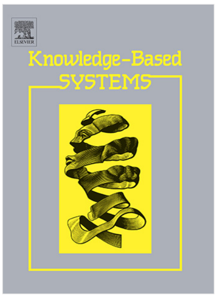
Journal Pre-proof



Knowledge graph embedding by translating in time domain space for link prediction

Qianjin Zhang, Ronggui Wang, Juan Yang, Lixia Xue

PII: S0950-7051(20)30693-6
DOI: https://doi.org/10.1016/j.knosys.2020.106564
Reference: KNOSYS 106564

To appear in: *Knowledge-Based Systems*

Received date : 27 May 2020
Revised date : 21 October 2020
Accepted date : 23 October 2020

Please cite this article as: Q. Zhang, R. Wang, J. Yang et al., Knowledge graph embedding by translating in time domain space for link prediction, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.106564.

Knowledge graph embedding by translating in time domain space for link prediction

Qianjin Zhang[a], Ronggui Wang[a], Juan Yang[a,*] and Lixia Xue[a]

[a] School of Computer and Information, Hefei University of Technology, Hefei 230601, China

ARTICLE INFO

Keywords:
Knowledge graph embedding
Link prediction
Cross-operation
Self-interaction

ABSTRACT

Knowledge graph embedding, which aims to learn distributed representations of entities and relations, has been proven to be an effective method for predicting missing links in knowledge graphs. Existing knowledge graph embedding models mainly consider that the space where head and tail entities are located has the same properties. However head and tail entities can be different types of objects, and they should not be located in the vector space with the same properties. In this paper, we propose a novel knowledge graph embedding model called TimE, which presents each entry of the head (or tail) entity embeddings as a point in time domain space, and the corresponding entry of tail (or head) entity embeddings as a point in frequency domain space. Specifically, TimE defines each relation as a composition relation, which consists of a translation between entities and a diagonal projection matrix that projects the entities into the time domain space. In addition, we propose a cross-operation to model inverse and symmetric relations. Experimental results show that TimE not only outperforms existing state-of-the-art models on several large-scale benchmark datasets for the link prediction task, but also can better capture diversity distribution of entity embeddings for different relation patterns effectively, and can also model all relation patterns (including symmetry/antisymmetry, inversion, and composition).

1. Introduction

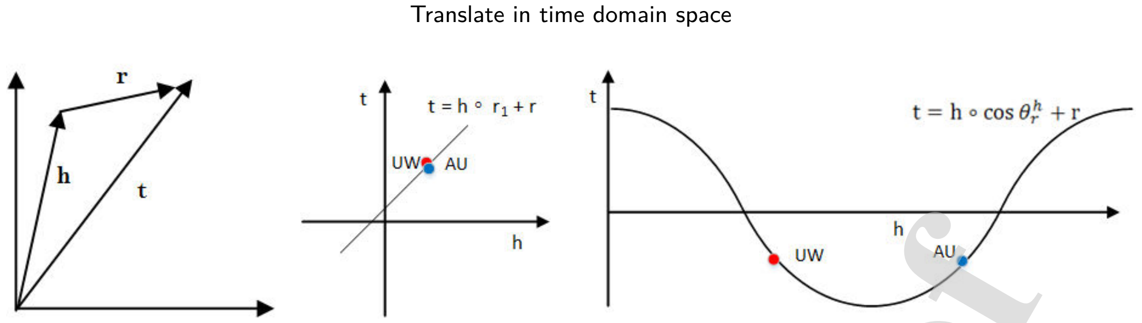
Knowledge graphs (KGs) such as Freebase [1],YAGO [2] and WordNet [3], are usually collections of factual triples, where each triple represents a human knowledge in the form of (head entity, relation, tail entity), or (h, r , t) in short. KGs have became a useful resources for many fields, such as question answering [4], knowledge inference [5], recommendation systems [6], [7]. Although some KGs are widely used for many practical AI tasks, they are still incomplete due to the limited or unknown information. Therefore, completing KGs and predicting missing links between entities have attracted much attention recently.

In a KG, facts are stored in the form of a directed graph, where entities are represented as nodes and relations between them as directed edges. However, as the limitation of symbolic representations of KGs, they cannot be directly applied to many machine learning tasks. Recently, inspired by word embeddings [8] that can encode semantic meaning of words into vectors, many embedding models [9–14] have proposed to embed entities and relations of a KG into low-dimensional vector spaces which can be used in various downstream tasks, such as logical queries [15],relation extraction [16], personalized search [17]. The existing embedding approaches can be roughly divided into three categories: translation-based methods, bilinear methods and neural network based methods. Translation-based methods measure the plausibility of a fact as the distance between the two entities after a translation enforced by the corresponding relation [18], e.g., TransE [9], TransR [19]. Bilinear methods measure the existence of a triple by matching latent semantics of entities and relations embodied in their vector space representations, [20], e.g., RESCAL [21], HolE [22], ComplEx [23]. Neural network based methods generally use a neural network as their score function, e.g. ConvE [24], ConvKB [25].

TransE [9] is the most representative translational distance model. Its main principle is that, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, if (h, r, t) is a fact in KG, where **h** and **t** are in the same space. Although the principle of TransE method is simple, it can capture the structure information of a KG efficiently, and has gained increasing popularity in recent years. Fig.1(a) gives a simple illustration of this idea. However, TransE does not perform well on 1-to-N, N-to-1, and N-to-N relations, and also can not work well on symmetric relations. Taking a N-to-1 relation *isLocatedIn* and a symmetric relation *friend* as an example, we have triples such as (*UW, isLocatedIn, Washington*), (*AU, isLocatedIn, Washington*), (*GU, isLocatedIn, Washington*), (*Y_LeCun, friend, G_Hinton*) and (*G_Hinton, friend, Y_LeCun*), where *UW, AU, GU* represent University of Washington, American University and Georgetown University, respectively. The entities *UW, AU, GU* will get the same embedding vectors and the relation *friend* will yield **0**, after the translation $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ is enforced by the relation *isLocatedIn* and relation *friend*, respectively.

In this paper, to address the above issues, we propose a novel knowledge graph embedding (KGE) model called TimE. Our motivation comes from the fact that head and tail entities can be different types of objects. As mentioned above, the type of head entity *UW, AU, GU* is university, while the type of tail entity *Washington* is place. According to the principle of TransE [9], each entry of entities and relations that are in triple (h, r, t) can be regarded as a linear function in a Cartesian coordinate system with h, t as the axes and r as the bias. However, as shown in Fig.1(b), if we project the head entity into a relation space with the same properties

[*] Corresponding author
ORCID(s): 0000-0002-1321-4329 (Q. Zhang)



(a) TransE (b) Linear projection of TransE (c) Non-linear projection of TimE

Fig. 1: Illustration of TransE, linear projection and non-linear projection, where ∘ represents Hadamard product, *UW* and *AU* represent University of Washington and American University, respectively. Subgraph (b) represents that when the head entity is projected into a relation-specific space using linear projection, the triple (h, r, t) is still a linear representation, where $\mathbf{r}_1$ represents the relation-specific linear projection vector. Subgraph (c) represents that the N-to-1 relations can be modeled by projecting the head entity in linear space into a relation-specific non-linear space, where $cos\boldsymbol{\theta}_r^h$ represents the relation-specific non-linear projection vector, $\boldsymbol{\theta}_r^t \in [0, 2\pi)^n$. Given two triples (*UW, isLocatedIn, Washington*) and (*AU, isLocatedIn, Washington*), according to subgraph (b), *UW, AU* will get the same embedding vectors, while according to subgraph (c), the embedding vectors of *UW, AU* are different.

