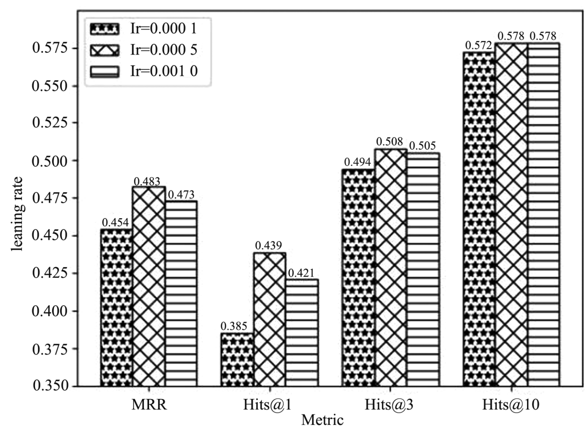

Fig. 2 The influence of the learning rate

3.4.3 The convergence study of H2GAT

This paper discusses the convergence of H2GAT. Fig. 3 illustrates the changes in the loss of MURP and H2GAT over epochs. The trend of loss with epoch clearly demonstrates that, when trained for an equal number of epochs, H2GAT consistently achieves a lower loss compared with MURP. This finding highlights the accelerated convergence process of the MURP model by leveraging the encoder of H2GAT to encode entities. Hence, this observation provides additional empirical evidence to support the effectiveness of H2GAT.

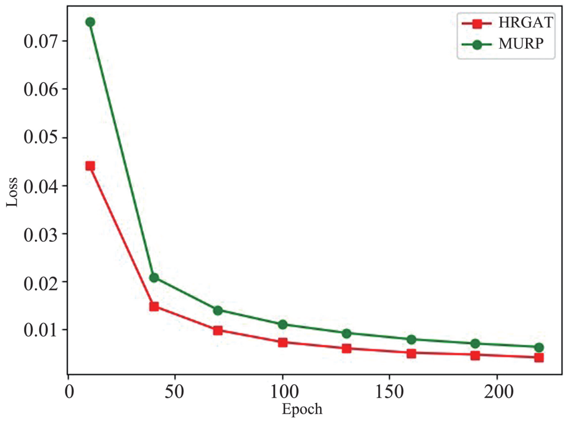

Fig. 3 The convergence study of MURP and H2GAT in WN18RR

4 Conclusions

This paper introduces the H2GAT model, a novel hierarchical graph attention network for embeddings of entities and relations in hyperbolic space. The model accomplishes hierarchical attention aggregation at the relation level and entity level in the tangent space of the Poincare disk model and projects it back to the hyperbolic space to train the embeddings of entities and relations using hyperbolic distance. Experimental results show that H2GAT has achieved state-of-the-art results on WN18RR and FB15K-237 in all metrics. Compared with the HSKGCN which also utilizes MURP as the decoder for knowledge embeddings in the hyperbolic space, H2GAT achieved better results which means the encoder of H2GAR can capture more local neighborhood information and hierarchical structural features of relations and entities in knowledge graphs. For future work, it can be planed to embed H2GAT in different hyperbolic model space and to combine the features captured in different space.

References

[1] BOLLACKER K, EVANS C, PARITOSH P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. Vancouver, Canada: Association for Computing Machinery, 2008: 1247-1250.

[2] AUER S, BIZER C, KOBILAROV G, et al. DBpedia: a nucleus for a web of open data[C]//International Semantic Web Conference. Berlin, Germany: Springer, 2007: 722-735.

[3] MAHDISOLTANI F, BIEGA J, SUCHANEK F M. YAGO3: a knowledge base from multilingual wikipedias [C]//The 7th Biennial Conference on Innovative Data Systems Research. Asilomar, USA: CIDR, 2013: 1-11.

[4] MUSTO C, BASILE P, SEMERARO G. Embedding knowledge graphs for semantics-aware recommendations based on DBpedia[C]//The 27th Conference on User Modeling, Adaptation and Personalization. Larnaca, Cyprus: ACM, 2019: 27-31.

[5] 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.

[6] WANG R, WANG M, LIU J, et al. Leveraging knowledge graph embeddings for natural language question answering[C]//Database Systems for Advanced Applications: 24th International Conference. Chiang Mai, Thailand: DASFAA, 2019: 659-675.

[7] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data [J]. Advances in Neural Information Processing Systems, 2013, 26: 2787-2795.

[8] 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: ACM, 2015: 687-696.

[9] YANG B, YIH W, HE X, et al. Embedding entities and relations for learning and inference in knowledge bases [EB/OL]. (2015-08-29) [2023-10-26]. https://arxiv.org/pdf/1412.6575.pdf.

[10] TROUILLON T, WELBL J, RIEDEL S, et al. Complex embeddings for simple link prediction[C]//International Conference on Machine Learning. New York, USA: PMLR, 2016: 2071-2080.

