

MulDE: Multi-teacher Knowledge Distillation for Low-dimensional Knowledge Graph Embeddings

Kai Wang*

School of Software, the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology
Dalian, Liaoning, China
kai_wang@mail.dlut.edu.cn

Qian Ma

School of Software, the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology
Dalian, Liaoning, China
Ninja666@mail.dlut.edu.cn

Yu Liu

School of Software, the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology
Dalian, Liaoning, China
yuliu@dlut.edu.cn

Quan Z. Sheng

Intelligent Computing Laboratory
Department of Computing
Macquarie University
Sydney, NSW, Australia
michael.sheng@mq.edu.au

ABSTRACT

Link prediction based on knowledge graph embeddings (KGE) aims to predict new triples to automatically construct knowledge graphs (KGs). However, recent KGE models achieve performance improvements by excessively increasing the embedding dimensions, which may cause enormous training costs and require more storage space. In this paper, instead of training high-dimensional models, we propose MulDE, a novel knowledge distillation framework, which includes multiple low-dimensional hyperbolic KGE models as teachers and two student components, namely Junior and Senior. Under a novel iterative distillation strategy, the Junior component, a low-dimensional KGE model, asks teachers actively based on its preliminary prediction results, and the Senior component integrates teachers' knowledge adaptively to train the Junior component based on two mechanisms: *relation-specific scaling* and *contrast attention*. The experimental results show that MulDE can effectively improve the performance and training speed of low-dimensional KGE models. The distilled 32-dimensional model is competitive compared to the state-of-the-art high-dimensional methods on several widely-used datasets.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → *Entity relationship models*.

KEYWORDS

Knowledge graph embeddings, link prediction, knowledge distillation, knowledge graph

*Corresponding author

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW'21, April 19–23, 2021, Ljubljana, Slovenia

© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-8312-7/21/04.

<https://doi.org/10.1145/3442381.3449898>

ACM Reference Format:

Kai Wang, Yu Liu, Qian Ma, and Quan Z. Sheng. 2021. MulDE: Multi-teacher Knowledge Distillation for Low-dimensional Knowledge Graph Embeddings. In *Proceedings of the Web Conference 2021 (WWW'21)*, April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3442381.3449898>

1 INTRODUCTION

Knowledge graphs (KGs), which describe human knowledge as factual triples in the form of (head entity, relation, tail entity), have shown great potential in various domains [27, 35, 42]. Popular real-world KGs such as Yago [28] and DBPedia [4], which typically contain an enormous quantity of entities, are still far from complete [6, 12]. Due to the high manual costs of discovering new triples [23], link prediction based on knowledge graph embeddings (KGE) [8, 39] has drawn considerable attention very recently as a means to overcome this problem. A typical KGE model first represents entities and relations as trainable continuous vectors. Then, given an entity and a relation of a triple (we call 'e-r query'), the model defines a scoring function to measure each candidate in the entity set and outputs the best one [34].

To achieve higher prediction accuracy, recent KGE models generally use high-dimensional embedding vectors up to 200 or even 500 dimensions [29, 40]. However, when we have millions or billions of entities in a KG, the high-dimensional model demands enormous training costs and storage space [3, 25]. This prevents downstream AI applications from updating KG embeddings promptly or being able to be deployed on mobile devices. Recently, several research efforts have drawn attention to this research issue by either improving the low-dimensional models (such as 8 or 32 dimensions) [7], or compressing pre-trained high-dimensional models [25]. However, the former cannot utilize the high-accuracy knowledge from high-dimensional models, and the latter suffers from high pre-training costs and cannot continue training when the KG is modified.

Can we transfer high-accuracy knowledge to a low-dimensional model while avoiding to train high-dimensional models? According to a preliminary experiment, we find that the ensemble of different 64-dimensional hyperbolic KGE models can outperform any single

higher-dimensional models. To this end, we determine to employ those low-dimensional hyperbolic models as multiple teachers and integrate their knowledge to train a smaller KGE model in a Knowledge Distillation (KD) process. Knowledge Distillation [13], which has been rarely applied in the knowledge graph domain, is a technology distilling ‘soft labels’ from a pre-trained big model (teacher) to train a small one (student). In our paper, soft labels come from the scores of candidate triples measured by the KGE model. Compared with a single high-dimensional teacher, we argue that there are at least three benefits of utilizing multiple low-dimensional teachers:

- *Reduce pre-training costs.* The number of parameters for multiple teachers is relatively lower than that of a high-dimensional state-of-the-art model. Besides, the pre-training speed of the former can be further improved by parallel techniques.
- *Guarantee teacher performance.* The ensemble’s accuracy exceeds some high-dimensional models, because prediction errors of a single teacher can be corrected by integrating multiple prediction results.
- *Improve distilling effect.* Based on recent knowledge distillation studies [19], a student model is easier to acquire knowledge from a teacher with a similar size. A teacher in low dimensions is more suitable than the one having hundreds of dimensions.

In this paper, we pre-train multiple low-dimensional hyperbolic KGE models, and propose a novel framework utilizing Multi-teacher knowledge Distillation for knowledge graph Embeddings, named **MulDE**. Different from conventional KD technologies, we design a novel *iterative distillation strategy* and two student components (i.e., *Junior* and *Senior*) in MulDE. In one iteration, the Junior component first predicts top-K candidates for each e-r query, and transfers to multiple teachers. Then, the Senior component integrates teacher results and generates soft labels with two mechanisms: *relation-specific scaling* and *contrast attention*. Finally, Junior receives soft labels and updates its parameters using a knowledge distillation loss. As a result, instead of receiving knowledge in one direction from teachers, the students can seek knowledge actively from teachers and thereby distinguish false-positive entities better.

We conduct extensive experiments on two widely-used datasets, FB15k-237 and WN18RR to validate our proposed model. The results show that MulDE can significantly improve the prediction accuracy and training speed of low-dimensional KGE models. The distilled 32-dimensional model outperforms the state-of-the-art low-dimensional models, and is comparable to some high-dimensional ones. According to ablation experiments, we prove the effectiveness of the iterative distillation strategy and the other major modules in MulDE. We also compare different teacher settings and conclude that using four different 64-dimensional hyperbolic models is optimal.

The rest of the paper is organized as follows. We briefly introduce the background and notations in our work in Section 2. Section 3 details the whole framework and basic components of the MulDE model. Section 4 reports the experimental studies, and Section 5 further discusses the experimental investigations. We discuss the related work in Section 6 and, finally, offer some concluding remarks in Section 7.

2 PRELIMINARIES

2.1 Definitions and Notations

Let E and R denote the set of entities and relations, a knowledge graph \mathcal{G} is a collection of factual triples (e_h, r, e_t) , where $e_h, e_t \in E$ and $r \in R$. N_e and N_r refer to the number of entities and relations, respectively. Given an e-r query $q = (e_{in}, r)$, where $e_{in} \in E$ and $r \in R$, the link prediction task is to find $e_{miss} \in E$, such that (e_{in}, r, e_{miss}) or (e_{miss}, r, e_{in}) should belong to the knowledge graph \mathcal{G} . According to the Cartesian product, the number of all possible e-r queries is equal to $N_e \times N_r$.