as the tail entity, the representation of (h, r, t) is still a linear function with TransE flaws, because the linear function can only model 1-to-1 relation. As shown in Fig.1(c), for TimE, we use a non-linear projection matrix (e.g. cosine projection matrix) to project the head (or tail) into a relation-specific space, which has different properties from the tail (or head). By doing this, there will be multiple entities, such as the *UW, AU* and GU, matching the given (?, *isLocatedIn, Washington*), which indicates that if the space where the head and tail are located has different properties, the linear representation of triple (h, r, t) can be transformed into a non-linear representation to capture the differences between entities (Detailed analysis is listed in section 4) and model the 1-to-N, N-to-1 relations. On the other hand, entities in different relation type spaces should have different distributed representations. We refer to this as diversity distribution of entity. As shown in Fig.2, the types of relations "*married*" and "*president_of*" are symmetric and antisymmteric, respectively. The entities *USA, B_H_Obama* in triple (*B_H_Obama, president_of, USA*) and the entities *M_L_Obama, B_H_Obama* in triple (*M_L_Obama, married, B_H_Obama*) should have different distribued representations. It is worth noting that the same entity *B_H_Obama* is in different relation type spaces(i.e. symmetry and antisymmtery), which indicates that entity *B_H_Obama* should have different distributed representations for different relation types.

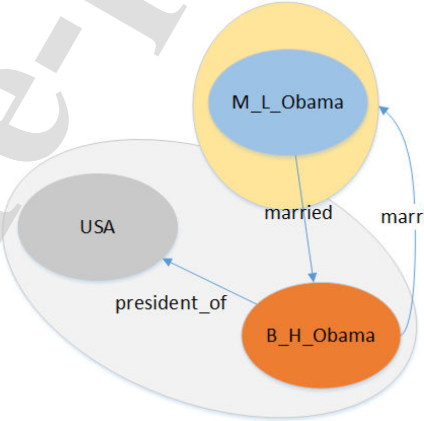


Fig. 2: Illustration of entities in different relation type spaces, where the relation *married* is a symmetric relation and the relation *president_of* is an antisymmetric relation.

Our proposed model TimE also belongs to the class of translation-based ones, which aims to model the diversity of distributed representations of entities and relations. Although there are some methods for modeling multiple types of entities and relations, they usually require additional data or process to obtain the diversity of distributed representations of entities and relations. For example, considering that entities and relations are in different space, CTransR [19] models the diversity of relations, but it requires additional cluster processing for relations; M-GNN [26] models the diversity of relation levels by coarsening the different structures contained in different relation types, however, it requires additional coarsen processing for different relation types, as well as additional neighbor information in KG; For the diversity of multi-step relation paths between entities, PaSKoGE [27] adaptively choose the optimal margin for each path by minimizing a path-specific margin-based loss function, nevertheless it needs to extract path information from original KG. Since TimE captures the diversity of distribution of entities and relations by projecting entities into relation-specific spaces with different properties, it only needs the triples in original KG, which is the data required by all KGE methods. Moreover, experimental results show that our proposed model TimE not only maintain simplicity and efficiency, but also can better capture the differences between entities and model all the relation patterns (including symmetry/antisymmetry, inversion, and composition), and the performance consistently outperform several state-of-the-art methods on the benchmark datasets.

In summary, our main contributions from this paper are as follows:

- We propose a novel KGE model TimE. It defines each relation as a composition relation, which consists of a translation between entities and a diagonal projection matrix that projects the entities into the time domain space. And TimE can better capture diversity distribution of entity embeddings.

- We propose a cross-opteration to model inverse and symmetric relations.

- This TimE learning framework applies to a wide range of KGE models, without the need of additional data or process.

- We evaluate our TimE for KG link prediction task on five benchmark datasets WN18RR [24], FB15k-237 [28], YAGO3-10 [29], WN18 [9], and FB15k [9]. Experimental results show that TimE with a modest parameter size achieves state-of-the-art results on complex and more challenging datasets.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 presents the details of TimE method. Section 4 presents the experimental protocols, datasets, evaluation metrics, and experimental results. In Section 5, we conclude our work and discuss future work.

2. Related work

Before proceeding, we first define our mathematical notations used in this paper. Let $\mathcal{E}$ and $\mathcal{R}$ represent the set of entities and relations respectively. We use lower-case letters h, r, and t to represent head entities, relations, and tail entities, respectively, where $h \in \mathcal{E}$, $t \in \mathcal{E}$, and $r \in \mathcal{R}$, and the corresponding bold lower-case letters $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ denote the column vectors of head entities, relations, and tail entities. Matrices are represented by bold upper-case letters, such as $\mathbf{M}$; score function by $f_r(\mathbf{h}, \mathbf{t})$. The i-**th** entry of a vector $\mathbf{x}$ is denoted as $[\mathbf{x}]_i$.

Let $\circ$: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ denote the Hadamard product between two vectors, i.e.,

$$[\mathbf{a} \circ \mathbf{b}]_i = [\mathbf{a}]_i \cdot [\mathbf{b}]_i$$

Let $\|\cdot\|_1$, $\|\cdot\|_2$ denote the ℓ_1 and ℓ_2 norm, respectively. Let $|\mathbf{x}|$ denote the absolute value function.

Let *diag(*$\mathbf{x}$), *diag(*cos$\mathbf{x}$) denote the diagonal projection maxtrix and diagonal cosine projection maxtrix, respectively.

A relation r is **symmetric** (**antisymmetric**) if r$(e_1, e_2) \Rightarrow$ r(e_2, e_1) (r$(e_1, e_2) \Rightarrow \neg$ r(e_2, e_1)) for any pair of entities e_1, e_2 $\in \mathcal{E}$. If any two entities e_1, e_2 such that $r_1(e_1, e_2) \Rightarrow r_2(e_2, e_1)$, relation r_2 is the **inverse** of relation r_1, and denotes r_2 as r_1^{-1}. Relation r_1 is **composed** of relation r_2 and relation r_3, if $r_2(e_1, e_2) \wedge r_3(e_2, e_3) \Rightarrow r_1(e_1, e_3)$ for any three entities e_1, e_2, e_3. For a golden triplet (h, r, t) corresponding to a fact in real world, it always gets a relatively higher score, and lower for a negative triplet.

Table 1 exhibits several previous state-of-the-art models as well as the model proposed in this paper. As described in Section 1, the existing KGE approaches can be roughly divided into three categories: translation-based models, bilinear models and neural network based models. Our proposed model TimE belongs to the class of translation-based ones.

2.1. Translation-based models

Translation-based models regard the relation as a translation vector $\mathbf{r}$ between the head entity $\mathbf{h}$ and tail entity $\mathbf{t}$. TransE [9] is the first translation-based model. The basic assumption of TransE is that entities and relations satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t} \in \mathbb{R}^n$, and the corresponding score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}$. However, the simple assumption may result in inability to deal with complicated entities and relations, such as 1-to-N, N-to-1 and N-to-N. To tackle this problem, TransH [30] allows entities to have distinct representations when involved in different relations, and the score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2$, where $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ are the first projections of entities onto relation-specific hyperplanes. TransR [19] deals with many-to-many problems by projecting the entity representations $\mathbf{h}$ and $\mathbf{t}$ into the space specific of relation $\mathbf{r}$, and the score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_2$. More recently, there are some other translation-based methods. For example, to overcome the problem of regularization in TransE, TorusE [11] defines KGE as translations on a compact Lie group, and the score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -min_{(x,y)\in([h]+[r])\times[t]}\|\mathbf{x} - \mathbf{y}\|_i$. In order to embed the common-sense concepts information of entities, KEC[32] measures the possibility of a triplet by projecting the loss vector onto a concept subspace, and the score function is defined as $f_r(h, t) = -\lambda\|\boldsymbol{l} - \mathbf{c}^T\boldsymbol{l}\mathbf{c}\|_2 + \|\boldsymbol{l}\|_2$, where $\boldsymbol{l} = \mathbf{h} + \mathbf{r} - \mathbf{t}$. To model and infer multiple types of relations, such as symmetry/anti-symmetry, inversion, and composition, RotatE [12] represents entities as points in a complex space and relations as rotations, and the corresponding score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_2$, where $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t} \in \mathbb{C}^k$, $|\mathbf{r}|_i = 1$.

