Swift and Sure: Hardness-aware Contrastive Learning for Low-dimensional Knowledge Graph Embeddings

Kai Wang[1,2,3]*, Yu Liu[1,2], Quan Z. Sheng[3]
[1]School of Software, Dalian University of Technology, Dalian, Liaoning, China
[2]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, Liaoning, China
[3]Intelligent Computing Laboratory, School of Computing, Macquarie University, Sydney, NSW, Australia
kai_wang@mail.dlut.edu.cn,yuliu@dlut.edu.cn,michael.sheng@mq.edu.au

ABSTRACT

Knowledge graph embedding (KGE) has shown great potential in automatic knowledge graph (KG) completion and knowledge-driven tasks. However, recent KGE models suffer from high training cost and large storage space, thus limiting their practicality in real-world applications. To address this challenge, based on the latest findings in the field of Contrastive Learning, we propose a novel KGE training framework called Hardness-aware Low-dimensional Embedding (HaLE). Instead of the traditional Negative Sampling, we design a new loss function based on query sampling that can balance two important training targets, *Alignment* and *Uniformity*. Furthermore, we analyze the hardness-aware ability of recent low-dimensional hyperbolic models and propose a lightweight hardness-aware activation mechanism, which can help the KGE models focus on hard instances and speed up convergence. The experimental results show that in the limited training time, HaLE can effectively improve the performance and training speed of KGE models on five commonly-used datasets. After training just a few minutes, the HaLE-trained models are competitive compared to the state-of-the-art models in both low- and high-dimensional conditions.

CCS CONCEPTS

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

KEYWORDS

Knowledge Graph Embedding, Contrastive Learning, Link Prediction, Knowledge Graph

ACM Reference Format:
Kai Wang, Yu Liu, and Quan Z. Sheng. 2022. Swift and Sure: Hardness-aware Contrastive Learning for Low-dimensional Knowledge Graph Embeddings. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485447.3511927

*Corresponding author.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3511927

1 INTRODUCTION

Knowledge Graph (KG) has shown great potential for recording semantic data from the Web as factual triples in the form of (head entity, relation, tail entity). Knowledge Graph Embedding (KGE) can further represent entities and relations in the continuous vector space, and has been widely utilized in automatic KG completion and knowledge-driven tasks, such as information retrieval, semantic matching, and question answering [14, 15, 32, 34]. However, recent KGE models usually utilize complicated computational structures and high-dimensional vectors up to 500 or even 1,000 dimensions [7, 11, 22]. Training such high-dimensional models demands prohibitive training costs and storage space, yet only achieving slight performance increase. Meanwhile, large-scale KGs in real-world industrial applications, usually contain millions or billions of entities and need to be updated constantly based on real-time business data [18, 23]. Consequently, current KGE models mostly still stay in laboratory environments and remain difficult to be deployed in practical applications [16, 29].

To reduce the training costs, a promising way is to design new loss functions. As the Negative Sampling loss has been indicated time-consuming and unstable, Sun et al. [22] proposed a Self-adversarial Negative Sampling loss, which uses the Softmax-normalized triple score as the weight of each negative sample to accelerate model convergence. Besides, recent research efforts from the community have proposed several non-sampling training strategies in All-Negative or Non-Negative ways [9, 13]. Unfortunately, these methods still have respective constraints and can only be applied to some specific KGE models. To reduce the space complexity, low-dimensional hyperbolic-based KGE models have drawn attention, such as MuRP, RotH, RefH and AttH [2, 5]. Although they can achieve good performance when using 32 or 64 dimensions, the calculations in the hyperbolic space are much more complicated than those in the Euclidean space [28].

Two valuable insights from the latest Contrastive Learning studies inspire us to analyze previous KGE efforts from a new perspective. First, Wang and Isola [33] identified key properties of Contrastive Learning, namely *Alignment* and *Uniformity*. Training negative samples to form a uniform vector distribution is beneficial for learning separable features. According to our analysis, different KGE training strategies share the same goal. Second, Wang and Liu [27] discovered that the temperature τ in the contrastive loss has a hardness-aware capability to control the strength of penalties on different negative samples. Similar attempts have been made in the hyperbolic geometry of recent hyperbolic KGE models and the self-adversarial loss [22]. These insights motivate us to re-think

the training target of KGE models and the weight assignment for different samples.

In this paper, we propose a novel KGE training framework, **H**ardness-**a**ware **L**ow-dimensional **E**mbedding (**HaLE**). To the best of our knowledge, our work is the first to achieve lightweight KGE training from a Contrastive Learning perspective. Instead of simply applying Contrastive Learning methods, we propose two innovative techniques: the *Query Sampling Loss* and the *Hardness-aware Activation Mechanism*. To balance the two properties, Alignment and Uniformity, the new loss function maximizes the scores of all positive instances, while forcing all entity vectors to stay away from a limited number of sampled query vectors in the vector space. As the result, the Query Sampling loss can provide stable gradients with small training costs. Furthermore, we propose the Hardness-aware Activation mechanism with novel activation functions, $Hanon(\cdot)$ and $Halin(\cdot)$, to flexibly control the strength of penalties of easy and difficult instances. Requiring few calculations, this mechanism has the potential to outperform the temperature trick and the hyperbolic geometry in hyperbolic-based KGE models.

We conduct extensive experiments on five commonly-used datasets, including FB15k-237, WN18RR, CoDEx-S/M/L. The results show that HaLE can significantly improve the training speed of multiple KGE models. Compared with previous KGE training strategies, the HaLE-trained models can obtain a higher prediction accuracy after training several minutes. Their performance is competitive compared to the state-of-the-art KGE models in both low- and high-dimensional conditions.

The rest of the paper is organized as follows. We introduce the background and notations in Section 2. Section 3 details the HaLE framework and its two major components. Section 4 reports the experimental studies, and Section 5 further discusses the related work. Finally, we provide some concluding remarks in Section 6.

2 BACKGROUND

In this section, we will briefly describe the Knowledge Graph Embedding and the Contrastive Learning techniques. The main notations that will be used in this paper are summarized in Table 4 in the Appendix A.1.

2.1 Knowledge Graph Embedding

A Knowledge Graph $\mathcal{G}$ is composed by a collection of triples (e_h, r, e_t), in which $r \in R$ is the relation between the head and tail entities $e_h, e_t \in E$. A KGE model is usually trained by the link prediction task to represent each entity $e \in E$ (or relation $r \in R$) as a d-dimensional continuous vector. Given a query $q = (e, r)$, link prediction aims to find the target entity $e_p \in E$ satisfying that (e, r, e_p) or (e_p, r, e) belongs to the knowledge graph $\mathcal{G}$.

To achieve this goal, the KGE model needs to score all candidate triples via a scoring function. Given a triple (e_h, r, e_t), the mainstream scoring function can be defined as $f(e_h, r, e_t) = \Psi(\Phi(e_h, r), e_t)$, which involves the following two operations:

- *Transform function* $\Phi(e_h, r)$ transforms the head vector $\mathbf{e}_h$ using the relation vector $\mathbf{r}$ and outputs the query vector $\mathbf{q}$;
- *Similarity function* $\Psi(\mathbf{q}, e_t)$ measures the similarity between the tail vector $\mathbf{e}_t$ and the transformed head vector $\mathbf{q}$.