[11] DETTMERS T, MINERVINI P, STENETORP P, et al. Convolutional 2d knowledge graph embeddings[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA: AAAI Press, 2018: 1-9.

[12] ZHOU Z, WANG C, FENG Y, et al. JointE: jointly utilizing 1D and 2D convolution for knowledge graph embedding [J]. Knowledge-Based Systems, 2022, 240: 108100.

[13] SHANG C, TANG Y, HUANG J, et al. End-to-end structure-aware convolutional networks for knowledge base completion[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Hawaii, USA: AAAI Press, 2019: 3060-3067.

[14] VASHISHTH S, SANYAL S, NITIN V, et al. Composition-based multi-relational graph convolutional networks [EB/OL]. (2020-01-18) [2023-10-26]. https://arxiv.org/pdf/1911.03082v2.pdf.

[15] NATHANI D, CHAUHAN J, SHARMA C, et al. Learn-

ing attention-based embeddings for relation prediction in knowledge graphs[EB/OL]. (2019-06-04) [2023-10-26]. https://arxiv.org/pdf/1906.01195.pdf.

[16] WU J, SHI W, CAO X, et al. DisenKGAT: knowledge graph embedding with disentangled graph attention network[C]//Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Virtual: ACM, 2021: 2140-2149.

[17] ZHAO Y, ZHOU H, XIE R, et al. Incorporating global information in local attention for knowledge representation learning[C]//Findings of the Association for Computational Linguistics. Bangkok, Thailand: ACL-IJCNLP, 2021: 1341-1351.

[18] YAO S, PI D, CHEN J. Knowledge embedding via hyperbolic skipped graph convolutional networks[J]. Neurocomputing, 2022, 480: 119-130.

[19] WANG Z, ZHANG J, FENG J, et al. Knowledge graph embedding by translating on hyperplanes[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2014, 28(1): 1112-1119.

[20] LIN Y, LIU Z, SUN M, et al. Learning entity and relation embeddings for knowledge graph completion[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2015, 29(1): 1-7.

[21] 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. Lyon, France: ACM, 2012: 271-280.

[22] NGUYEN D Q, NGUYEN T D, NGUYEN D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network[EB/OL]. (2018-03-13) [2023-10-26]. https://arxiv.org/pdf/1712.02121.pdf.

[23] SCHLICHTKRULL M, KIPF T N, BLOEM P, et al. Modeling relational data with graph convolutional networks [C]//The Semantic Web: 15th International Conference. Crete, Greece: Springer International Publishing, 2018: 593-607.

[24] YE R, LI X, FANG Y, et al. A vectorized relational graph convolutional network for multi-relational network alignment[C]//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China: IJCAI, 2019: 4135-4141.

[25] SUN Z, DENG Z H, NIE J Y, et al. Rotate: knowledge graph embedding by relational rotation in complex space [EB/OL]. (2019-02-26) [2023-10-26]. https://arxiv.org/pdf/1712.02121.pdf.

[26] ZEB A, SAIF S, CHEN J, et al. Complex graph convolutional network for link prediction in knowledge graphs [J]. Expert Systems with Applications, 2022, 200: 116796.

[27] BALAZEVIC I, ALLEN C, HOSPEDALES T. Multi-relational poincaré graph embeddings[J]. Advances in Neural Information Processing Systems, 2019, 32: 1-11.

[28] KOLYVAKIS P, KALOUSIS A, KIRITSIS D. Hyperbolic knowledge graph embeddings for knowledge base completion[C]//The Semantic Web: 17th International Conference. Crete, Greece: Springer International Publishing, 2020: 199-214.

[29] CANNON J W, FLOYDW J, KENYON R, et al. Hyperbolic Geometry[J]. Flavors of Geometry, 1997, 31: 59-115.

[30] TOUTANOVA K, CHEN D, PANTEL P, et al. Representing text for joint embedding of text and knowledge bases[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: EMNLP, 2015: 1499-1509.

[31] DIEFENBACH D, SINGH K, MARET P. Wdaquacore1: a question answering service for RDF knowledge bases[C]// Proceedings of the Web Conference 2018. Geneva, Switzerland: ACM, 2018: 1087-1091.

[32] CHAMI I, WOLF A, JUAN D C, et al. Low-dimensional hyperbolic knowledge graph embeddings [EB/OL]. (2020-05-01) [2023-10-26]. https://arxiv.org/pdf/2005.00545v1.pdf.

[33] JIANG D, WANG R, YANG J, et al. Kernel multi-attention neural network for knowledge graph embedding[J]. Knowledge-Based Systems, 2021, 227: 107188.

XU Hao, born in 1999. He is pursuing his M. S. degree in Institute of Microelectronics of the Chinese Academy of Sciences. He also received his B. S. degree from Southeast University in 2021. His research interests include knowledge graph completion, knowledge graph embeddings and recommender systems.