2.2. Bilinear models

Bilinear models measure the existence of a triple by matching latent semantics of entities and relations embodied in their vector space representations. RESCAL [21] is one of the earliest bilinear model. Each relation is represented as a full rank matrix, and the score function of a fact (h, r, t) is defined as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^T\mathbf{M}_r\mathbf{t}$. Due to higher redundancy of full rank matrices, RESCAL is prone to overfitting. Extensions of RESCAL have been proposed by restricting on matrix $\mathbf{M}_r$. DistMult [31] restricts the matrices $\mathbf{M}_r$ to diagonal matrices, and the score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^T diag(\mathbf{r})\,\mathbf{t}$. However, these simplified models have the problem that they can only deal with symmetric relations which is not powerful enough for general KGs. HolE [22] shares the expressive power of RESCAL and simplicity of DistMult by employing circular correlation. ComplEx [23] extends DistMult [31] into the complex number field and takes the conjugate of the embedding of the tail entity to better model asymmetric and inverse relations. ANALOGY [33] adds normality

Table 1
The score functions of several KGE models, where $\langle \cdot \rangle$ denotes the generalized dot product, $\circ$ denotes the Hadamard product, $\bigotimes$ denotes circular correlation, σ denotes activation function and $*$ denotes 2D convolution. $\bar{\cdot}$ denotes conjugate for complex vectors in ComplEx, and 2D reshaped embeddings for real vectors in ConvE.

Model	Score Function	Parameters
TransE[9]	$-\lVert \mathbf{h}+\mathbf{r}-\mathbf{t} \rVert_{1/2}$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
TransH[30]	$-\lVert \mathbf{h}_\perp+\mathbf{r}-\mathbf{t}_\perp \rVert_2$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
ManiFoldE[14]	$-(\lVert \mathbf{h}+\mathbf{r}-\mathbf{t} \rVert_2^2-\theta_r^2)^2$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
RotatE[12]	$-\lVert \mathbf{h}\circ\mathbf{r}-\mathbf{t} \rVert_2$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{C}^k,\ \lvert\mathbf{r}\rvert_i = 1$
HolE[22]	$\langle \mathbf{r},\ \mathbf{h} \bigotimes \mathbf{t} \rangle$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
DistMult[31]	$\mathbf{h}^T diag\,(\mathbf{r})\ \mathbf{t}$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
ComplEx[23]	$\mathrm{Re}(\mathbf{h}^T diag\,(\mathbf{r})\ \bar{\mathbf{t}})$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{C}^k$
ConvE[28]	$\left\langle \sigma\left(vec\left(\sigma\left(\left[\bar{\mathbf{r}},\bar{\mathbf{h}}\right] * \omega\right)\right)\mathbf{W}\right),\mathbf{t}\right\rangle$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$
TimE(ours)	$-\lVert \mathbf{h}+\mathbf{r}-\mathbf{t}_\perp \rVert_{1/2}-\lVert \mathbf{t}-\mathbf{r}-\mathbf{h}_\perp \rVert_{1/2}$, where $\mathbf{t}_\perp = \mathbf{t}^T\, diag\left(\cos\boldsymbol{\theta}_r^t\right)$ $\mathbf{h}_\perp = \mathbf{h}^T\, diag\left(\cos\boldsymbol{\theta}_r^h\right)$	$\mathbf{h},\ \mathbf{r},\ \mathbf{t} \in \mathbb{R}^n$ $\boldsymbol{\theta}_r^t \in [0,2\pi)^n$ $\boldsymbol{\theta}_r^h \in [0,2\pi)^n$

constrains to $\mathbf{M}_r$ so as to model the analogical properties of entities and relations.

2.3. Neural network based models

Neural network-based models have attracted much attention in recent years. Neural Tensor Network (NTN) [34] represents each relation as a bilinear operator and uses a fully connected neural network to give a score for a given triple. ConvE [24] and ConvKB [25] use convolutional neural networks (CNNs) to capture the relationships among different dimensional entries of entities, and a fully connected neural network as its score function. R-GCN [20] uses Graph Convolutional Networks as its encoder to obtain embeddings of entities, and a task-specific decoder to learn embeddings of relations. In order to address the issue of vanishing gradient on discrete data and accelerate the convergence of KGE model, [35] proposed an adversarial network with the Wasserstein distance.

Two relevant works to our method are TransD [36] and RotatE [12]. TransD models the diversity of relations and entities by using dynamic matrics to project entities into a relation-specific space. Howerver, for the connectivity patterns of relations, it underperforms well on symmetric relations. For RotatE [12], despite it can model all the connectivity patterns of relations, it does not take into account the diversity of relations and entities. In this paper, our proposed model TimE focuses on both the connectivity patterns and the diversity of relations and entities. Experimental results show that modeling the relation levels can improve the performance on the datasets with more hierarchical relations. However, for TimE, it does not consider the relation levels, and we leave the work of modeling the hierarchy of relations with TimE as our future work.

3. Our approach

In this section, we provide details of our proposed model TimE. Our model defines each relation as a composition relation, which consists of a translation between entities and a diagonal projection matrix that projects the entities into the time domain space.

Definition 1. *For $\forall h$, t, if r is symmetric (or inverse), and $r(h,\ t_\perp)$, $r(h_\perp,\ t)$ hold, where $h_\perp$ and $t_\perp$ represent the projection of h and t in a relation-specific space, respectively, then:*

$r(t_\perp,\ h)$ (or $r^{-1}(t_\perp,\ h)$) corresponding to $r(h,\ t_\perp)$ are holded on the symmetric (or inverse) relation corresponding to $r(h_\perp,\ t)$, and vice versa.

A clause with such form is a cross-operation

3.1. Motivation

Our method is inspired by the generation process of sine wave or cosine wave in time domain space, which can be regarded as the projection of a uniformly rotating circle in the frequency domain space on a straight line. The basic idea of TimE is as follows (see Fig.3): For predicting tail, i.e. (h, r, ?), first, we view each entry of the candidate tail entities $\mathbf{t}$ as a point on a uniformly rotating circle in the frequency domain space and the corresponding entry of head entity $\mathbf{h}$ as a point in time domain space. Second, we define the relation r as a composition relation, which consists of a relation-specific diagonal cosine projection matrix and a translation between entities, where the relation-specific diagonal cosine projection matrix projects the tail entities onto the cosine line on the right side of Fig.3. Finally, instead of using $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$, we use $\mathbf{h}+\mathbf{r} \approx \mathbf{t}_\perp$ to measure candidate (h, r, t) whether is a fact or not, where $\mathbf{t}_\perp = diag(\cos\boldsymbol{\theta}_r^t)$ presents the projection of $\mathbf{t}$ in time domain space.