Taken two typical KGE models, TransE [3] and DistMult [35], as examples, TransE's transform function is $\Phi(e_h, r) = \mathbf{e}_h + \mathbf{r}$ and its similarity function is equivalent to the L_1 or L_2 distance, while the transform function of DistMult is $\Phi(e_h, r) = \mathbf{e}_h \cdot \mathbf{r}$ and its similarity function is the dot product operation. The similarity score s produced by the scoring function $f(\cdot)$ is regarded as the triple score. Most KGE models are trained by minimizing a Negative Sampling loss, to make the score of the qualified triple higher than those of negative samples. In addition, recent researchers start to work on effective low-dimensional hyperbolic models in the KGE domain, which are detailed in Appendix A.2.

2.2 A Contrastive Learning Perspective

Contrastive Learning has achieved remarkable success in unsupervised representation learning for image and sequential data [6]. Without human supervision, Contrastive Learning methods attempt to learn the invariant representation of different views of the same instance by attracting positive pairs and separating negative pairs. Given a training set of positive pairs D_{pos}, a common design of the contrastive loss is formulated as:

$$\mathcal{L}_{cont} = \mathbb{E}\left[-\log \frac{exp(f(x_i, x_j)/\tau)}{\sum_k exp(f(x_i, x'_k)/\tau) + exp(f(x_i, x_j)/\tau)}\right], \quad (1)$$

where $(x_i, x_j) \in D_{\text{pos}}$, $f(\cdot)$ is the similarity function, and x' is the sampled negative instances. The temperature τ is a hyper-parameter, which controls the strength of penalties on different negative samples and provides a hardness-aware capability to discriminate between easy and difficult samples [27].

Recent research has proved that contrastive loss improves representation quality by optimizing two key properties asymptotically [33]: (1) Alignment: to achieve the training target, two samples forming a positive pair should be assigned by similar features; and (2) Uniformity: to preserve maximal information, feature vectors should be roughly uniformly distributed in the vector space. We argue that, to learn invariant representations in a self-supervised way, KGE and Contrastive Learning essentially share the common properties. Given an existing triple (e_h, r, e_t), the query $q = (e_h, r)$ and the tail entity e_t can be regarded as a positive pair. The KGE model assigns the positive triple with an optimal score, which is equivalent to aligning the query vector $\mathbf{q}$ with $\mathbf{e}_t$ and separating them with the other entity vectors.

Furthermore, compared with Contrastive Learning working on images or sequential data, KGE has its own characteristics, which inspire us to propose two innovative techniques. First, negative samples for any entity are in a fixed range, i.e., the KG entity set, such that a more efficient strategy instead of negative sampling can be utilized to achieve a uniform vector distribution. Second, the positive and negative samples of the entity are parameter-sharing and mutually restricted in the vector space. Therefore, training KGE models should focus on samples that are hard to distinguish and avoid to overfit easy samples.

3 METHODOLOGY

Based on the above comparative analysis between KGE and Contrastive Learning, and the newest findings in the two domains, we propose a novel training strategy for KGE models, namely **H**ardness-**a**ware **L**ow-dimensional **E**mbedding (**HaLE**). Specifically, we propose a new Query Sampling loss to achieve both Alignment

and Uniformity, and to overcome the drawbacks of the traditional Negative Sampling loss, which will be detailed in Sec. 3.1. To achieve the hardness-aware ability like the temperature mechanism in Contrastive Learning, we propose a novel Hardness-aware Activation mechanism in Sec. 3.2. Finally, the HaLE framework and several HaLE-based KGE models are described in Sec. 3.3.

3.1 Query Sampling Loss

Negative sampling has been proved effective and widely used to learn KG embedding and word embedding [3, 10]. Only considering a subset of negative instances, Negative Sampling helps reduce the time complexity of one training epoch. However, randomly sampling negative instances for each triple requires additional training time, especially for large-scale KGs. The uncertainty in the sampling procedure brings fluctuations into KGE training and impedes model convergence [30]. To omit Negative Sampling, recent work proposed two representative non-sampling approaches, i.e., All-Negative training and Non-Negative training [9, 13]. The general loss functions of the two approaches are as follows:

$$\mathcal{L}_{allneg}(T) = \mathop{\mathbb{E}}_{(e,r,e_p)\in T}\left[-log(\frac{exp(f(e,r,e_p))}{\sum_{i}^{n_E} exp(f(e,r,e_i))})\right], \tag{2}$$

$$\mathcal{L}_{nonneg}(T) = \mathop{\mathbb{E}}_{(e,r,e_p)\in T}\left[-f(e,r,e_p)\right] + g(\mathbf{E}), \tag{3}$$

where T is the triple set of $\mathcal{G}$ and $g(\mathbf{E})$ is a regularization function.

The two approaches have respective drawbacks. The former approach uses all entities except the target one as negative instances. It can provide a stable gradient for each epoch, while dramatically increasing computational complexity. In the latter approach, training positive triples only can minimize computation but tends to sink the model into a trivial optimum. Although previous work proposed some modifications to mitigate negative effects, they can only be applied to certain scoring functions and usually limit the prediction accuracy of KGE models.

Therefore, we argue that the feasible strategy replacing Negative Sampling should be somewhere between the two extreme approaches. We first combine the two training strategies to overcome their flaws. For all training triples in T, we sample a small subset of triples $\hat{T}$ to conduct the All-Negative training, while training the rest triples via the Non-Negative approach. Based on Eq. 2 and Eq. 3, we can re-organize the combined loss function as:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{allneg}(\hat{T}) + \mathcal{L}_{nonneg}(T-\hat{T}) \\
&= \mathop{\mathbb{E}}_{(e,r,e_p)\notin\hat{T}}\left[-f(e,r,e_p)\right] + \mathop{\mathbb{E}}_{(e,r,e_p)\in\hat{T}}\left[-log\frac{exp(f(e,r,e_p))}{\sum_{i}^{n_E} exp(f(e,r,e_i))}\right] \\
&= \mathop{\mathbb{E}}_{(e,r,e_p)\in T}\left[-f(e,r,e_p)\right] + \mathop{\mathbb{E}}_{(e,r,e_p)\in\hat{T}}\left[log\sum_{i}^{n_E} exp(f(e,r,e_i))\right] \\
&= -\frac{1}{n_T}\sum^{T} f(e,r,e_p) + \frac{1}{n_{\hat{T}}}\sum^{\hat{T}} LSE(f(e,r,E)),
\end{aligned} \tag{4}$$

where the number of sampled triples $n_{\hat{T}} = \alpha n_T$ and the hyper-parameter $\alpha \in [0,1]$ determines the sampling proportion. $LSE(\cdot)$ is the LogSumExp function, and the regularization part in $\mathcal{L}_{nonneg}$ is omitted.