Knowledge Graph Embeddings aim to represent each entity $e \in E$ and each relation $r \in R$ as d -dimensional continuous vectors. A KGE model \mathcal{M} defines a scoring function $F : E \times R \times E \rightarrow \mathcal{R}$ to score each triple through embedding vectors. Given an e-r query q , \mathcal{M} outputs a prediction sequence containing sorted scores of all candidate triples, denoted as $\{(c, s) | c \in E, s \in \mathcal{R}\}$. In the sequence, candidates with higher scores are more likely to be true.

For each candidate triple (e_{in}, r, e_c) , \mathcal{M} learns the relation vector r as a transformation between two entity vectors e_{in} and e_c . From a probabilistic model point of view, regarding the e-r query vector q as a transformed entity vector, the plausibility of the triple can be expressed as a joint probability $p(q, e_c)$:

$$p(q, e_c) = \frac{e^{D(q, e_c)}}{\sum_{i=1}^{N_e \times N_r} \sum_{j=1}^{N_e} e^{D(q, e_j)}} \quad (1)$$

where $e \in \mathcal{R}^d$ is the embedding vector of an entity e , the query vector $q = e_{in} \circ r$, \circ refers to any vector transformation, and $D(q, e)$ refers to the distance between two entity vectors in the scoring function F .

2.2 Preliminary Analysis of Low-Dimensional Embeddings

In a KGE model, the parameters of embedding vectors usually account for most of the total parameters. Given a large-scale KG, reducing the embedding dimensions can significantly save storage space. However, small embedding dimensions lead to low representation capability, limiting the prediction accuracy of the KGE model. The information quantity of a low-dimensional embedding vector space is not infinite, and cannot accommodate the information of numerous triples when facing millions of entities.

To the best of our knowledge, there is no theoretical analysis of the minimum embedding dimensions required by a knowledge graph. From the view of the information entropy, a KG \mathcal{G} has a certain degree of uncertainty. An e-r query points to one or more target entities. Meanwhile, given a large-scale entity set, there will be hidden triples that are not discovered inevitably. On this basis, when using a KGE model \mathcal{M} to encode \mathcal{G} , we can train \mathcal{M} to minimize its information entropy, thereby ensure \mathcal{M} is sufficient to keep the information of the KG.

According to the above analysis, we argue that there is a sufficient condition to ensure \mathcal{M} ’s entropy $H_{\mathcal{M}}$ is smaller than the entropy of KG $H_{\mathcal{G}}$. However, the information entropy of both sides is hard to measure. Different knowledge graphs contain various entity types and topological structure, while a KGE model has its specific scoring function and randomly initialized parameters. These factors

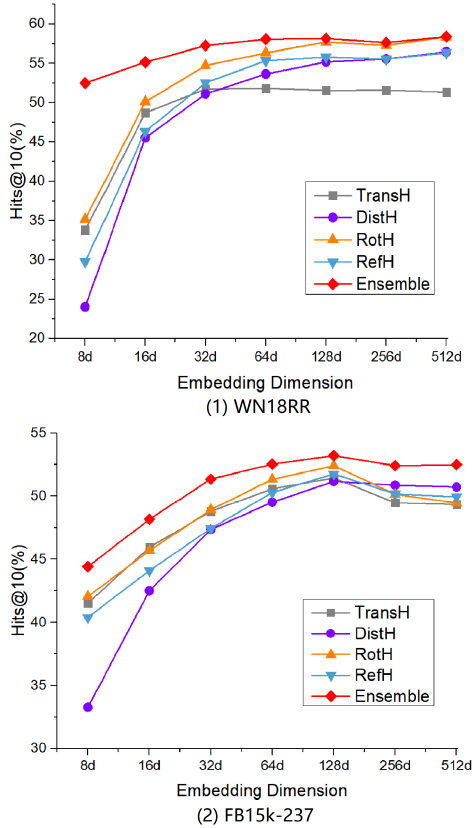


Figure 1: The changes of model performance (Hits@10) with the growth of the embedding dimensions on two different datasets. The Ensemble model adds the prediction results of the four models directly.

hinder measuring the minimum dimension of a knowledge graph by theoretical derivation. To this end, we conduct a preliminary experiment to analyze the possible value of the minimum dimension on two specific KG datasets, WN18RR [5] and FB15k237 [31]. Focusing on the low-dimensional KGE task, we employ four different hyperbolic KGE models [7] and compare their performance with different embedding dimensions. The experimental results are shown in Fig. 1.

There are two inspirations we find from this experiment: (1) On the two datasets, the prediction accuracy of all four models keeps increasing with the growth of embedding dimensions until around 64. When the embedding dimensions are more than 64, the model performance only slightly improves or starts to fluctuate. This indicates that 64-dimensional models can already achieve high performance while using much smaller parameters than those high-dimensional ones. (2) When integrating the scores of all hyperbolic models, the 64-dimensional ensemble is already better than any higher-dimensional models. Therefore, instead of training a high-dimensional model and compressing, we can utilize the low-dimensional ensemble as a lightweight source of high-accuracy knowledge.

Overall, this preliminary analysis indicates that a 64-dimensional space has an excellent capability to represent a normal-size KG

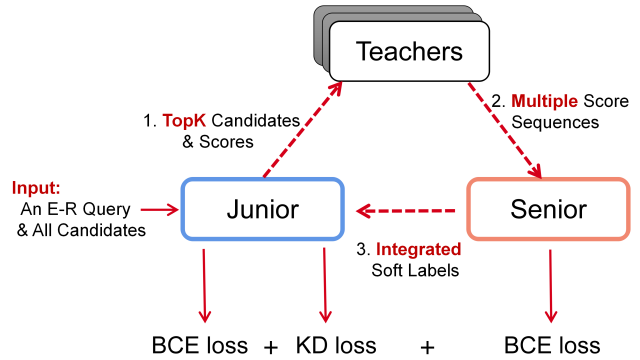


Figure 2: An illustration of the MulDE framework. ‘BCE loss’ refers to the binary cross-entropy loss, and ‘KD loss’ refers to the loss function of knowledge distillation.

dataset, which provides a guidance for the selection of teacher dimensions in our framework. Furthermore, inspired by the ensemble’s results, we focus on utilizing multiple low-dimensional models as teachers to reduce training costs while ensuring prediction accuracy.

3 METHODOLOGY

The proposed Multi-teacher Distillation Embedding (MulDE) framework utilizes multiple pre-trained low-dimensional models as teachers. Under a novel iterative distillation strategy, it integrates prediction sequences from different teachers, and supervises the training process of a low-dimensional student model. Figure 2 illustrates the architecture of the MulDE framework, including multiple teachers and two student components (Junior and Senior):

- **Multiple teachers** are regarded as the data source of prediction sequences, and have no parameter updates in the training process. We employ four different hyperbolic KGE models as teachers. The details about pre-trained teachers will be described in Section 3.1.
- **The Junior component** is the target low-dimensional KGE model. Given an e-r pair, the Junior component sends its top-K predicted entities to teachers, and gets the corresponding soft labels from the Senior component. We will detail the Junior component in Section 3.2.
- **The Senior component** acquires prediction sequences from teachers, and then generates soft labels through two mechanisms: *relation-specific mechanism* and *contrast attention mechanism*. The details of the Senior component will be discussed in Section 3.3.