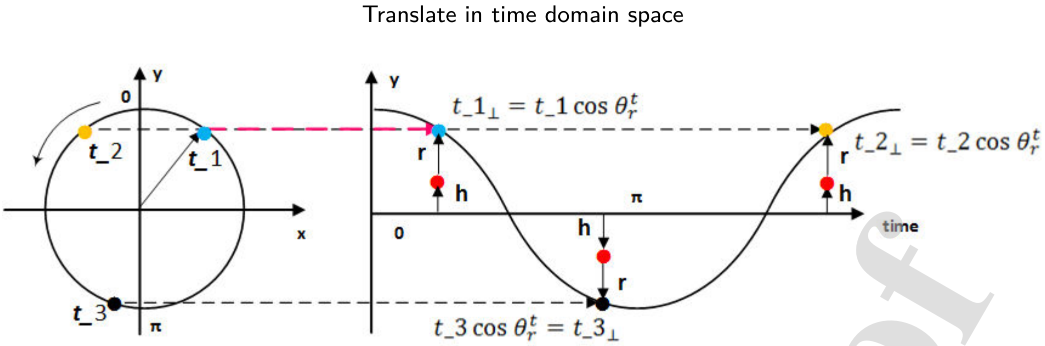

Translate in time domain space
y
0
t_2
t_1
x
t_3
π
y
$t\_1_\perp = t\_1 \cos\theta_r^t$
r
h
π
0
h
r
$t\_2_\perp = t\_2 \cos\theta_r^t$
r
h
time
$t\_3 \cos\theta_r^t = t\_3_\perp$

Fig. 3: Illustration of our proposed TimE. The left part represents the rotation of entities with the same modulus in frequency domain space, where the radius represents modulus of entities, arrow represents the direction of rotation, 0 represents the starting position of the rotation and each point on the circle represents an entry of the tail entity in the frequency domain space. And the corresponding cosine line on the right part represents the projection of the rotated entities on the straight line y, where each point on the cosine line represents the projection of tail entity from frequency domain space to time domain space at time i $\in$ (1, 2, 3) and r represents the translation between h and $t\_i_\perp$, i.e. h + r $\approx$ $t\_i_\perp$, i $\in$ (1, 2, 3) and $\cos\theta_r^t$ represents the entry of the diagonal cosine projection matrix of tail entities.

3.2. TimE

As described in Section 1, head and tail entities are different types of objects, they should not be in the same space. For given three triples (h, r, t_1), (h, r, t_2) and (h, r, t_3), as shown on the left side of Fig.3, each entry in the tail entities embeddings, they have the same modulus in frequency domain space, while the phase in time domain space on the right side of Fig.3 is different. And then, for each relation r, we define a diagonal relation-specific cosine matrix to project tail entities into time domain space. The element-wise projection of tail entities can be written as:

$$\mathbf{t}_\perp = \mathbf{t}^T \, diag\left(\cos\boldsymbol{\theta}_r^t\right) \tag{1}$$

where $\cos\boldsymbol{\theta}_r^t$ is a learnable parameter representing the diagonal cosine projection matrix of **t**. Specifically, each entry in the embeddings of tail entity and diagonal projection matrix represents the modulus of tail entity in frequency domain space and the phase in time domain space, respectively. Therefore, the amplitude of time domain can be represented by the embeddings of $\mathbf{t}_\perp$ and **h**. Then, for each relation r, we define an additional embedding to represent the translation between $\mathbf{t}_\perp$ and **h**. And the distance function of TimE can be defined as:

$$d_r^t(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}_\perp\|_{1/2} \tag{2}$$

3.3. Cross-operation

According to Eq. (2), if $\mathbf{r} = \mathbf{0}$, then

$$\mathbf{h} = \mathbf{t}^T \, diag\left(\cos\boldsymbol{\theta}_r^t\right) \tag{3}$$

which will yield $diag\left(\cos\boldsymbol{\theta}_r^t\right) = \pm\mathbf{1}$ for symmetric relations. Therefore, TimE addressed the flaws of TransE [9] which can not work well on symmetric relations. Given a triple (**h**, **r**, **t**), if r is a symmetric relation, the corresponding triple $\left(\mathbf{t}_\perp, \mathbf{r}, \mathbf{h}\right)$ should be a fact in KG, after projecting the tail entity into time domain space. However, according to the principle of TimE, the symmetric triple actually obtained is $\left(\mathbf{t}, \mathbf{r}, \mathbf{h}_\perp\right)$.

To address above issues, we propose a cross-operation to model symmetric and inverse relations. For each relation r, we define an additional diagonal relation-specific cosine matrix to project head entities into relation-specific time domain space, and the element-wise projection of head entities can be written as:

$$\mathbf{h}_\perp = \mathbf{h}^T \, diag\left(\cos\boldsymbol{\theta}_r^h\right) \tag{4}$$

where $\cos\boldsymbol{\theta}_r^h$ is a learnable parameter representing the diagonal cosine projection matrix of **h**. And the distance function of cross-operation can be defined as:

$$d_r^c(\mathbf{h}, \mathbf{t}) = \|\mathbf{t} - \mathbf{r} - \mathbf{h}_\perp\|_{1/2} \tag{5}$$

According to Eq.(2) and Eq.(5), if r is a symmetric (*or inverse*) relation, and $\left(\mathbf{h}, \mathbf{r}, \mathbf{t}_\perp\right)$ $\left(or\ \left(\mathbf{h}, \mathbf{r}^{-1}, \mathbf{t}_\perp\right)\right)$ holds on Eq. (2), then the corresponding $\left(\mathbf{t}_\perp, \mathbf{r}, \mathbf{h}\right)$ $\left(or\ \left(\mathbf{t}_\perp, \mathbf{r}^{-1}, \mathbf{h}\right)\right)$ will be holded on Eq. (5), and vice versa. That is why we named it cross-operation. By doing this, TimE can model all the relation patterns. Combining Eq. (2) and Eq. (5), the final distance function of TimE is:

$$d_r(\mathbf{h}, \mathbf{t}) = d_r^t(\mathbf{h}, \mathbf{t}) + d_r^c(\mathbf{h}, \mathbf{t}) \tag{6}$$

Self-interaction When evaluating the models, we find that adding a self-interaction of relations to the diagonal projection matrix can improve the performance of TimE. For simplicity, we represent the diagonal matrices $diag\left(\cos\boldsymbol{\theta}_r^t\right)$ and $diag\left(\cos\boldsymbol{\theta}_r^h\right)$ as column vectors $\cos\boldsymbol{\theta}_r^t$ and $\cos\boldsymbol{\theta}_r^h$ respectively. The element-wise projection of head entities and tail entities can be rewritten as:

$$\mathbf{t}_\perp = \mathbf{t} \circ \left(\cos\boldsymbol{\theta}_r^t + \boldsymbol{\theta}_r^t\right) \tag{7}$$

$$\mathbf{h}_{\perp} = \mathbf{h} \circ \left(\cos \boldsymbol{\theta}_r^h + \boldsymbol{\theta}_r^h\right) \tag{8}$$

3.4. Loss function

Following [12], we use negative sampling loss with self-adversarial training. The loss function defined as:

$$\begin{aligned} L = & - \log \sigma \left(\gamma - d_r\left(\mathbf{h}, \mathbf{t}\right)\right) \\ & - \sum_{i=1}^{n} p\left(h_i', r, t_i'\right) \log \sigma \left(d_r\left(\mathbf{h}_i', \mathbf{t}_i'\right) - \gamma\right) \end{aligned} \tag{9}$$

where γ is a fixed margin, σ is the sigmoid function, $\left(h_i', r, t_i'\right)$ is the i-**th** negative triple, and $p\left(h_i', r, t_i'\right)$ is the probability distribution of sampling negative triples, which is defined as:

$$p\left(h_j', r, t_j' \middle| \left\{\left(h_i, r_i, t_i\right)\right\}\right) = \frac{\exp \alpha f_r\left(\mathbf{h}_j', \mathbf{t}_j'\right)}{\sum_i \exp \alpha f_r\left(\mathbf{h}_i', \mathbf{t}_i'\right)} \tag{10}$$

where α is the temperature of sampling.