Drawing on the idea of Contrastive Learning, we can explain the intuitive sense of the reorganized loss function. From Eq. 4 we can see that the reorganized loss function contains two modules, which exactly satisfy two key properties of Contrastive Learning, Alignment and Uniformity. The first module is to achieve the Alignment property by maximizing the scores of all positive triples. In the vector space, this module maximizes the similarity of the query vector and its target vector in every pair. The second module minimizes the similarity of each sampled query vector with all entity vectors in $\mathcal{G}$. As all entity vectors are forced to stay away from these query vectors, the vector distribution tends to be uniform. Focusing on the two key properties, we propose a new query sampling loss for KGE models, which is defined as:

$$\mathcal{L} = -\frac{\lambda}{n_T}\sum^{T} f(e,r,e_p) + \frac{1}{n_{\hat{T}}}\sum^{\hat{T}} LSE(f(e,r,E)), \tag{5}$$

where λ is a hyper-parameter to balance the contributions of two modules. As proved by a recent work [27], learning both Alignment and Uniformity is a trade-off process. A completely uniform vector distribution makes alignment impossible, while aligning all positive pairs perfectly causes the clustered vectors indistinguishable. Therefore, the hyper-parameter λ is necessary to keep KGE training stable.

A question may be raised on what is the difference between Negative Sampling and Query Sampling losses. Negative sampling loss focuses on the relative score order of the positive triple and negative samples. Only a few entity vectors are involved in each training batch and thus entity vectors have multifarious and ever-changing optimization directions. In the Query Sampling loss, staying away from sampled queries are the common training target shared by all entity vectors, so most entities have a stable optimization direction. As the goal of uniformity is clarified, we can also recognize the root flaws of the two extreme approaches. Non-Negative training utilizes the embedding regularization to achieve a global uniformity but ignores the specific query distribution, while All-Negative training is extremely redundant to separate all query and entity vectors. Considering above problems, our strategy samples a small proportion of queries from the query distribution to achieve a good uniformity property.

3.2 Hardness-aware Activation Mechanism

Although our query sampling loss can keep training stable, it treats all negative instances equally. As KGE training goes on, a large percentage of negative instances have been far away from the query vector, and thus provide less meaningful information. It would be more efficient if the loss can focus on negative instances that are difficult to be distinguished.

To solve this issue, the contrastive loss usually employs the temperature τ. As shown in Eq. 1, the feature similarities are multiplied by $\frac{1}{\tau}$ before going through the Softmax function. As proved by the recent work [27], this temperature gives the contrastive loss a hardness-aware ability to control the strength of

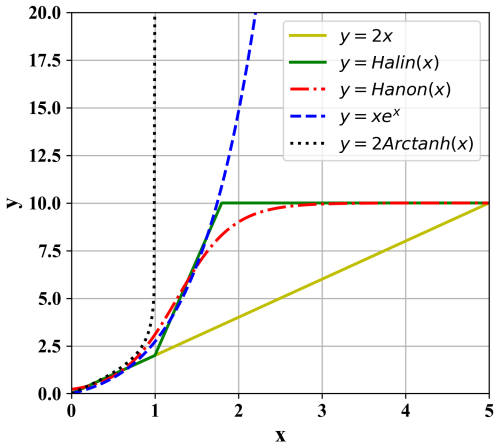

Figure 1: Different activation functions mentioned in this paper. The $y = 2x$ is equivalent to using temperature. The β values of $Hanon(\cdot)$ and $Halin(\cdot)$ are 3 and 10, respectively. The upper bound γ of two functions is 10. Best viewed in color.

penalties on hard negative samples. However, in the All-Negative training condition, the number of negative instances could be much higher. Therefore, the distribution of the Softmax-normalized scores would be much more uniform, even using a higher temperature.

This problem guides us to review the recent low-dimensional hyperbolic-based KGE models [2, 5]. We find a key point overlooked in the past: the nonlinear function in the hyperbolic distance is beneficial to achieve the hardness-aware ability. As described in Appendix A.2, there are two nonlinear functions utilized in previous low-dimensional KGE models, i.e., $h(x) = 2Arctanh(x) = \ln \frac{1+x}{1-x}$ and $h(x) = xe^x$. In Fig. 1, we can observe that their derivatives are always greater than one when $x \geq 0$. Besides, they have a similar trend as the value is relatively low at first and then increases rapidly. We argue that these nonlinear functions can be regarded as the activation function of KGE models. Given a KGE model with $\Psi(\mathbf{q}, e) = -\|\mathbf{q} - \mathbf{e}\|_2^2$, the instance with higher L_2 distance is more likely to be negative. The nonlinear activation function can further amplify the distance value of an easy negative instance. Such that, the penalties of the indistinguishable negative instances would be strengthened in the loss.

Based on our observations, we design a Hardness-aware Activation mechanism to replace the hyperbolic geometry. As shown in Fig. 1, two existing nonlinear functions are upper unbounded, so they can only achieve a 'soft constraint' to negative instances. Some easy instances that have been successfully separated are still involved in gradient computing. There is a significant waste on large-scale KGs and it also negatively affects the alignment of the other triples. To this end, we attempt to add a 'hard constraint' by designing suitable upper-bounded activation functions. Referring to the existing nonlinear functions, we propose two novel hardness-aware activation functions, $Hanon(\cdot)$ and $Halin(\cdot)$, which are formulated as follows:

$$Hanon(x) = \frac{1}{\gamma^{-1} + e^{-\beta(x-0.5)}} \tag{6}$$

$$Halin(x) = \begin{cases} 2x & 0 < x < 1 \\ min(\beta(x-1)+2, \gamma) & x \geq 1 \end{cases} \tag{7}$$

where β and γ are the two hyper-parameters to control the soft and hard constraints. We set the soft-constraint parameter β according to the slope of $h(x) = xe^x$. The curves of the two novel functions are shown in Fig. 1.

The $Hanon(\cdot)$ can be regarded as a variant of the Sigmoid function but be upper bounded by γ. The $Halin(\cdot)$ is a piecewise linear function. It has different slopes at the two sides of $x = 1$ and cuts the gradients when the value is more than γ. It is clear that $Hanon(\cdot)$ and $Halin(\cdot)$ have similar slopes to the two previous functions when $x \leq \gamma$. Then our functions use the 'hard-constraint' to cut the gradients of the distinguishable instances whose distance is bigger than γ. We further improve the Hardness-aware Activation by multiplying the similarity score with a relation-specific trainable parameter. Applied by previous low-dimensional KGE models, this technology is beneficial to encoding hierarchical relationships and improves prediction accuracy [28]. Finally, given a query (e, r) and its target entity e_p, the triple score based on L_2 distance via the Hardness-aware Activation is defined as:

$$f_{ha}(e, r, e_p) = -h(c_r \cdot \|\Phi(e, r) - \mathbf{e}_p\|_2^2), \tag{8}$$

where $h(\cdot)$ denotes the activation function $Hanon(\cdot)$ or $Halin(\cdot)$, and c_r is the relation-specific scalar parameter.

Note that, the Hardness-aware Activation mechanism proposed in this work can be an alternative to the hyperbolic geometry in low-dimensional KGE models, because the former inherits the advantages of both trainable curvatures and nonlinear activation. Besides, our solution is based on Euclidean space operations and only works in the training process, hence it is more efficient than hyperbolic KGE models. In Sec. 4, we will report our evaluation of multiple kinds of Euclidean-based KGE models using the proposed hardness-aware activation, and compare the performance of different activation functions.