As shown in Fig. 2, unlike traditional one-way guidance, the iterative distilling strategy in MulDE forms a novel *circular interaction* between students and teachers. In each iteration, the Junior component makes a preliminary prediction based on an e-r query, and selects those indistinguishable entities (top-K) to ask multiple teachers. In this way, Junior can effectively correct its prediction results and outperform the other models in the same dimension level. Meanwhile, rather than processing the fixed teacher scores, the Senior component can adjust parameters continually according

to the Junior’s feedback, and generate soft labels according to training epochs and the Junior’s performance adaptively. The learning procedure of MulDE will be detailed in Section 3.4.

3.1 Pre-trained Teacher Models

To ensure the performance of the low-dimensional teachers, we employ a group of hyperbolic KGE models proposed by Chami et al. [7], including TransH, RefH, and RotH.

Taking the RotH model as an example, it uses a d -dimensional Poincaré ball model with trainable negative curvature. Embedding vectors are first mapped into this hyperbolic space, and a relation vector is regarded as a rotation transformation of entity vectors. Then, RotH employs a hyperbolic distance function to measure the difference of transformed head vector and candidate tail vector. Similarly, TransH uses Möbius addition to imitate TransE [6] in hyperbolic space, and RefH replaces the rotation transformation with a reflection one.

Considering the effectiveness of aforementioned models using inner product transformation, such as DistMult [37] and ComplEx [32], we add another hyperbolic model named DistH into this group to get more accurate prediction sequences. The scoring functions of these models are as follows:

$$\text{TransH} : F(e_h, r, e_t) = D_{hyp}(\mathbf{e}_h \oplus \mathbf{r}, \mathbf{e}_t) \quad (2)$$

$$\text{DistH} : F(e_h, r, e_t) = D_{hyp}(\mathbf{e}_h \circ \mathbf{r}, \mathbf{e}_t) \quad (3)$$

$$\text{RotH} : F(e_h, r, e_t) = D_{hyp}(\text{Rot}(\mathbf{r})\mathbf{e}_h, \mathbf{e}_t) \quad (4)$$

$$\text{RefH} : F(e_h, r, e_t) = D_{hyp}(\text{Ref}(\mathbf{r})\mathbf{e}_h, \mathbf{e}_t) \quad (5)$$

$$D_{hyp}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \arctanh(\sqrt{c} \|\mathbf{x} \oplus \mathbf{y}\|) \quad (6)$$

where D_{hyp} is the hyperbolic distance, \oplus is Möbius addition operation, \circ is inner product operation and c is the space curvature.

We select pre-trained low-dimensional models from this group as teachers $M_T = \{M_{T_1}, M_{T_1}, \dots, M_{T_m}\}$, where m is the number of teachers. Furthermore, to verify the performance of hyperbolic space in multi-teacher knowledge distillation, we prepare a corresponding Euclidean model ‘ModelE’ for each hyperbolic model ‘ModelH’.

3.2 Junior Component

Our MulDE is a general framework, in which the Junior component can be any existing KGE model. In this paper, we focus on the low-dimensional situation, thus selecting several effective low-dimensional models as Junior, whose initialization follows their original settings, e.g., a random normal distribution. After the knowledge distillation training, Junior can make the link prediction faster than high-dimensional models and achieve similar precision.

Conventional knowledge distillation methods are mostly used for the classification problem. However, the link prediction task is a learning to rank problem, which is usually learned by distinguishing the target entity with randomly negative samples. A straightforward solution is to learn the teachers’ score distributions of the positive and negative samples. We argue that it has at least two drawbacks. First, this teaching task is too easy for multiple teachers. Every teacher will output a result close to the hard label without a noticeable difference. Second, the negative sampling can rarely hit

those indistinguishable entities, making critical knowledge owned by teachers hard to pass on to the student model. Therefore, we design a novel distilling strategy for KGE models, including two parts of supervision.

Soft Label Loss. Given an e-r query, Junior evaluates all candidate entities in the entity set. Then, it selects the top-K candidates $C_{top} = \{c_1, c_2, \dots, c_K | c_i \in E\}$ with higher scores $S_{top} = \{F_j(e, r, c_i) | c_i \in C_{top}\}$. At the beginning of an iteration, this prediction sequence $\{C_{top}, S_{top}\}$ is sent to every teacher model, and the Senior component returns soft labels L_{top} after integrating different teacher outputs. We define the soft label supervision as a Kullback-Leibler (KL) divergence, the loss function is

$$\mathcal{L}_{Soft} = \frac{1}{N} \sum \sigma'(L_{top}) \cdot \log(\sigma'(L_{top}) / \sigma'(S_{top})) \quad (7)$$

where σ' is the Softmax function, and N is the number of e-r queries.

Hard Label Loss. Meanwhile, a hard label supervision based on conventional negative sampling is employed:

$$\begin{aligned} \mathcal{L}_{Hard} = & -\frac{1}{N} \sum (L_{neg} \cdot \log(\sigma(S_{neg})) \\ & + (1 - L_{neg}) \cdot \log(1 - \sigma(S_{neg}))) \end{aligned} \quad (8)$$

where \mathcal{L}_{Hard} is a binary cross-entropy loss, σ is the Sigmoid function, S_{neg} contains scores of positive target and sampling negative ones, and L_{neg} is a one-hot label vector.

Finally, Junior loss can be formulated by weighted sum of \mathcal{L}_{Soft} and \mathcal{L}_{Hard} with a hyper-parameter γ to balance the two parts:

$$\mathcal{L}_J = \gamma \mathcal{L}_{Soft} + (1 - \gamma) \mathcal{L}_{Hard} \quad (9)$$

We argue that both supervisions are necessary. The hard label supervision gives the Junior model opportunity of learning independently, and randomly negative sampling helps handle more general queries. Meanwhile, the novel soft label supervision corrects the top-K prediction of the Junior model, encouraging it to imitate the answer of teachers for these challenging questions. In the experiments, we evaluate four different low-dimensional Junior models mentioned in Section 3.2. It is worth mentioning that as a generic framework, MulDE can also be applied to the previous KGE models, such as DistMult [37] and RotatE [29].

3.3 Senior Component

As the bridge between multiple teachers and the Junior component, the Senior component aims to integrate different score sequences from teachers and generate soft labels L_{top} suitable for the Junior component. There are two parts of input sending from teachers to the Senior component, the top-K Junior scores S_{top} and multiple score sequences $\{S_{T_1}, S_{T_1}, \dots, S_{T_m}\}$ (m is the number of teachers) from the teacher models. All of the above sequences have length K corresponding to K candidate entities in C_{top} . In the Senior component, we utilize two mechanisms for knowledge integration:

Relation-specific Scaling Mechanism. According to previous research [1], different KGE models show performance advantages in different relations. To improve the integrated soft labels, every teacher model should contribute to the relations that the model is good at. To this end, we define a trainable scaling matrix $W_{Rel} \in \mathcal{R}^{N_r \times m}$, and assign an adaptive scaling value (ranges from 0 to 1)

for each teacher sequence. The i -th scaled sequence S'_{Ti} is computed as:

$$S'_{Ti} = S_{Ti} \times \sigma(W_{Rel}[r', i]) \quad (10)$$

where r' refers to the index of the relation r in an e-r query. Note that all scores in a sequence are adjusted in proportion, while their relative ranks have no changes. Attempts have been made to adjust each triple score individually, but the Senior loss is hard to converge and fluctuates sharply.