3.5. Number of parameters

The total number of parameters for TimE is $\left(n_e + 3n_r\right) \times d$, where n_e and n_r denote the number of entities and relations respectively, d is the embedding dimension. For general KGE models, there are $n_e + n_r$ embeddings, TimE only has $2n_r$ additional embeddings from the diagonal projection matrices. As $n_r \ll n_e$ in most KGs, this does not add a lot of extra parameters compared with the simplest translation-based model TransE [9].

3.6. Connection to RotatE

The projection of head or tail entities which defined in our proposed model TimE can be regarded as a rotation of the modulus part in frequency domain space. This shares similarities with RotatE [12], in which the authors report that they view each relation as a rotation from the source entity to the target entity in the complex vector space. For our TimE, if $\mathbf{r} + \mathbf{r}^{-1} = \mathbf{0}$ or $\mathbf{r} = \mathbf{0}$, TimE degenerates to a special case of RotatE with only real parts. However, the aims between RotatE and TimE are different. RotatE aims to model all the relation patterns including symmetry/antisymmetry, inversion, and composition. It encourages two linked entities have a same distribution, no matter what the relation pattern is. TimE views each relation as a composition relation. It aims to model the diversity distribution of entities, i.e. different relation patterns lead to different distributions of entities, and it can also model all the relation patterns mentioned above.

4. Experiments

4.1. Datasets

We evaluate TimE for KG link prediction task on five benchmark datasets WN18RR [24], FB15k-237 [28], YAGO3-10 [29], WN18 [9], and FB15k [9]. The details of these datasets are shown in Table 2.

Table 2
Statistics of datasets, where #Ent and #Rel denote the number of entities and relations, respectively, and #Tr, #Va, and #Te denote the number of training, validation, and test triples, respectively.

	FB15k	WN18	FB15k-237	WN18RR	YAGO3-10
#Ent	14,951	40,943	14,541	40,943	123,182
#Rel	1,345	18	237	11	37
#Tr	483,142	141,442	272,115	86,835	1,079,040
#Va	50,000	5,000	17,535	3,034	5,000
#Te	59,071	5,000	20,466	3,134	5,000

WN18 is extracted from WordNet [3], a large lexical knowledge graph about English. FB15k is extracted from Freebase [1], a huge knowledge graph that describes the general facts about movies, actors, awards,sports, and sport teams in the world. WN18RR and FB15k-237 are subsets of WN18 and FB15k respectively, in which inverse relations are deleted. YAGO3-10 is a subset of YAGO3 [29], in which contains relations among people and their attributes, and consists of entities with at least 10 relations per.

4.2. Experimental settings

Evaluation Protocol The link prediction task aims to predict a missing entity for a given relation and another entity, i.e, $(h, r, ?)$ or $(?, r, t)$. The optimal results can be calculated by ranking the scores on test triples according to the score function f_r.

Following [9], for each triple (h, r, t) in the test dataset, we replace either h or t by each of other entities in $\mathcal{E}$ to create a set of corrupted candidate triples and calculate their scores according to the score function Eq.(6). Then we rank the scores in descending order.We also use "Filtered" settings as in [9] which does not take triples appearing in the training, validation, or test set into accounts at ranking. We choose three standard evaluation measures as our evaluation metrics: Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits at N (H@N).

Training Protocol We implement our model using Pytorch with Adam optimizer [37], and perform a grid search for optimal hyperparameters. The grid search settings are as follows: dimension embedding among $\{125, 250, 500, 1000\}$, ℓ_1-norm or ℓ_2-norm, batch-size among $\{512, 1024, 2048\}$, margin γ among $\{3, 6, 9, 12, 18, 24, 30\}$, self-adversarial sampling temperature α among $\{0.5, 1.0\}$. We add two additional coefficients to the self-interation and distance function at training, i.e. $\mathbf{t}_{\perp} = \mathbf{t} \circ \left(\omega_{11} \cos \boldsymbol{\theta}_r^t + \omega_{12} \boldsymbol{\theta}_r^t\right)$, $\mathbf{h}_{\perp} = \mathbf{h} \circ \left(\omega_{21} \cos \boldsymbol{\theta}_r^h + \omega_{22} \boldsymbol{\theta}_r^h\right)$, $d_r\left(\mathbf{h}, \mathbf{t}\right) = \lambda_1 d_r^t\left(\mathbf{h}, \mathbf{t}\right) + \lambda_2 d_r^c\left(\mathbf{h}, \mathbf{t}\right)$, where $\omega_{11}, \omega_{12}, \omega_{21}, \omega_{22}, \lambda_1, \lambda_2 \in \mathbb{R}$.

4.3. Main results

In this part, we show the link prediction results of TimE on five common datasets, as well as several state-of-the-art models with their published results from the original papers. Our results on FB15k and WN18 are shown in table 3; results on FB15k-237, WN18RR, and YAGO3-10 are shown

Table 3
Link prediction results on FB15k and WN18. Results [★] are taken from [22]. Other results are taken from the corresponding original papers.

	FB15k					WN18				
	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10
TransE [9] [★]	.463	-	.297	.578	.749	.495	-	.113	.888	.943
TransD(bern) [36]	-	91	-	-	.773	-	212	-	-	.922
HolE [22]	.524	-	.402	.613	.739	.938	-	.930	.945	.949
ComplEx [23]	.692	-	.599	.759	.840	.941	-	.936	.945	.947
R-GCN+ [38]	.696	-	.601	.760	.842	.819	-	.697	.929	.964
M-GNN+ [26]	.747	-	.671	.795	.876	.948	-	.940	.950	**.967**
PaSKoGE [27]	-	-	-	-	.88	-	-	-	-	.95
SimplE [39]	.727	-	.660	.773	.838	.942	-	.939	.944	.947
TorusE [11]	.733	-	.674	.771	.832	.947	-	.943	.950	.954
ConvE [24]	.745	64	.670	.801	.873	.942	504	.935	.947	.955
RotatE [12]	**.797**	**40**	.746	**.830**	**.884**	**.949**	309	**.944**	.952	.959
TimE(**ours**)	.795	45.9	**.747**	.825	.879	**.949**	**259**	.943	**.954**	.961

Table 4
Link prediction results on FB15k-237, WN18RR and YAGO3-10. Results [♣] are taken from [25]. Results [♡] are taken from [24]. Other results are taken from the corresponding original papers
.

	FB15k-237					WN18RR					YAGO3-10				
	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10
TransE [9] [♣]	.294	347	-	-	.465	.226	3384	-	-	.501	-	-	-	-	-
ComplEx [23] [♡]	.247	339	.158	.275	.428	.44	5261	.41	.46	.51	.36	6351	.26	.40	.55
ConvE [24]	.316	246	.239	.350	.491	.46	5277	.39	.43	.48	.52	2792	**.45**	.56	.66
ConvKB [25]	**.396**	257	-	-	.517	.248	**2554**	-	-	.525	-	-	-	-	-
RotatE [12]	.338	177	.241	.375	.533	.476	3340	**.428**	.492	.571	.495	1767	.402	.550	.670
TimE (**ours**)	.346	**171**	**.250**	**.382**	**.537**	**.477**	2858	**.428**	**.493**	**.577**	**.530**	**1685**	.442	**.586**	**.688**

in table 4.