3.3 The HaLE Framework

To achieve swift and sure KGE training, the HaLE framework utilizes the Query Sampling loss to keep training stable and the Hardness-aware Activation mechanism to accelerate model convergence. To integrate the two key techniques better, we apply the hardness-aware activation to two parts of our loss function in an asymmetric way. Specifically, we square the activated scores of positive instances in the Alignment part. Such that, the positive instances would get much stricter regularization than negative ones. The false-negative instances would get bigger gradients than negative instances, and a positive instance whose L_2 distance is close to zero would make less contribution to the loss. We find that it can accelerate the model convergence further. Therefore, the final loss function of the HaLE framework is formulated as follows:

$$\mathcal{L}_{HaLE} = -\frac{\lambda}{n_T} \sum^{T} f_{ha}(e, r, e_p)^2 + \frac{1}{n_{\hat{T}}} \sum^{\hat{T}} LSE(f_{ha}(e, r, E)), \tag{9}$$

To verify the performance of our HaLE framework, we select five representative KGE models: TransE [3], DistMult [35], RotatE

[22], RotE [5], and RotL [28]. These models utilize five different transform functions to generate the query vector in the Euclidean space, which are formulated as follows:

$$TransE: \Phi_T(e_h, r) = \mathbf{e}_h + \mathbf{r}, \tag{10}$$

$$DistMult: \Phi_D(e_h, r) = \mathbf{e}_h \cdot \mathbf{r}, \tag{11}$$

$$RotatE: \Phi_R(e_h, r) = Rot(\mathbf{r}, \mathbf{e}_h), \tag{12}$$

$$RotE: \Phi_E(e_h, r) = Rot(\mathbf{r}, \mathbf{e}_h) + \mathbf{r}', \tag{13}$$

$$RotL: \Phi_L(e_h, r) = Rot(\mathbf{r}, \mathbf{e}_h) \oplus_\alpha \mathbf{r}', \tag{14}$$

where $Rot(\cdot)$ denotes the vector rotation operation, $\mathbf{r}$ and $\mathbf{r}'$ are two different relation vectors corresponding to the same relation r. To make a fair comparison, we use the L_2-distance squared similarity function $\Psi(\mathbf{q}, e) = -\|\mathbf{q} - \mathbf{e}\|_2^2$. Although some previous works use the dot product function with specific regularization items, it has been proven to be equivalent to L_2 distance squared [37].

Compared with the previous Negative Sampling, All-Negative, and Non-Negative approaches, our HaLE framework can achieve swift and sure KGE training for several reasons:

- The training process in one epoch is greatly accelerated. The negative sampling for each triple is omitted, and the query sampling can significantly reduce the All-Negative training cost.
- HaLE can provide a stable training target. In each step of parameter optimization, we compute the gradients of all positive triples and force all entity vectors to stay away from the same part of queries.
- The total training time is reduced. The new loss can avoid parameter fluctuation, and the hardness-aware activation can focus on difficult instances. As a result, the HaLE-trained model can converge quickly in several epochs.

4 EXPERIMENTS

4.1 Experimental Setup

To verify the performance of HaLE, we focus on the link prediction, the most typical and challenging task for KG embeddings. Different from previous KGE research efforts that pursuing a higher prediction accuracy, we concentrate on the *training efficiency* of KGE models, which is critical for them to be applied in practice.

To compare the training efficiency of different strategies, we employ five representative KGE models as mentioned in Sec. 3.3 and train them in the specific space and time conditions. For the space condition, we set the dimension number of the low-dimensional models as 32 and high-dimensional ones as 256. For the time condition, we set a maximum training time according to the KG size of each dataset, as shown in Table 1. Following the previous work [3], we adopt two kinds of evaluation metrics in the 'Filter' mode: (1) MRR, 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.

4.1.1 Datasets. Our experimental studies are conducted on five commonly used datasets. WN18RR [4] is a subset of the English lexical database WordNet. FB15k237 [24] is extracted from Freebase including knowledge facts about movies, actors, awards, and sports.

Table 1: Statistics of the datasets.

Dataset	n_R	n_E	#Train	#Valid	#Test	Time
FB15k237	237	14,541	272,115	17,535	20,466	1,200s
WN18RR	11	40,943	86,845	3,034	3,134	600s
CoDEx-S	42	2,034	32,888	1,827	1,828	300s
CoDEx-M	51	17,050	185,584	10,310	10,311	1,200s
CoDEx-L	69	77,951	551,193	30,622	30,622	1,200s

Compared with the FB15k dataset, it removes inverse relations because many test triples can be obtained simply by inverting triples in the training set. CoDEx-S/M/L [17] are three KG datasets with different scales extracted from Wikidata and Wikipedia. We only use positive triples in each dataset for a fair comparison. The statistics of the datasets are given in Table 1. 'Train', 'Valid', and 'Test' refer to the number of triples in the training, validation and test sets.

4.1.2 Comparing Methods. We compare different training strategies mentioned in Sec. 3, including Negative Sampling (SamNeg) [5], Self-Adversarial Negative Sampling (AdvNeg) [22], All-negative Training (AllNeg) [9] and Non-negative Training (NonNeg) [13]. SamNeg and AdvNeg utilize the binary cross entropy loss, while AllNeg utilizes the cross-entropy loss to compute all candidate entities. We implement a general NonNeg strategy which uses a square loss to maximize positive triple scores and a global regularization to constrain the distance between each entity vector and the center vector of entity matrix. In the HaLE framework, we use the activation function $Hanon(\cdot)$ by default. We also compare multiple activation functions shown in the Fig. 1.

4.1.3 Implementation Details. We select the hyper-parameters of our model via grid search according to the metrics on the validation set. For previous strategies, we select the learning rate among $\{0.0005, 0.001, 0.005\}$, the number of negative samples among $\{50, 256, 512\}$, the batch size among $\{256, 512, 1,024\}$. For the HaLE framework, we select the sampling proportion α among $\{0.05, 0.1, 0.2\}$, the balancing ratio λ among $\{0.1, 0.3, 0.5, 1.0\}$, the hard-constraint parameter γ among $\{5, 10, 20\}$. All experiments are performed on Intel Core i7-7700K CPU @ 4.20GHz and NVIDIA GeForce GTX1070 GPU, and are implemented in Python using the PyTorch framework.

4.2 Experimental Results

As Negative Sampling is the most commonly used training strategy in the KGE domain, we first compare the limited-time performance of different KGE models trained by Negative Sampling and HaLE (with Hanon). Meanwhile, we provide public results of several fully-trained KGE models and measure their training time via the corresponding open-source codes. The 32-dimensional results on WN18RR and FB15k237 are shown in Table 2, while 256-dimensional results on three CoDEx datasets are shown in Table 3. Due to space constraint, the other experimental results including the ablation experiments and the visualization of entity embeddings can be found in Appendix A.3 and A.4.

4.2.1 Low-dimensional Performance Comparison. From Table 2, we have the following observations. Setting the limited training time as 20 minutes for FB15k237 and ten minutes for WN18RR, the five different models trained by HaLE significantly outperform

Table 2: Low-dimensional link prediction results on the WN18RR and FB15k237 datasets. The symbol '*' means the model is fully-trained, otherwise the model is trained in limited time. 'Cost' means training time (minutes). The best score of fully-trained models underlined and the best score of limited-trained models in Bold.