Contrast Attention Mechanism. As Junior is randomly initialized, a preliminary division of the whole entity set is needed in the early model training. At this time, the distribution of its scores would be significantly different from that of the trained teacher. A disparate soft label would block the training process. Therefore, at the early period of model training, Senior is encouraged to generate soft labels having a similar distribution with Junior's scores.

Specifically, we contrast the Junior sequence with each teacher sequence and evaluate their similarity. An attention mechanism is designed to make the integrated sequence more similar to the Junior sequence. The calculation process of soft labels L_{top} is as follows:

$$p_i = \frac{1}{K} \sum \sigma'(S_{Ti}) \cdot \log(\sigma'(S_{Ti})/\sigma'(S_{top})) \quad (11)$$

$$L_{top} = \sum_i (S'_{Ti} \times \sigma'(-p_i/\tau) \times m) \quad (12)$$

where p_i is the KL divergence, a higher p_i means more difference between S_{Ti} and S_{top} . To ensure that the contrast attention only works at the early tens of training epochs, we set a temperature parameter $\tau = \exp(\lfloor n/5 \rfloor)$. With the increase of the training epochs n , τ increases exponentially and p_i of different teachers tend to be equal. So that, after the first few epochs, the soft labels L_{top} are only adjusted by relation-specific scaling mechanism to achieve higher performance.

In order to train the parameters in W_{Rel} , a binary cross-entropy loss is utilized to evaluate the performance of scaled scores S'_{Ti} in the Senior component as follows:

$$\begin{aligned} \mathcal{L}_S = & -\frac{1}{N} \sum (L_{seq} \cdot \log(\sigma(\sum_i S'_{Ti}))) \\ & + (1 - L_{seq}) \cdot \log(1 - \sigma(\sum_i S'_{Ti})) \end{aligned} \quad (13)$$

where L_{seq} is a one-hot vector in which the position of the target entity of the e-r query is 1 and the rest is 0. Besides, if the target entity is not in the candidate set, L_{seq} is the zero vector.

3.4 Learning Algorithm

There are three roles in the proposed MulDE framework, namely *Teacher* (pre-trained hyperbolic models), *Senior Student* (integration mechanisms), and *Junior Student* (the target low-dimensional model). As the teacher models have no parameter fine-tuning, we use a combined loss function to train the two student components and minimize the loss by exploiting the Adam optimizer [14]. The complete loss function \mathcal{L}_{final} is as follows:

$$\mathcal{L}_{final} = \mathcal{L}_S + \mathcal{L}_J + \lambda \|\Theta\|_2^2 \quad (14)$$

where $\lambda \|\Theta\|_2^2$ is the parameter regularization term.

Algorithm 1 The Learning Procedure of MulDE

Require: Q : the training KG triples;
 E : the total entity set of KG;
Ensure: Θ_J : the parameters of Junior;
 Θ_S : the parameters of Senior;

// Pre-training Phase
1: **for** $i = 1, 2, \dots, m$ **do**
2: Train the i^{th} teacher model M_{Ti} with Q ;
3: **end for**

// Training Phase
4: Initialize parameters (Θ_J, Θ_S) ;
5: **for** each training epoch **do**
6: **for** each $(e_{in}, r, e_{miss}) \in Q$ **do**
7: $S_r \leftarrow \mathbf{JuniorScoreFunction}(e_{in}, r, E, \Theta_J)$;
8: $\{C_{top}, S_{top}\} \leftarrow \mathbf{GatherTopKCandidates}(S_r, E)$;
9: $\{L_{neg}, S_{neg}\} \leftarrow \mathbf{RandomNegSamples}(S_r, E)$;
10: **for** $i = 1, 2, \dots, m$ **do**
11: $S_{Ti} \leftarrow \mathbf{TeacherScoreFunction}(M_{Ti}, e_{in}, r, C_{top})$;
12: **end for**
13: $\{S'_{Ti}\} \leftarrow \mathbf{RelationScaling}(\{S_{Ti}\}, \Theta_S)$;
14: $L_{top} \leftarrow \mathbf{ConstrastAttention}(S_{top}, \{S'_{Ti}\})$;
15: **end for**
16: Compute the Senior loss with L_{top} ;
17: Compute the Junior loss with S_{top} and S_{neg} ;
18: Update parameters (Θ_J, Θ_S) by gradient descent;
19: **end for**

The learning algorithm of MulDE is presented in Algorithm 1. We emphasize the preciseness of this learning procedure. It should be noted that in the pre-training and training phases, both teachers and the Junior component cannot access the triples in the validation and test sets, including the negative sampling process.

Computing Efficiency. We carefully consider the computing efficiency of MulDE in low-dimensional condition. As the parameters in teacher models are fixed and Junior is any KGE model, the only additional burden is the Senior component. Senior's space complexity can be ignored, as the scaling mechanism only uses $N_r \times m$ parameters and the contrast attention mechanism is parameter-free. For time complexity, Senior only deals with score sequences instead of d-dimensional vectors, thus its calculation amount is less than 1/d of the time needed by Junior. Inevitably, the training cost of iterative distillation strategy in one epoch is more than a single KGE model, due to the calculations of teachers and the selection of topK candidates. However, we find that the convergence speed of MulDE is relatively faster, which would obviously reduce the total training times. Therefore, it is of high-efficiency to utilize MulDE to enhance existing low-dimensional KGE models.

Table 1: Statistics of the datasets.

Dataset	N_r	N_e	#Train	#Valid	#Test
FB15k237	237	14,541	272,115	17,535	20,466
WN18RR	11	40,943	86,845	3,034	3,134

Table 2: Link prediction results on the two datasets. The best score of high-dimensional models underlined and the best score of 32-dimensional models in **Bold. The symbol ‘*’ means this score outperforms that of all previous low-dimensional models.**

Type	Dim	Methods	FB15K237			WN18RR		
			MRR	Hits@10	Hits@1	MRR	Hits@10	Hits@1
High-dimensional Models	200d-500d	TransE	0.256	0.456	0.152	0.207	0.476	0.012
		DistMult	0.286	0.445	0.202	0.412	0.484	0.372
		ComplEx	0.283	0.447	0.202	0.431	0.513	0.395
		ConvE	0.316	0.501	0.237	0.430	0.520	0.400
		RotatE	0.338	0.533	0.241	0.476	0.571	0.428
		TuckER	<u>0.358</u>	0.544	<u>0.266</u>	0.470	0.526	0.443
		QuatE	0.348	<u>0.550</u>	<u>0.248</u>	0.488	0.582	0.438
		RefH	0.346	0.536	0.252	0.461	0.568	0.404
		RotH	0.344	0.535	0.246	<u>0.496</u>	<u>0.586</u>	<u>0.449</u>
		AttH	0.348	0.540	0.252	0.486	0.573	0.443
Low-dimensional Models	32d	RotatE	0.290	0.458	0.208	0.387	0.491	0.330
		TuckER	0.306	0.475	0.223	0.428	0.474	0.401
		RefH	0.312	0.489	0.224	0.447	0.518	0.408
		RotH	0.314	0.497	0.223	0.472	0.553	0.428
		AttH	0.324	0.501	0.236	0.466	0.551	0.419
Our MulDE Models	32d	TransH	0.308	0.488	0.217	0.231	0.518	0.081
		MulDE-TransH	0.328*	0.511*	0.236*	0.267	0.540	0.094
		DistH	0.293	0.474	0.202	0.439	0.511	0.399
		MulDE-DistH	0.326*	0.509*	0.235	0.460	0.545	0.417
		RefH	0.302	0.474	0.215	0.453	0.526	0.414
		MulDE-RefH	0.325*	0.508*	0.233	0.479*	0.569*	0.434*
		RotH	0.310	0.489	0.221	0.463	0.547	0.416
MulDE-RotH	0.328*	0.515*	0.237*	0.481*	0.574*	0.433*		