From Table 3 we can see that:

1) On FB15k, TimE is the best model in terms of H@1, better than that of RotatE [12]. For the other four metrics, TimE obtains the second best performance, which is slightly worse than the previous state-of-the-art RotatE. Strikingly, TimE obtains a 3.5% higher H@10 than R-GCN+ [20], which can capture more additional neighbor information.
2) On WN18, TimE achieves the best H@3, MRR, and MR. In terms of H@10, TimE is slightly worse than that of R-GCN+ [20].
3) From the comparison of the experiment results, we can see that TimE can capture the diversity distribution of entities, and obtains comparable performance compared with previous state-of-the-art models on these two datasets which consist of all the relation patterns. On FB15k almost 81% of the test set triplets (e_1, r, e_2) can be inferred by either (e_1, r^{-1}, e_2) or (e_2, r^{-1}, e_1) that appeared in the training set. On WN18, the ratio reached 94% [28]. Therefore, the main relation patterns are inversion on these two datasets. TimE will yield $\mathbf{r} + \mathbf{r}^{-1} = \mathbf{0}$ for inverse relations and $\mathbf{r} = \mathbf{0}$ for symmetric relations. In this case, TimE degenerates to a special case of RotatE with only real parts. That is why our proposed model doesn't significantly outperform the previous state-of-the-art on FB15k and WN18.

From Table 4 we can see that:

1) On YAGO3-10, TimE obtains a 0.035 higher MRR, 4% higher H@1, 3.6% higher H@3, 1.8% higher H@10 than RotatE, respectively.
2) On FB15k-237, TimE obtains a 0.9% higher H@1, 0.7% higher H@3 than RotatE, respectively.
3) On WN18RR, TimE performs as good as RotatE on metrics of H@1. In terms of H@3, H@10, and MRR, TimE is the best model, better than that of RotatE.
4) Compared to WN18RR, YAGO3-10 and FB15k-237 contain entities with high relation-specific indegree [24]. For example, for link prediction task, given a relation and a tail entity (?, *isLocatedIn*, *France*), the missing head entity corresponds to over 900 true entities. From the comparison of the experiment results, it can also see that TimE significantly outperforms the previous state-of-the-art on YAGO3-10 and FB15k-237 which indicates that TimE can capture the differ-

ences between entities efficiently. However, although the inverse relations are removed in WN18RR, most of the relation patterns in it are symmetry, such as *_simialr_to*, *_also_see*. So, on WN18RR, TimE doesn't significantly outperform the previous state-of-the-art as much as that on YAGO3-10 and FB15k-237. This is consistent with our hypothesis that the head and tail entities are located in different spaces.

4.4. Analysis

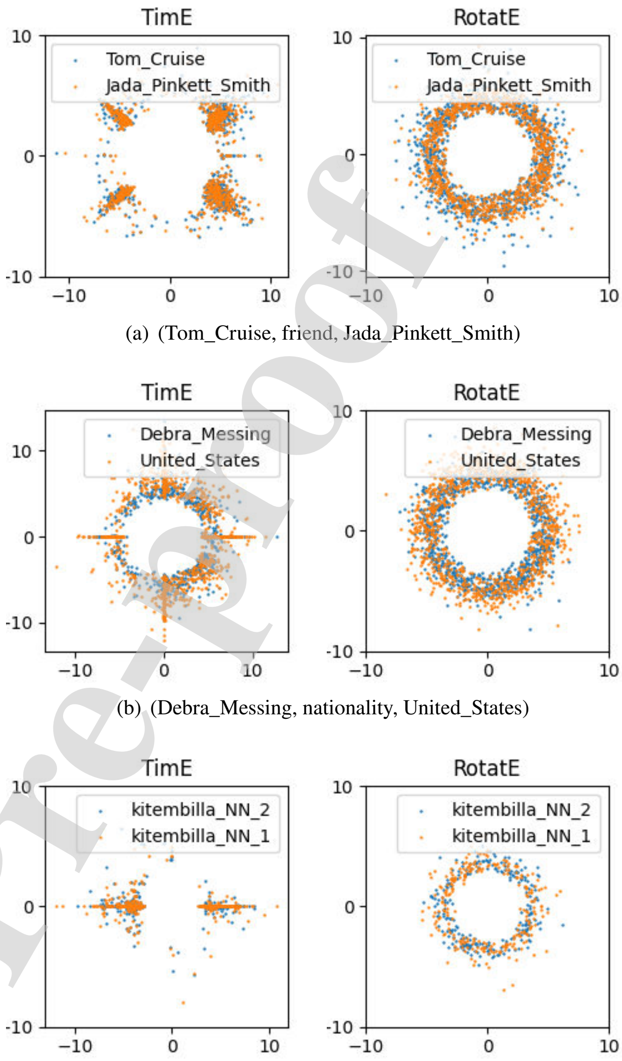
In this part, we first verify whether the differences between entities are captured efficiently by TimE. Then, we show that cross-operation can model all relation patterns including symmetry/antisymmetry, inversion, and composition by analyzing the embedding of relations. Finally, we investigate parameter efficiency of TimE.

Entity embeddings For a fair comparison, we compare our proposed TimE with the previous state-of-the-art RotatE [12], which also uses self-adversarial technique proposed in [12]. As mentioned in Section 2, RotatE represents each element in the entity embeddings as a point in a complex space, which also can be regarded as a point on a 2D plane. So we can plot the entity embeddings on a 2D plane. For TimE, each entry in the embeddings of the head entity h (or tail entity t) and the diagonal projection matrices represent the modulus in frequency domain space and the phase in time domain space , respectively. Therefore, the entity embedding generated by our proposed model TimE also can be plotted on a 2D plane by transforming the head and tail entity embedding into the same frequency domain space. Following [40], we use logarithm operation trick to better show the differences between entities.

Intuitively, entity embeddings have different distributions for different relation patterns. The first column of Fig.4 gives an illustration of the diversity of entity embedding distributions of TimE, where *friend* represents relation */celebrities/celebrity/celebrity_friends./celebrities/friendship/friend* and nationality represents relation */people/person/nationality*.

From the first column of Fig.4 we can see that:

1) Generally, for symmetry pattern, it needs the head and tail entities satisfy r(h, t) and r(t, h). However, according to the cross-operation, the symmetric pattern also can be represented by r(h, -t) and r(-t, h). Fig.4(a) shows that all the entries of the head entity embeddings or tail entity embeddings on a 2D plane are clustered near the straight line y = x and y = -x, which indicates that the symmetric distribution of entities can be captured by TimE.
2) As shown in Fig.4(b), for antisymmetry pattern, since there are no other constrains except for the translation between entities, all the entries of the entity embeddings are evenly located on the circle.
3) From Fig.4(c) we can see that all the entries of the head entity embeddings or tail entity embeddings on a 2D plane are clustered near the straight line y = 0, which indicates that $\boldsymbol{\theta}_r^t = \boldsymbol{\theta}_r^h = \mathbf{0}$ or $\boldsymbol{\theta}_r^t = \boldsymbol{\theta}_r^h = \boldsymbol{\pi}$. Therefore, for inversion pattern, the score function of TimE is the same as TransE [9], which indicates that TimE can capture the inverse distribution of entities, since it is reported in [9] that TransE can model the inverse relation pattern.
4) In summary, from the first column of Fig.4, we can see that for different relation patterns, all the entity embeddings learned by TimE have different distributed representations. However, in RotatE, entity embeddings in all the second columns of the three subgraphs of Fig.4 have the same distribution, which make it difficult to capture the diversity distributions of entity embeddings.