Type	Methods	FB15K237				WN18RR			
		MRR	Hits@10	Hits@1	Cost	MRR	Hits@10	Hits@1	Cost
Fully-trained Hyperbolic-based Models	MuRP [2]*	.323	.501	.235	335	.465	.544	.420	117
	RefH [5]*	.312	.489	.224	252	.447	.518	.408	84
	RotH [5]*	.314	.497	.223	242	.472	.553	.428	82
	AttH [5]*	.324	.501	.236	276	.466	.551	.419	94
Limited Time Negative Sampling Trained	TransE [3]	.243	.422	.154	20	.177	.417	.045	10
	DistMult [35]	.278	.445	.194	20	.351	.482	.283	10
	RotatE [22]	.223	.391	.141	20	.346	.460	.285	10
	RotE [5]	.246	.424	.159	20	.355	.480	.290	10
	RotL [28]	.140	.266	.079	20	.295	.368	.254	10
Limited Time HaLE Trained	TransE [3]	.314	.492	.224	20	.212	.492	.028	10
	DistMult [35]	.308	.483	.222	20	.447	.533	.399	10
	RotatE [22]	.307	.479	.219	20	.451	.536	.406	10
	RotE [5]	.313	.486	.226	20	.460	.542	.416	10
	RotL [28]	**.316**	**.493**	**.228**	20	**.471**	**.558**	**.424**	10

Table 3: High-dimensional link prediction results on the CoDEx datasets. The symbol '*' means the model is fully-trained, otherwise the model is trained in limited time. 'SamNeg-' means the model is trained by Negative Sampling, and 'Cost' means training time (minutes). The best score of fully-trained models underlined and the best score of limited-trained models in Bold.

Methods	CoDEx-S				CoDEx-M				CoDEx-L			
	MRR	Hits@10	Hits@1	Cost	MRR	Hits@10	Hits@1	Cost	MRR	Hits@10	Hits@1	Cost
RESCAL [12]*	.404	.623	.293	10	.317	.456	.244	34	.304	.419	.242	67
TransE [3]*	.354	.634	.219	9	.303	.454	.223	77	.187	.317	.116	34
ComplEx [25]*	.465	.646	.372	6	.337	.476	.262	87	.294	.400	.237	50
ConvE [7]*	.444	.635	.343	9	.318	.464	.239	139	.303	.420	.240	688
TuckER [1]*	.444	.638	.339	39	.328	.458	.259	152	.309	.430	.244	440
SamNeg-TransE [3]	.301	.544	.177	5	.178	.327	.107	20	.144	.260	.086	20
SamNeg-DistMult [35]	.360	.589	.246	5	.255	.395	.182	20	.228	.353	.164	20
SamNeg-RotatE [22]	.327	.546	.214	5	.182	.327	.110	20	.159	.281	.099	20
SamNeg-RotE [5]	.328	.549	.214	5	.183	.330	.112	20	.155	.270	.097	20
SamNeg-RotL [28]	.313	.534	.205	5	.162	.259	.106	20	.055	.113	.026	20
HaLE-TransE [3]	.353	.620	.223	5	.313	.467	.230	20	.300	.436	.226	20
HaLE-DistMult [35]	.403	.629	.289	5	.314	.462	.236	20	.299	.427	.230	20
HaLE-RotatE [22]	.407	.635	.289	5	.324	.474	.244	20	.302	.435	.229	20
HaLE-RotE [5]	**.409**	**.639**	.291	5	**.326**	**.475**	**.246**	20	**.308**	**.438**	.237	20
HaLE-RotL [28]	.408	**.639**	**.292**	5	.324	.474	.244	20	**.308**	**.438**	**.238**	20

the ones trained by Negative Sampling (SamNeg) on both datasets. The MRR and Hits@10 of all models have an average 5% increase. The results indicate the effectiveness of the HaLE framework. In the five models, the HaLE-trained RotL model achieves the best performance in all metrics on two datasets, HaLE-RotE is the second. It proves the effectiveness of the rotation-translation form in the transform function. In addition, we find that the SamNeg-trained RotL model is weaker than others, because the effect of the flexible addition operation relies on the nonlinear activation, which does not exist in the normal L_2-distance similarity function. Compared with fully-trained hyperbolic-based models, the simplest TransE model achieves competitive performance on FB15k237 after being trained in less than 20 minutes by HaLE. The HaLE-trained RotL model obtains the state-of-the-art MRR and Hits@10 on WN18RR, which has no hyperbolic geometry and only costs less than 10 minutes. The results indicate that HaLE can improve the Euclidean-base models in low-dimensional conditions, only spending around one tenth of the training time of the hyperbolic-based models.

4.2.2 High-dimensional Performance Comparison. In the high-dimensional condition, HaLE shows a significant advantage over Negative Sampling. In the limited training time, HaLE can accelerate model convergence while SamNeg-trained models fail due to the unstable gradients. This difference is more significant in the large-scale KGs. On the CoDEx-L dataset, the performance of TransE trained by HaLE is almost SamNeg-trained TransE. Table 3 also lists the results of five fully-trained KGE models using more than 256 dimensions, which are detailed-tuned by a powerful hyperparameter optimization method [17]. From the

table, we can see that the HaLE-trained models show strong competitiveness. Especially on CoDEx-M, the limited-time trained HaLE-TransE model (trained in 20 minutes) outperforms that of the 512-dimensional fully-trained TransE model (trained for more than one hour, 77 minutes). Although the Hits@1 of the optimal HaLE-RotL model is still lower than those benchmarks on CoDEx-L, HaLE-RotL achieves great Hits@10 results using less training time and fewer parameters. Training in a limited time, HaLE-trained models already obtain similar performance to the state-of-the-art models on five datasets. These results prove the efficiency of our HaLE framework on keeping high prediction accuracy.

4.2.3 Efficiency Comparison for Query Sampling. To verify the training efficiency of HaLE, we select the 32-dimensional RotE model and compare Query Sampling loss with four previous training strategies as mentioned in Sec. 4.1.2. The performance changes of the validation set as training proceeds are shown in the three upper line charts in Fig. 2. It is clear that our HaLE achieves remarkable efficiency on the three datasets comparing with the previous strategies. Besides, there are some common observations in the three results. Except NonNeg, SamNeg (Negative Sampling) is the worst one whose Hits@10 slowly increases in the first 50 seconds, indicating the negative effect of unstable gradients. Assigning different weights to negative instances, AdvNeg has a much faster convergence speed than SamNeg. Outperforming SamNeg and AdvNeg, the AllNeg strategy has good performance on two large datasets FB15k237 and CoDEx-M. As the WN18RR is relatively sparse, AllNeg is slightly inefficient to train all negative instances using a uniform gradient. These results prove the effectiveness of the All-Negative training and hardness-aware ability. Without negative instances, the NonNeg strategy is the only unstable one. Keeping increasing in the first 80 seconds, the Hits@10 of NonNeg starts decreasing, because its negative constraint is not powerful enough to avoid over-fitting. Our HaLE achieves the fastest convergence speed on the three datasets. Especially on WN18RR, HaLE achieves more than 0.5 Hits@10 in the first 30 seconds, which is already better than the final results of the others, indicating that HaLE can achieve a swift and sure KGE training, and has considerable potential in practical applications.