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. Our experimental studies are conducted on two commonly used datasets. WN18RR [5] is a subset of the English lexical database WordNet [18]. FB15k237 [31] is extracted from Freebase including knowledge facts about movies, actors, awards, and sports. It is created by removing inverse relations, because many test triples can be obtained simply by inverting triples in the training set. The statistics of the datasets are given in Table 1. ‘Train’, ‘Valid’, and ‘Test’ refer to the amount of triples in the training, validation, and test sets.

Baselines. We implement MulDE by employing four hyperbolic KGE models as student or teacher models, including TransH, DistH, RefH, and RotH. Although AttH [7] achieves better performance in one or two metrics, it is a combination of RefH and RotH. To verify the influence of the hyperbolic space, we also pre-train a corresponding Euclidean model ‘ModelE’ for each hyperbolic model ‘ModelH’.

The compared KGE models are in two classes: (1) High dimensional models, including the state-of-the-art KGE methods in Euclidean space or Hyperbolic space. All of them adopt embedding vectors in more than 200 dimensions. (2) Low-dimensional models, including RotH, RefH and AttH proposed by Chami et al. in [7]. Benefiting from hyperbolic vector space, they have shown obvious advantages in the low-dimensional condition.

Implementation Details. All experiments are performed on Intel Core i7-7700K CPU @ 4.20GHz and NVIDIA GeForce GTX1080 Ti GPU, and are implemented in Python using the PyTorch deep learning framework [22].

We select the hyper-parameters of our model via grid search according to the metrics on the validation set. For the teacher models, we pre-train different teachers with the embedding dimensions among {64, 128, 256, 512}. According to the preliminary analysis, we set the embedding dimensions as 64 in the main experiments. We select the learning rate among {0.0005, 0.001, 0.005}, and the number of negative samples among {8, 50, 255}.

For the MulDE framework, we empirically select the Junior’s embedding dimensions among {8, 16, 32, 64}, the length K of prediction sequences among {100, 300, 500}, the learning rate among {0.0005, 0.001, 0.005}, and the balance hyper-parameter γ among {0.01, 0.1, 0.5}.

Evaluation Metrics. For the link prediction experiments, we adopt two evaluation metrics: (1) MRR (Mean Reciprocal Rank), the average inverse rank of the test triples, and (2) Hits@N, the proportion of correct entities ranked in top N. Higher MRR and Hits@N mean better performance. Following the previous works, we process the output sequence in the Filter mode. It is worth mentioning that, for pre-trained models, we only remove those entities appearing in the training dataset.

4.2 Link Prediction Task

We first evaluate our model in the 32-dimensional vector space, which is the same as the low-dimensional setting of Chami et al. [7]. MulDE-modelH represents the model whose Junior model is a 32-dimensional hyperbolic model, and its four teachers, TransH, DistH, RotH and RefH, are pre-trained in the 64-dimensional space. We compare distilled Junior models with its original results, as well as with the state-of-the-art models in both low and high dimensions. The experimental results are shown in Table 2.

From the table, we can have the following observations. The four different Junior models trained by MulDE significantly outperform their original performance on the both datasets. The MRR and Hits@10 of all four models have an average 5% increase. Especially, the Hits@10 of RotH improves from 0.547 to 0.574 on WN18RR, and the Hits@1 of DistH improves from 0.202 to 0.235 on FB15k-237. The results illustrate the effectiveness of knowledge distillation for low-dimensional embedding.

Compared with previous low-dimensional models, MulDE-RotH achieves the state-of-the-art results in all metrics on the two datasets. Although the AttH model combines both RotH and RefH, it is weaker than our next-best model MulDE-RefH in 5 metrics. Besides, the TransH model only utilizes a simple scoring function, but exceeds all previous low-dimensional models on the FB15k-237 dataset after the knowledge distillation training.

Compared with the high-dimensional models, the 32d MulDE-RotH exceeds multiple 200-dimensional models, including TransE, DistMult, ComplEx, and ConvE. Although its performance is lower than some of the latest RotatE, TuckER and QuatE models, MulDE-RotH has shown strong competitiveness in some metrics with much less parameters. Significantly, the Hits@10 of MulDE-RotH on WN18RR outperforms most state-of-the-art models, and its MRR and Hits@1 metrics are very close to those of the high-dimensional RotH model on the two datasets.

Overall, we can conclude that the MulDE framework can successfully improve the low-dimensional hyperbolic models. The performance of those distilled models is even comparable to some high-dimensional models.

4.3 Comparison with Different Dimensions

We further evaluate our MulDE framework with different dimensions. For the student embedding dimensions, we still focus on low-dimensional models with dimensions lower than 100, and compare distilled models with the original ones. For the teacher embedding dimensions, we select multiple dimension sizes over 64, including 128, 256, and 512. Then, we measure the performance gap between the 32-dimensional student and different high-dimensional teachers. The experimental results are shown in Table 3.

The performance of different low-dimensional student models are shown in Table 3 (a). The results show that the accuracy of the original models reduces when the embedding dimensions decrease. Although the distilled Junior models outperform their original models, the 8d and 16d models achieve relatively poor results. In contrast, from the ‘growth’ metrics, MulDE contributes more improvements for lower-dimensional models. The MRR and Hits@1 of 8d RotH increase more than 90% and 150%, which is much more significant

Table 3: The improvements of MulDE-RotH on WN18RR with different teacher and student dimensions. Bold indicates better RotH results with same dimension, while Growth is the increase rate between the two models in the same region.