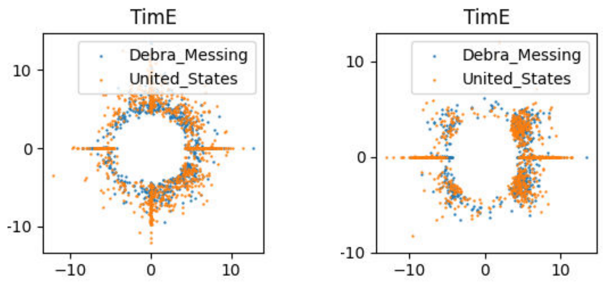
(a) (Tom_Cruise, friend, Jada_Pinkett_Smith)

(b) (Debra_Messing, nationality, United_States)

(c) (__kitembilla_NN_2, _part_of, __kitembilla_NN_1)

Fig. 4: Visualization of the embeddings of several entity pairs for different relation patterns. Subgraphs (a), (b), (c) represent symmetric relation, antisymmetric relation, and inverse relation, respectively. The triples in (a) and (b) are extracted from FB15k-237, and the triple in (c) is extracted from WN18.

As mentioned in Section 3, for TimE, given a triple (h,r,t), the head entity h and tail entity t are located in different spaces (time domain or frequency domain), which indicates



(a)

(b)

Fig. 5: (Debra_Messing, nationality, United_States). Visualization of the embeddings in different time domain space of TimE. Subgraph (a) represents mapping head entity into time domain space. Subgraph (b) represents mapping tail entity into time domain space

that TimE can better capture the differences between entities of antisymmetric relation triples. The visualization results in Fig.5 are as expected.

Relation embeddings In Figure 5, we plot the distribution histograms of four relation patterns to investigate that TimE can model all the relation patterns by our proposed cross-operation. In this part, we ignore the influence of relational self-interaction, which will be investigated detail in ablation studies. For simplicity, we denote $\boldsymbol{\theta}_r^t + \cos\boldsymbol{\theta}_r^t$ and $\boldsymbol{\theta}_r^h + \cos\boldsymbol{\theta}_r^h$ as $\mathbf{r}_h$ and $\mathbf{r}_t$, respectively. Note that, we use the multiplicative scale to better show the differences between relation patterns. As all the translations have values far less than one, the differences between the distribution of relation patterns will be clearer after multiplying by a larger number.

Table 5

Parameter efficiency on FB15k-237. The number of parameters of the first two rows is generated by the optimal configuration that reported in [12].

Model	Para. count	Emb. size	MRR	H@10	H@3	H@1
TransE [9]	14.78M	1000	.332	.531	.372	.233
RotatE [12]	29.32M	1000	.338	.533	.375	.241
TimE(**ours**)	15.25M	1000	.346	.537	.382	.250
TimE(**ours**	7.63M	500	.340	.533	.377	.246

From Fig.6(a)-(k) we can see that:

1) For symmetry pattern, cross-operation requires $\mathbf{r}_h = \mathbf{r}_t$, and $\mathbf{r} = \mathbf{0}$. Fig.6(a) - (c) illustrate the embedding of a symmetric relation *_similar_to*, which extract from WN18RR. We can find that the embedding of translation and the diagonal projection matrices satisfy $\mathbf{r}$=0, but $\mathbf{r}_h \neq \mathbf{0}$ and $\mathbf{r}_h = \mathbf{r}_t$. It shows that TimE can work well on symmetric relations. However, Fig.6(d) shows that the embedding of an anti-symmetric relation *region*, which extract from FB15k-237, does not has such property.
2) For inversion pattern, cross-operation requires $\mathbf{r} + \mathbf{r}^{-1} = \mathbf{0}$, $\mathbf{r}_h = \mathbf{r}_t^{-1}$ and $\mathbf{r}_t = \mathbf{r}_h^{-1}$. As there are no inverse relations in FB15k-237 and WN18RR, we investigate the embedding of inverse relations on WN18. Fig.6(e) illustrates the element-wise addition values of the relation r_1 = *_hypernym* and its inverse relation r_2 = *_hyponym* are $\mathbf{0}$, which presents $\mathbf{r}_1 = -\mathbf{r}_2$. From Fig.6(f)-(g) we can find that the embedding of the diagonal projection matrices satisfy $\mathbf{r}_h = \mathbf{r}_t^{-1}$ and $\mathbf{r}_t = \mathbf{r}_h^{-1}$. It indicates that TimE can also work well on inverse relations.
3) For composition pattern, TimE only requires the embedding of translations of the composed relation to be the addition of the other two relations. Fig.6(h)-(k) illustrate composition relation *winner*$_1$ is the addition of the other two relaions *award* and *winner*$_2$ which extract from FB15k-237.

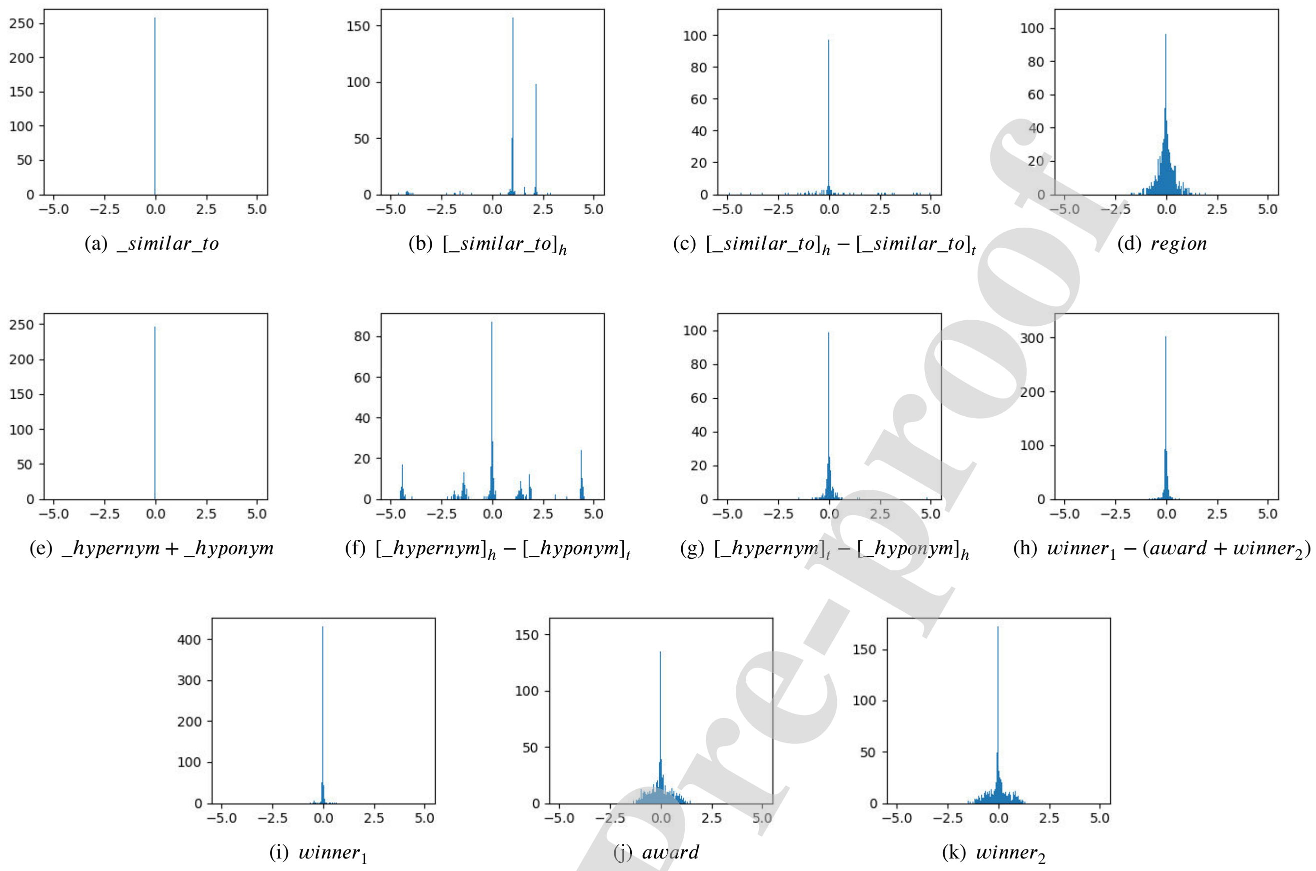
Parameter efficiency of TimE We investigate parameter efficiency of TimE. For a fair comparison, we consider Rotate and TransE with self-adversarial sampling and negative sampling loss. From table 5 we can see that the number of parameters of TimE with the best performance on FB15k-237 is about half of RotatE, just slightly more than TransE. From table 5 we can also see that TimE with 7.63M parameters can still achieve state-of-the-art results, and performs better than the previous best model RotatE with 29.32M parameters on all metrics, except that H@10 is equal.