4.2.4 Efficiency Comparison for Hardness-aware Activation. To verify the hardness-aware activation mechanism and our novel activation functions, we compare the performance of different activation functions in the HaLE framework. The results are shown in the three lower line charts in Fig. 2. We can find similar observations on the three datasets. At first, utilized in hyperbolic models, the Arctanh-based function is obviously weaker than others. As its definition range is $(-1, 1)$, it relies on the normalization effect of hyperbolic geometry. In the two linear functions, the $y = 2x$ function is simulating the temperature control in Contrastive Learning and our $Halin(\cdot)$ can be regarded as an extended version of the former with soft- and hard-constraints. It is clear that the Hits@10 of $y = 2x$ increases faster in the first few rounds but gets a lower final Hits@10. Then we can see that the performance of $Hanon(\cdot)$, $Halin(\cdot)$ and $y = xe^x$ are very close. On the CoDEx-M and WN18RR datasets, our Hanon function performs the best. It is because $Hanon(\cdot)$ has an additional hard-constraint property to limit the outputted maximum. The $y = xe^x$ function used in the original RotL model is slightly better than $Hanon(\cdot)$ in FB15k237. The linear function $Halin(\cdot)$ achieves similar performance with the two nonlinear ones, proving that the soft- and hard-constraints are the key properties of the hardness-aware activation.

5 RELATED WORK

Recently, knowledge graph completion via KGE has been an active research topic [31]. Dozens of KGE models have been proposed, which can be divided into three categories from the perspective of the scoring function: (1) geometric distance based models, including TransE [3], TransD [8], RotatE [22], QuatE [36]; (2) tensor factorization based models, including RESCAL [12], DistMult [35], ComplEx [25], TuckER [1]; and (3) deep learning based models, including ConvE [7], ConvKB [11], RGCN [20], SACN [21], CompGCN [26]. All current KGE models suffer from the same issues of low speed and high cost in the training phase. The problem become much more serious when processing large-scale KGs with millions or billions of entities. Recently, several researchers have worked on this issue via different technical channels.

Reducing Parameters. Limiting the vector dimensions as 32 or 64, several low-dimensional KGE models are proposed to achieve competitive performance with less trainable parameters. MuRP [2] is the first KGE model based on hyperbolic vector space, and outperforms previous models in the low-dimensional condition. It embeds KG triples in the Poincaré ball model using the Möbius matrix-vector multiplication and Möbius addition operations. To further capture logical patterns in KGs, Chami et al. [5] propose a series of hyperbolic KGE models, including RotH, RefH, and AttH. These models utilize vector rotation or reflection operations to replace the multiplication operation between the head entity and relation vectors. Based on the RotH model, Wang et al. [28] propose two Euclidean-based lightweight models, RotL and Rot2L. Eliminating complex hyperbolic operations, the two models have lower computational complexity and faster convergence speed.

Replacing Negative Sampling. Most current KGE models are trained via Negative Sampling, which considers only a subset of negative instances to reduce the time complexity of each training epoch. However, Negative Sampling usually extends the convergence time of KGE models because of additional sampling calculations and unsteady training gradients. To solve this issue, Li et al. [9] propose an efficient All-Negative training framework and reduce the complexity of All-Negative calculations by dis-entangling the interactions between entities. However, this framework can only be applied to KGE models using a square-based loss. Its accuracy is lower than models trained by the Negative Sampling loss, especially in the low-dimensional condition. Peng et al. [13] employ segmented embeddings for parallel processing, and propose a Non-Negative strategy utilizing two vector constraints to replace Negative Sampling. However, these techniques cause a decrease in accuracy and force the model to use higher-dimensional vectors to represent each entity (e.g., 2,000 dimensions in [13]). Besides, this framework is based on Orthogonal Procrustes Analysis, and cannot be applied to existing KGE models directly.

Accelerating Model Convergence. Some recent research efforts design new loss functions to accelerate the convergence of KGE

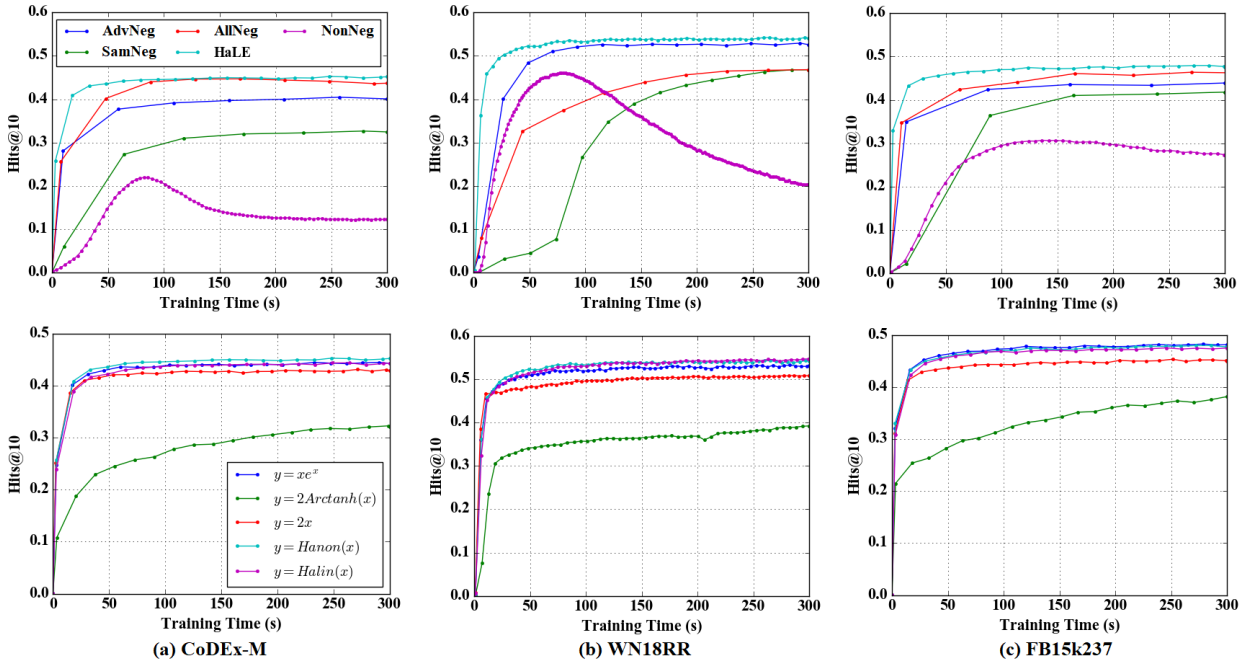

Figure 2: The Hits@10 of 32-dimensional RotE-based model as training proceeds on three datasets. The upper charts compare different loss functions, while the lower charts compare different activation functions. Best viewed in color.

models. Sun et al. [22] propose a self-adversarial negative sampling technique for efficiently and effectively training the RotatE model. It can be regarded as an improved binary cross-entropy loss, treating the normalized triple score as the weight of each negative sample. Another way is adding soft label loss based on the knowledge distillation technique. DistilE [38] utilizes an additional soft-label loss based on the knowledge distillation technique. A pre-trained high-dimensional KGE model generates soft labels for each training sample and accelerates the convergence of the small student model. MulDE [29] is a multi-teacher knowledge distillation framework for KGE models. Instead of a high-dimensional model, MulDE employs multiple low-dimensional models as teachers jointly supervising the student model via a novel iterative distillation strategy. Although the knowledge distillation framework can train a student KGE model quickly, it still requires the pre-training of teacher models and cannot really reduce training cost.