Methods	Dim	MRR	Hits@10	Hits@1
Teachers	64d	0.487	0.581	0.440
Origin	8d	0.210	0.352	0.140
Junior	8d	0.399	0.483	0.350
<i>Growth</i>		90.05%	37.41%	150.32%
Origin	16d	0.412	0.501	0.358
Junior	16d	0.464	0.547	0.419
<i>Growth</i>		12.60%	9.07%	17.22%
Origin	32d	0.463	0.547	0.416
Junior	32d	0.481	0.574	0.433
<i>Growth</i>		3.91%	4.91%	4.14%
Origin	64d	0.477	0.564	0.429
Junior	64d	0.482	0.579	0.430
<i>Growth</i>		1.05%	2.77%	0.21%

(a) Various Student Dimensions

Methods	Dim	MRR	Hits@10	Hits@1
Origin	32d	0.463	0.547	0.416
Teachers	64d	0.487	0.581	0.440
Junior	32d	0.481	0.574	0.433
<i>Growth</i>		-1.23%	-1.20%	-1.59%
Teachers	128d	0.488	0.582	0.437
Junior	32d	0.480	0.572	0.430
<i>Growth</i>		-1.64%	-1.71%	-1.60%
Teachers	256d	0.483	0.578	0.434
Junior	32d	0.473	0.569	0.424
<i>Growth</i>		-2.07%	-1.60%	-2.30%
Teachers	512d	0.479	0.584	0.423
Junior	32d	0.469	0.574	0.412
<i>Growth</i>		-2.09%	-1.71%	-2.60%

(b) Various Teacher Dimensions

than that of 64d RotH. Furthermore, the performance of 64d distilled models are very close to the 32-dimensional ones. As a result, to save storage space in applications, 32d models can be used.

The results in Table 3 (b) support our preliminary analysis to some extent. At first, the performance of the teacher ensemble is similar when its dimension exceeds 64. The Hits@10 of 256d teachers is even worse than that of 64d. It proves that when encoding the same KG, the capacity of the model tends to be stable when the embedding dimensions exceed the lower bound (e.g., near 64 for WN18RR). Besides, with the increase of teacher dimensions, MRR and Hits@1 of the Junior model decrease, indicating that the teacher with over-high dimensions would be harder to transfer knowledge to a low-dimensional student. The experimental results motivate us to apply 64-dimensional teacher models, which achieve higher performance and save more pre-training costs.

Table 4: The results of ablation experiments on WN18RR and FB15k237. The student model of all variants is RotH. TopK refers to the iterative distilling strategy replacing random candidates as TopK candidates, while RS and CA refer to two mechanisms in the Senior component. KD512d and KD64d are two single-teacher models with only 512d and 64d RotH model as the teacher, respectively.

Methods	FB15K237			WN18RR		
	MRR	Hits@10	Hits@1	MRR	Hits@10	Hits@1
MulDE	0.328	0.515	0.237	0.481	0.574	0.433
MulDE w/o TopK	0.321	0.502	0.231	0.456	0.544	0.418
MulDE w/o RS	0.325	0.511	0.234	0.476	0.570	0.430
MulDE w/o CA	0.322	0.509	0.233	0.481	0.571	0.422
MulDE w/o Senior	0.323	0.501	0.232	0.459	0.532	0.419
KD512d w/ TopK	0.322	0.502	0.232	0.469	0.564	0.421
KD64d w/ TopK	0.324	0.506	0.229	0.467	0.553	0.425
KD64d	0.321	0.498	0.230	0.452	0.540	0.414

4.4 Ablation Studies about Distillation Strategy

We further make a series of ablation experiments to evaluate different modules in MulDE. There are three main improvements: an iterative distilling strategy (TopK), a relation-specific scaling mechanism (RS), and a contrast attention mechanism (CA). Therefore, we test the performance of MulDE with respect to the three modules. Besides that, we also compare MulDE with single-teacher distillation in which the teacher is the same KGE model as the student with 64 or 512 dimensions. To make it harder still, we apply the iterative distilling strategy in these single-teacher models instead of random candidate labels. The experimental results are shown in Table 4.

At first, for the iterative distilling strategy, it is clear that the performance of MulDE dramatically decreases when removing this module. Especially, the Hits@10 on WN18RR reduces from 0.574 to 0.544 (-5.3%). This proves the necessity of using top-K labels when applying knowledge distillation in link prediction tasks. In contrast, although two mechanisms are helpful, the contributions of RS and CA are relatively small. Further improvement of the Senior component will be one of our future work.

Although single RS or CA has limited performance improvement, the existence of Senior is necessary for the interactive distillation strategy. As the results show that, when eliminating the Senior component, the Hits@10 on two datasets decreases to 0.501 and 0.532 respectively. We argue that it benefits from Senior integrating teacher scores adaptively. As both RS and CA provide a weighted operation to different teachers, eliminating Senior would make teacher scores and soft labels fixed in the training process. Only Junior can reduce the KL divergence in KD loss, and get stuck in a locally optimal solution easier. Oppositely, Senior can provide variable soft labels and adjust KL divergence in another direction.

The single-teacher knowledge distillation can improve the original RotH model, but its performance is worse than MulDE. The iterative distilling strategy also positively affects the single-teacher framework, which increases Hits@10 on WN18RR by around 2%. A higher-dimensional teacher component (512d) can enhance the performance, but its training cost is unavoidable higher than MulDE.

Overall, the experimental results indicate the effectiveness of the three major modules in MulDE. Compared with the single-teacher strategy, our framework shows apparent improvements.

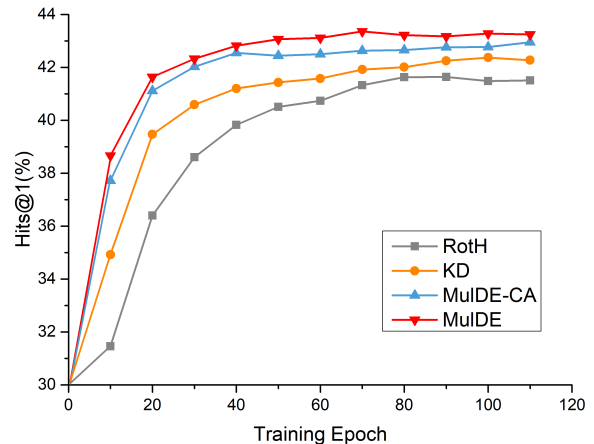


Figure 3: The Hits@1 of 32-dimensional RotH as training proceeds on WN18RR. RotH is the original training mode, KD is the single-teacher knowledge distillation process, MulDE and MulDE-CA are our models with and without the contrast attention mechanism.

5 DISCUSSIONS

In this section, we particularly discuss several arresting questions around MulDE.

Q1: To what extent distillation strategies accelerate student training speed?

Fig. 3 shows the convergence of 32-dimensional RotH with different training processes. As expected, the student model trains faster under the supervision of a teacher. We can observe that MulDE converges faster than original RotH, and achieves higher accuracy. In contrast, the single-teacher KD model is much slower and continuously increases before 100 epochs. We also analyze the convergence of MulDE without the contrast attention mechanism, i.e., MulDE-CA. Benefiting from contrast attention, the Hits@1 of MulDE in the early period (epoch=10) is around 3% higher than MulDE-CA, while it also leads to a higher final accuracy from 0.422 to 0.433. This indicates that the contrast attention can reduce the gap between student scores and teacher scores in the early period.

Q2: Whether the hyperbolic space contributes to the result?

We evaluate the MulDE framework by employing hyperbolic KGE models based on their performance in low dimensions. To answer the question, we analyze the contributions of hyperbolic models when they are teachers or students, and also verify the effectiveness of MulDE in Euclidean space. The results are shown in Fig. 4 (a). Here ‘teaH’ and ‘stuH’ represent using hyperbolic models, while ‘teaE’ and ‘stuE’ indicate using corresponding Euclidean models. The teaH_stuH model is equal to MulDE-RotH, and the other variants assign different space types to teachers and students. Among the four models, teaE_stuE is weaker than the other three, which indicates the advantage of hyperbolic space. Even so, the 0.466 Hits@1 is still equal to that of the previous AttH. Considering the middle two models, we can conjecture that using hyperbolic space in the teacher components are more effective than using a hyperbolic student model.