In this paper, we focus on the lightweight training of KGE models, which holds great potential for realizing a lifelong iterative process of many important Web applications such as Web search and recommendations. KGE can provide semantically-rich representations for entities, which can enhance the information extraction models to extract new knowledge triples from the Web. As HaLE effectively reduces the training time, KGE models can be rapidly updated to support iterative processing.

6 CONCLUSION

Recent knowledge graph embedding (KGE) models excessively pursue prediction accuracy but ignore the training efficiency. In this paper, we propose a novel Hardness-aware Low-dimensional Embedding (HaLE) framework to achieve a swift and sure KGE training. Motivated by the newest findings in the Contrastive Learning domain, we propose two key techniques: *Query Sampling Loss* and *Hardness-aware Activation*. We describe the connections of the two techniques with previous KGE achievements and prove their effectiveness by comparing with four previous training strategies in the link prediction task. The experimental results show that HaLE can achieve both higher prediction accuracy and faster convergence speed in limited training time.

These positive results encourage us to explore further research activities in the future. Instead of using artificially designed activation functions, we will apply the Neural Architecture Search technology to find more powerful activation functions automatically. Facing large-scale KGs, All-Negative training is still a burden. We aim to filter out some negative instances before measuring scores based on the hard-constraint mechanism. Finally, we will apply HaLE to other time-consuming KGE models such as ConvE [7] and SACN [21], and more KG tasks such as KG alignment and triple classification, to further verify the performance of the proposed framework.

ACKNOWLEDGMENTS

This research is partially supported by the National Natural Science Foundation in China (Grant: 61672128) and the Fundamental Research Fund for Central University (Grant: DUT20TD107). Quan Z. Sheng is partially supported by Australian Research Council (ARC) Future Fellowship Grant FT140101247, and Discovery Project Grant DP200102298. Kai Wang would like to thank the incomparable love and support from Dan Lin over the past decade. No matter how hard life may get unawares, his love for her is sweet and sure. He also would like to thank Macquarie University for offering the Cotutelle PhD Scholarship.

REFERENCES

[1] Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China, 5185–5194.

[2] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Multi-relational Poincaré Graph Embeddings. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 4465–4475.

[3] Antoine Bordes, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, December 5-8, 2013, Lake Tahoe, Nevada, United States*. 2787–2795.

[4] 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.

[5] 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.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. 1597–1607.

[7] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 1811–1818.

[8] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China, 687–696.

[9] Zelong Li, Jianchao Ji, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Chong Chen, and Yongfeng Zhang. 2021. Efficient Non-Sampling Knowledge Graph Embedding. In *Proceedings of WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 1727–1736.

[10] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

[11] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana, 327–333.

[12] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. 809–816.

[13] Xutan Peng, Guanyi Chen, Chenghua Lin, and Mark Stevenson. 2021. Highly Efficient Knowledge Graph Embedding Learning with Orthogonal Procrustes Analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. 2364–2375.

[14] Paolo Rosso, Dingqi Yang, Natalia Ostapuk, and Philippe Cudré-Mauroux. 2021. RETA: A Schema-Aware, End-to-End Solution for Instance Completion in Knowledge Graphs. In *Proceedings of the WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 845–856.

[15] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[16] 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.

[17] Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. 8328–8350.

[18] Tara Safavi, Danai Koutra, and Edgar Meij. 2020. Evaluating the Calibration of Knowledge Graph Embeddings for Trustworthy Link Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. 8308–8321.

[19] Graciela S.Birman and Abraham A.Ungar. 2001. The hyperbolic derivative in the Poincaré ball model of hyperbolic geometry. *J. Math. Anal. Appl.* 254 (2001), 321–333.

[20] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of the 15th Extended Semantic Web Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018*. 593–607.

[21] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 3060–3067.

[22] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[23] Pedro Tabacof and Luca Costabello. 2020. Probability Calibration for Knowledge Graph Embedding Models. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[24] 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.

[25] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 2071–2080.

[26] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[27] Feng Wang and Huaping Liu. 2021. Understanding the Behaviour of Contrastive Loss. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. 2495–2504.

[28] Kai Wang, Yu Liu, Dan Lin, and Michael Sheng. 2021. Hyperbolic Geometry is Not Necessary: Lightweight Euclidean-Based Models for Low-Dimensional Knowledge Graph Embeddings. In *Findings of EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*. 464–474.

[29] 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 WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 1716–1726.

[30] Menghan Wang, Mingming Gong, Xiaolin Zheng, and Kun Zhang. 2018. Modeling Dynamic Missingness of Implicit Feedback for Recommendation. In *Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 6670–6679.

[31] 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, 12 (2017), 2724–2743.

[32] Shen Wang, Xiaokai Wei, Cicero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, Philip S. Yu, and Isabel F. Cruz. 2021. Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion. In *Proceedings of the WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 1761–1771.

[33] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. 9929–9939.

[34] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with Knowledge Graph for Recommendation. In *Proceedings of the WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 878–887.

[35] 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 *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

[36] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. In *Proceedings of the Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 2731–2741.

[37] Zhanqiu Zhang, Jianyu Cai, and Jie Wang. 2020. Duality-Induced Regularizer for Tensor Factorization Based Knowledge Graph Completion. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[38] Yushan Zhu, Wen Zhang, Hui Chen, Xu Cheng, Wei Zhang, and Huajun Chen. 2020. DistilE: Distiling Knowledge Graph Embeddings for Faster and Cheaper Reasoning. *CoRR* abs/2009.05912 (2020). https://arxiv.org/abs/2009.05912

A APPENDIX

A.1 Summary of Notations

The main notations used in this paper and their descriptions are summarized in Table 4.

Table 4: Summary of the notations in this paper.

Symbol	Description
$\mathcal{G}$	A knowledge graph (KG)
T	The set of existing triples in a KG
E, R	The set of entities (E) or relations (R) in a KG
n_T, n_E, n_R	The item number in a specific set
e, r	An entity (e) or a relation (r) in a KG
$q = (e, r)$	A query containing an entity and a relation
e_p	The target entity of a query
$\mathbf{e}, \mathbf{r}$	Embedding vector of the entity e or the relation r
$\mathbf{q}$	Embedding vector of the query q
d	Dimension of the embedding vectors
$f(e_h, r, e_t)$	The scoring function of a KGE model
$\Phi(e_h, r)$	The transform function of a KGE model
$\Psi(q, e_t)$	The similarity function of a KGE model
$h(s)$	The activation function for triple scores

A.2 Hyperbolic KGE Models

Recently, researchers have proposed effective low-dimensional KGE models based on hyperbolic geometry, such as MuRP, RotH, RefH and AttH [2, 5]. These hyperbolic KGE models initialize the entity embedding vectors in the d-dimensional Poincaré ball [19] with negative curvature -c ($c > 0$): $\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 < \frac{1}{c}\}$, where $\|\cdot\|$ denotes the L_2 norm. The transform functions employed are similar to previous Euclidean-based models, but apply the Möbius addition and Möbius matrix-vector multiplication in the hyperbolic space. The similarity function is the key component of hyperbolic models. They utilize the hyperbolic distance to measure the similarity among entity vectors, which is defined as:

$$\Psi_{hyp}(\mathbf{q}, e_p) = -\frac{2}{\sqrt{c}} Arctanh(\sqrt{c}\| -\mathbf{q} \oplus_c \mathbf{e}_p\|)^2, \tag{15}$$

where $Arctanh(x) = \frac{1}{2}\ln\frac{1+x}{1-x}$, and $\oplus_c$ is the Möbius addition operation. To eliminate the complicated calculations in hyperbolic space, Wang et al. [28] analyzed the effect of hyperbolic geometry and proposed a lightweight Euclidean-based model RotL, whose similarity function is defined as:

$$\Psi_{euc}(\mathbf{q}, e_p) = -\varphi\left(\left\|\frac{\alpha(-\mathbf{q} + \mathbf{e}_p)}{1 - \mathbf{q} \cdot \mathbf{e}_p}\right\|\right) \tag{16}$$

where α is a relation-specific trainable parameter, and the Arctanh function is replaced by a simpler nonlinear function $\varphi(x) = xe^x$.