Q3: How about the contribution of different teacher choices on results?

The complete MulDE framework utilizes all four hyperbolic models as teachers, because the whole ensemble can obtain better accuracy than the other combinations. We further analyze the contribution of every single teacher model in the ensemble. As shown in Fig. 4 (b), the performance of MulDE outperforms the other four teacher combinations. In terms of combining two teachers, only using TransH and DistH is obviously poorer than using RotH and RefH, which indicates the importance of the latter models. Comparing two models using three teachers, we find that RotH contributes more than RefH. It is reasonable because the accuracy of the original RotH is already higher than RefH. We also trial other combinations of the four models (e.g., T+R1, D+R1, T+R2, D+R2), and the experimental results prove the contribution of every teacher model in MulDE. One of our future work will be discovering a better teacher combination by adding other different models.

Q4: Which parts of prediction results obtain improvement after enhancing by MulDE?

The failed prediction of KGE models may be caused by various reasons, e.g., KG in-completion, similar entities, and data sparseness. As MulDE can effectively improve Junior’s accuracy, we deeply explore those improved prediction samples in evaluating phase. Tabel 5 shows two groups of prediction results about two triples on the FB15k237 test set.

In the first example, when predicting the genre of this science-fiction show, the original RotH model fails to rank the target ‘Crime Fiction’ in the top five candidates, while both the teacher ensemble and our MulDE-RotH rank the target at second. It indicates that MulDE-RotH outperforms RotH, benefiting from the supervision of teachers. Although MulDE-RotH still predicts a wrong tail entity, ‘Fantasy’ is reasonable in the human view. A similar situation is shown in the second example, in which MulDE-RotH successfully finds the correct answer ‘Telecommunications’ after training by MulDE. It should be noted that MulDE-RotH does not directly reproduce teachers, as its results are different from teachers’.

In case studies on the FB15237 test set, the teacher ensemble correctly predicts 1,907 more samples than 32-dimensional original RotH, while the number of improved samples for the 32-dimensional

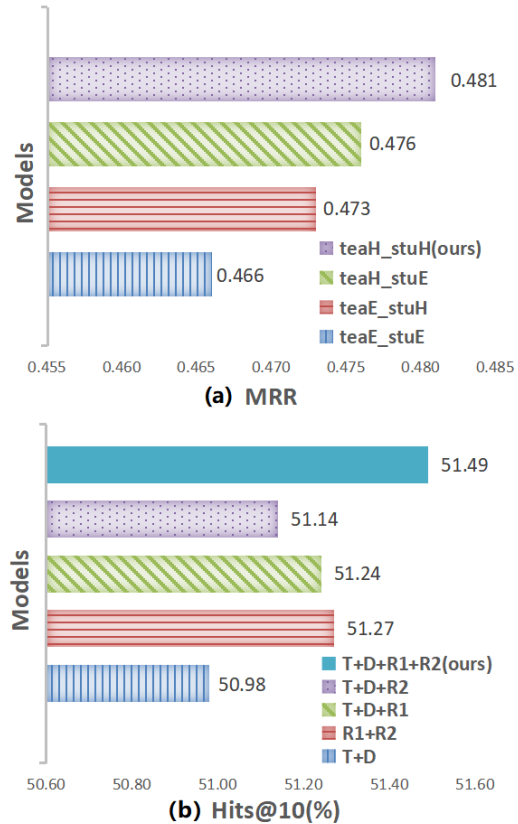


Figure 4: Performance of 32-dimensional student with different variants. (a) The MRR on WN18RR with different settings of vector space. (b) The Hits@10 on FB15k237 with different teacher combinations. We use T, D, R1, R2 refers to TransH, DistH, RotH and RefH.

MulDE-RotH is 1,808 and 75.6% of samples are same with teachers’. The results indicate that a staggering proportion of improved samples of MulDE-RotH are benefited from the supervision of teachers.

6 RELATED WORK

In this section, we discuss recent research advance in the KGE domain, and introduce relevant Knowledge Distillation research.

6.1 Knowledge Graph Embeddings

Predicting new triples using KGE methods has been an active research topic over the past few years [24]. Benefiting from the simple calculation and good performance, some early methods such as TransE [6], DistMult [37], ComplEx [32] have been widely used in various AI tasks. With the rise of deep learning, several CNN-based methods have been proposed, such as ConvE [10] and ConvKB [20]. These methods achieve good performance in link prediction tasks, but heavily depend on more parameters [26, 33]. Recently, there are several non-neural methods proposed. RotatE [29], inspired by Euler’s identity, can infer various relation patterns with a new rotation-based scoring function. Balazevic et al. [2] propose a linear

Table 5: The prediction results of different models about two triples. The correct candidate is in bold.

Query	Methods	Top 5 Candidates
(The X-Files, /tv/tv_program/genre)	RotH MulDE-RotH Teachers	Psychological Thriller, Fantasy, Adventure Film, Comedy Drama, Period Piece Fantasy, Crime Fiction , Horror, Adventure Film, Comedy Drama Fantasy, Crime Fiction , Psychological Thriller, Horror, Adventure Film
(Comcast, /business/business _operation/industry)	RotH MulDE-RotH Teachers	Software, Computer hardware, Retail, Manufacturing, Telecommunications Telecommunications , Retail, Software, Video Game, Computer Hardware Telecommunications , Software, Computer Hardware, Manufacturing, Retail

model based on Tucker decomposition of the binary tensor representation of knowledge graph triples. QuatE [40] and OTE [30] further improve the RotatE method. The former introduces more expressive hypercomplex-valued representations in the quaternion space, while the latter (i.e., OTE) extends the RotatE method from a 2-D complex domain to a high dimensional space with orthogonal transforms to model relations.

Since a single KGE model faces a contradiction between complexity and performance, a straightforward solution is to take an ensemble of multiple KGE methods to predict the same target. Krompass et al. [15] designed a simple ensemble consisting of multiple KGE methods, in which the final prediction results rely on an average score of all sub-modules. Wang et al. [36] proposed a relation-level ensemble that combines multiple individual models to pick the best model for each relation.

6.2 KGE Model Compression

Research related to the KGE model compression is relatively recent and new, featuring two representative approaches. One possible solution is to compress pre-trained high-dimensional models. Sachan [25] utilized Embedding Compression methods to convert high-dimensional continuous vectors into discrete codes. Although generating a compressed model retaining high-accuracy knowledge, a pre-trained high-dimensional model is necessary. Furthermore, the compressed model with discrete vectors cannot continue the training task when the KG is modified. Another solution introduces new theories to improve low-dimensional KGE models directly. For example, Chami et al. [7] introduced the hyperbolic embedding space with trainable curvature, and proposed a class of hyperbolic KGE models, which outperforms previous Euclidean-based methods in low-dimension. However, the limited number of parameters inevitably declines the model performance, and the low-dimensional models cannot utilize the high-accuracy knowledge from high-dimensional ones.