A.3 Ablation Experimental Results

We make a series of ablation experiments to evaluate different modules in HaLE. Specifically, we compare the performance of two training strategies after using our hardness-aware activation mechanism. In Table 5, 'Activation' and 'RelRatio' refer to the activation function and the trainable relation-specific parameters, while 'PosSquare' refers to using the square of positive scores. 'HA + SamNeg' and 'HA + AllNeg' represent applying hardness-aware activation to the SamNeg and AllNeg training strategies, respectively. The results prove the effectiveness of the three major modules in hardness-aware activation, and indicate that this mechanism can also increase the performance of Negative Sampling and All-Negative training strategies.

Table 5: The results of ablation experiments on the 32d HaLE-RotL model.

Dataset	Methods	MRR	Hits@10	Hits@1
CodexM	HaLE-RotL	**0.309**	**0.454**	**0.231**
	w/o Activation	0.289	0.436	0.212
	w/o PosSquare	0.241	0.351	0.185
	w/o RelRatio	0.293	0.438	0.214
	HA + SamNeg	0.297	0.443	0.219
	HA + AllNeg	0.262	0.415	0.182
FB15k237	HaLE-RotL	**0.316**	**0.493**	**0.228**
	w/o Activation	0.294	0.456	0.212
	w/o PosSquare	0.268	0.405	0.202
	w/o RelRatio	0.295	0.460	0.212
	HA + SamNeg	0.314	0.490	0.225
	HA + AllNeg	0.287	0.454	0.204
WN18RR	HaLE-RotL	**0.471**	**0.558**	**0.424**
	w/o Activation	0.426	0.520	0.376
	w/o PosSquare	0.469	0.558	0.420
	w/o RelRatio	0.454	0.548	0.404
	HA + SamNeg	0.457	0.553	0.394
	HA + AllNeg	0.449	0.540	0.387

A.4 Visualization of Entity Embeddings Distribution

We also visualize the vector distribution of entity embeddings using the t-SNE dimensionality reduction method. In Fig. 3, we compare the entity distributions of the best MRR checkpoint of multiple training strategies and find that model performance is related to the cluster formation in the entity distribution. On the WN18RR dataset, NonNeg and ALLNeg have a higher proportion of entities that are evenly distributed, indicating that they cannot effectively distinguish these entities. On CoDEx-M and FB15k237, NonNeg has fewer clusters than the other strategies, which explains why it gets relatively weak performance. On the contrary, there are more clusters and clearer boundaries between different clusters in HaLE's entity distributions, indicating that HaLE can better separate entity embedding vectors.

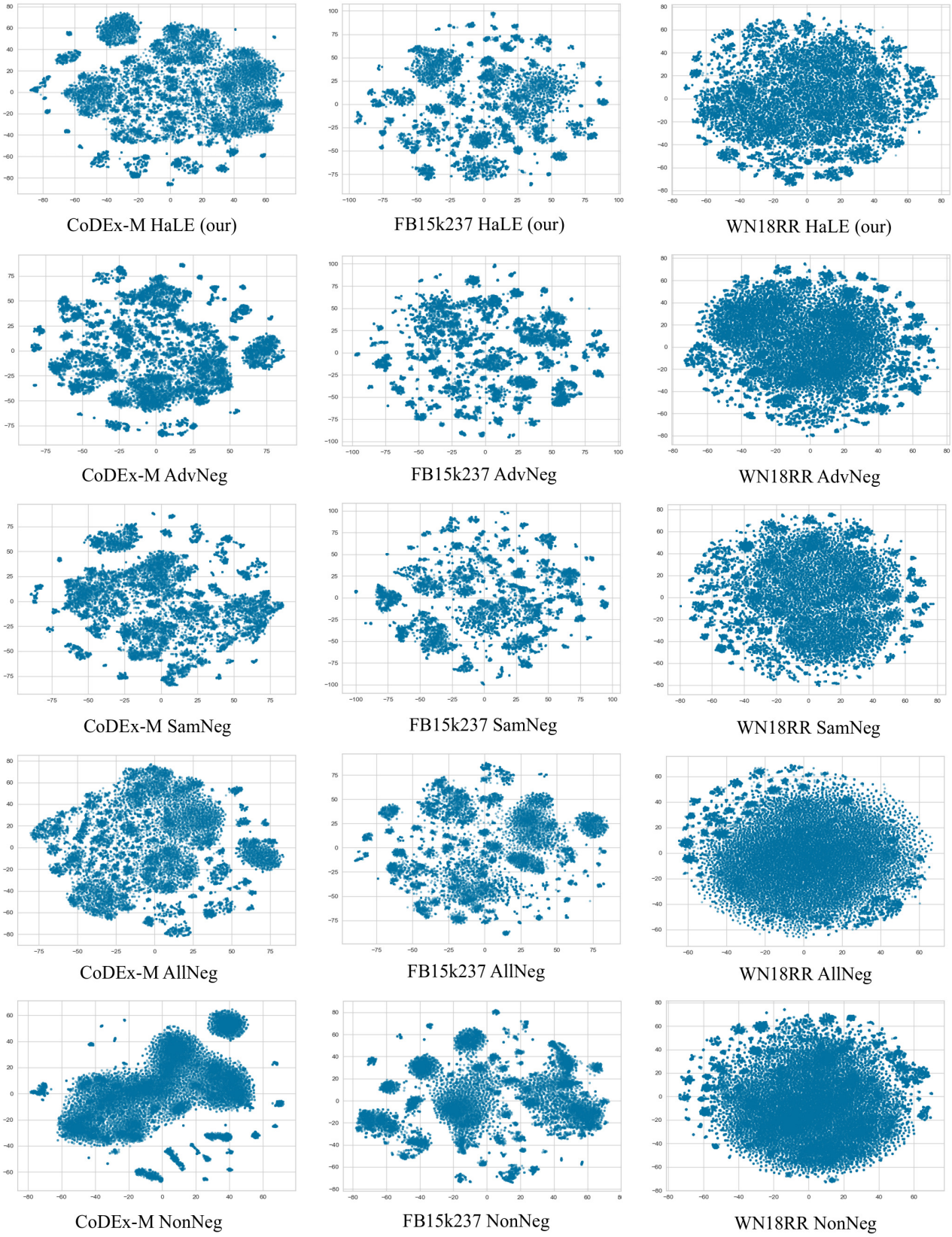

Figure 3: The visualization of entity embeddings of KGE models trained by different strategies. The vector distributions are generated by t-SNE.