6.3 Knowledge Distillation

Knowledge Distillation (KD) aims to transfer knowledge from one machine learning model (i.e., the teacher) to another model (i.e., the student) [21, 41]. Hinton et al. [13] introduced the first KD framework, which applies the classification probabilities of a trained model as soft labels and defines a parameter called “Temperature” (T) to control the soft degree of those labels. Specifically, when $T \rightarrow 0$, the soft labels become one-hot vectors, i.e., the hard labels. With the increase of T , the labels become softer. When $T \rightarrow \infty$, all classes share the same probability [9, 17]. Inspired by this, several KD-based approaches are proposed in different research domains.

Furlanello et al. [11] proposed Born-Again Networks in which the student is parameterized identically to their teachers. Yang et al. [38] added an additional loss term to facilitate a few secondary classes to emerge and complement to the primary class. Li et al. [16] proposed a method to distill human knowledge from a teacher model to enhance a pedestrian attribute recognition task. To the best of our knowledge, our work is the first to apply the KD technologies in the link prediction of KGs.

7 CONCLUSION

Recent knowledge graph embedding (KGE) models tend to apply high-dimensional embedding vectors to improve their performance. These models can be hardly applied in practical knowledge graphs (KGs) due to large training costs and storage space. In this paper, we propose MulDE, a novel multi-teacher knowledge distillation framework for knowledge graph embeddings. By integrating multiple hyperbolic KGE models as teachers, we present a novel iterative distillation strategy to extract high-accuracy knowledge and train low-dimensional students adaptively. The experimental results show that the low-dimensional models distilled by MulDE outperform the state-of-the-art models on two widely used datasets. Compared with general single-teacher knowledge distillation methods, MulDE can accelerate student training speed.

These positive results encourage us to explore the following further research activities in the future:

- We will further research the relationship between embedding dimensions and the KG scale, and explore the influence of different relation transformations.
- To take advantages of multiple teachers, we will further improve the knowledge integration in the Senior component, and achieve higher-accuracy soft labels in the low dimensional models.
- Regarding the choice of multiple teachers, we will further analyze the effectiveness of the ensemble. In particular, we will investigate new teacher combinations by considering emerging relation transformations.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation in China (Grant: 61672128) and the Fundamental Research Fund for Central University (Grant: DUT20TD107). Quan Z. Sheng has been partially supported by Australian Research Council (ARC) Future Fellowship Grant FT140101247, and Discovery Project Grant DP200102298.

REFERENCES

- [1] Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic Re-evaluation of Knowledge Graph Completion Methods: An Experimental Study. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14–19, 2020*. 1995–2010.
- [2] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, EMNLP-IJCNLP, 2019*. 5184–5193.
- [3] Caleb Belth, Xinyi Zheng, Jilles Vreeken, and Danai Koutra. 2020. What is Normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020*. ACM / IW3C2, 1115–1126.
- [4] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - A crystallization point for the Web of Data. *J. Web Semant.* 7, 3 (2009), 154–165.
- [5] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94 (2014), 233–259.
- [6] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 2013)*. 2787–2795.
- [7] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*. 6901–6914.
- [8] Feihu Che, Dawei Zhang, Jianhua Tao, Mingyue Niu, and Bocheng Zhao. 2020. ParamE: Regarding Neural Network Parameters as Relation Embeddings for Knowledge Graph Completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7–12*. 2774–2781.
- [9] Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling Task Knowledge from How-To Communities. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7, 2017*. ACM, 805–814. <https://doi.org/10.1145/3038912.3052715>
- [10] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*. 1811–1818.
- [11] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-Again Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. 1602–1611.
- [12] Gaole He, Junyi Li, Wayne Xin Zhao, Peiju Liu, and Ji-Rong Wen. 2020. Mining Implicit Entity Preference from User-Item Interaction Data for Knowledge Graph Completion via Adversarial Learning. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24*. 740–751.
- [13] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015).
- [14] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of International Conference on Learning Representations (ICLR 2015)*.
- [15] Denis Krompass and Volker Tresp. 2015. Ensemble Solutions for Link-Prediction in Knowledge Graphs. In *PKDD/ECML 2nd Workshop on Linked Data for Knowledge Discovery*.
- [16] Qiaozhe Li, Xin Zhao, Ran He, and Kaiqi Huang. 2019. Pedestrian Attribute Recognition by Joint Visual-semantic Reasoning and Knowledge Distillation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. 833–839.
- [17] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. 2019. Knowledge Distillation via Instance Relationship Graph. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019*. Computer Vision Foundation / IEEE, 7096–7104. <https://doi.org/10.1109/CVPR.2019.00726>
- [18] George A. Miller. 1995. WORDNET: A Lexical Database for English. *Commun. ACM* 38 (1995), 39–41.
- [19] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghahemzadeh. 2020. Improved Knowledge Distillation via Teacher Assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. 5191–5198.
- [20] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2017. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2017 Conference of the North American Chapter of the Association for Computational Linguistics*, Vol. 2. 327–333.
- [21] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational Knowledge Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019*. Computer Vision Foundation / IEEE, 3967–3976. <https://doi.org/10.1109/CVPR.2019.00409>
- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035.
- [23] Heiko Paulheim. 2018. How much is a Triple? Estimating the Cost of Knowledge Graph Creation. In *Proceedings of the 17th International Semantic Web Conference (ISWC 2018)*, Vol. 2180.
- [24] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.
- [25] Mrinmaya Sachan. 2020. Knowledge Graph Embedding Compression. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*. 2681–2691.
- [26] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Ivan Titov, Max Welling, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. *Proceedings of Extended Semantic Web Conference (2018)*, 593–607.
- [27] Yuxuan Shi, Gong Cheng, and Evgeny Kharlamov. 2020. Keyword Search over Knowledge Graphs via Static and Dynamic Hub Labelings. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24, 2020*. ACM / IW3C2, 235–245. <https://doi.org/10.1145/3366423.3380110>
- [28] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8–12, 2007*. 697–706.
- [29] Zhiqing Sun, Zhihong Deng, Jianyun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019*.
- [30] Yun Tang, Jing Huang, Guangtao Wang, Xiaodong He, and Bowen Zhou. 2020. Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*. 2713–2722.
- [31] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. 57–66.
- [32] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33th International Conference on Machine Learning*. 2071–2080.
- [33] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net. https://openreview.net/forum?id=ByIA_C4Pr
- [34] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017), 2724–2743.
- [35] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24*. 99–109.
- [36] Yanjie Wang, Rainer Gemulla, and Hui Li. 2017. On Multi-Relational Link Prediction with Bilinear Models. In *Proceedings of Thirty-First AAAI Conference on Artificial Intelligence*. 4227–4234.
- [37] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015*.
- [38] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L. Yuille. 2019. Training Deep Neural Networks in Generations: A More Tolerant Teacher Educates Better Students. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence, 2019*. 5628–5635.
- [39] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen. 2020. Relation Adversarial Network for Low Resource Knowledge Graph Completion. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20–24*. 1–12.
- [40] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP 2019)* (2019).
- [41] Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding Knowledge Distillation in Non-autoregressive Machine Translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net. <https://openreview.net/forum?id=ByGFVAEKDH>
- [42] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. 1943–1949.