4.5. Ablation studies

Table 6 and table 7 show the results from our ablation studies of cross-operation and self-interaction on four benchmarks. We can see that the cross-operation can significantly improve the performance of TimE on almost all datasets, especially those containing inverse relations. Specifically, the cross-operation can increase the H@1 and H@3 scores on FB15k by 5% and 2.3%, respectively, which shows the effectiveness of cross-operation. From table 6 and table 7, we can also see that the self-interaction also can improve the performance of TimE on all datasets. However, self-interaction is more effective for datasets that do not contain inverse relations, such as FB15k-237, WN18RR.

5. Conclusion

In this paper, we introduced TimE — a novel embedding model for link prediction, which presents each entry of the head (or tail) entity embeddings as a point in time domain space, and the corresponding entry of tail (or head) entity embeddings as a point in frequency domain space, and each relation is defined as a composition relation to capture diversity distribution of entity embeddings. In addition, we propose a cross-operation to model inverse and symmetric relations. Experimental results show that TimE outperforms several existing state-of-the-art models on five large-scale benchmarks for the link prediction task. Moreover, in our analysis, we show that TimE can better capture the differences between entities, and also can model all the relation patterns (including symmetry/anti-symmetry, inversion, and composition). In future work, we will focus on improving



(a) *_similar_to* (b) $[\_similar\_to]_h$ (c) $[\_similar\_to]_h - [\_similar\_to]_t$ (d) *region*

(e) $\_hypernym + \_hyponym$ (f) $[\_hypernym]_h - [\_hyponym]_t$ (g) $[\_hypernym]_t - [\_hyponym]_h$ (h) $winner_1 - (award + winner_2)$

(i) $winner_1$ (j) *award* (k) $winner_2$

Fig. 6: Histogram of relation embeddings, where $[x]_h$ and $[x]_t$ represent the head and tail diagonal projection matrices embeddings of relation x, respectively, - and + represent element-wise addition and element-wise subtraction, respectively. Subgraphs (a)-(c), represent the histogram of symmetric relations, where the relation *_similar_to* is extracted from WN18RR. Subgraph (d) represents the histogram of antisymmetric relations, where the relation *region* is extracted from FB15k-237. Subgraphs (e)-(g) represent the higtogram of inverse relations, where the relations *_hypernym* and *_hyponym* are extracted from WN18. Subgraphs (h)-(k) represent the histogram of composition relaitons, where the relations $winner_1$, *award* and $winner_2$ are extracted from FB15k-237. For simplicity, we use $winner_1$ to represent relation /award/award_winner/awards_won/award/award_honor/award_winner, *award* to represent relation /award/award_nominee/award_nominations./award/award_nomination/award, $winner_2$ to represent relation /award/award_category/winners./award/award _honor/award_winner and *region* to represent relation /organization/organization/headquarters./location/mailing_address/state_province_region.

Table 6
Ablation results on FB15k and WN18.

	FB15k					WN18				
	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10
TimE(non-cross,non-self)	.761	50.4	.700	.801	.865	.946	**214**	.939	.950	.958
TimE(non-self)	**.796**	46.8	**.750**	.824	.877	**.949**	246	**.944**	.953	.960
TimE	.795	**45.9**	.747	**.825**	**.879**	**.949**	259	.943	**.954**	**.961**

Table 7
Ablation results on FB15k-237 and WN18RR.

	FB15k-237					WN18RR				
	MRR	MR	H@1	H@3	H@10	MRR	MR	H@1	H@3	H@10
TimE(non-cross,non-self)	.345	192	**.253**	.381	.528	.462	3435	.418	.473	.557
TimE(non-self)	.340	**170**	.243	.378	.535	.470	2919	.420	.486	.571
TimE	**.346**	171	.250	**.382**	**.537**	**.477**	**2858**	**.428**	**.493**	**.577**

the reliability of TimE prediction results, as we believe that the reliability of the embedding approach is as important as high-precision prediction.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial or not-for-profit sectors.

References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, in: Proc. ACM SIGMOD 14th Int. Conf. Manage. Data, 2008, pp. 1247–1250.

[2] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: A Core of Semantic Knowledge, in: Proc. 16th Int. Conf. World Wide Web, 2007, pp. 697–706.

[3] G. A. Miller, WordNet: A Lexical Database for English, COMMUNICATIONS OF THE ACM. 38 (1995) 39–41.

[4] D. Guo, D. Tang, N. Duan, M. Zhou, J. Yin, Dialog-to-Action: Conversational Question Answering Over a Large-Scale Knowledge Base, in: Proc. Adv. Neural Inf. Process. Syst, 2018, pp. 2942–2951.

[5] B. Y. Lin, X. Chen, J. Chen, X. Ren, KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning, in: Proc. EMNLP-IJCNLP, 2019, pp. 2829–2839.

[6] H. Wang, F. Zhang, X. Xie, M. Guo, DKN: Deep Knowledge-Aware Network for News Recommendation, in: Proc. 27th Int. Conf. World Wide Web, 2018, pp. 1835–1844.

[7] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, KGAT: Knowledge Graph Attention Network for Recommendation, in: Proc. ACM SIGKDD 25th Int. Conf. Knowl. Discovery Data Mining, 2019, pp. 950–958.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, in: Proc. Adv. Neural Inf. Process. Syst, 2013, pp. 3111–3119.

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: Proc. Adv. Neural Inf. Process. Syst, 2013, pp. 2787–2795.

[10] Y. Lin, Z. Liu, M. Sun, Modeling Relation Paths for Representation Learning of Knowledge Bases, in: Proc. Conf. Empirical Methods Natural Language Process, 2015, pp. 705–714.

[11] T. Ebisu, R. Ichise, TorusE: Knowledge Graph Embedding on a Lie Group, in: Proc. 32th AAAI Conf. Artif. Intell, 2018, pp. 1819–1826.

[12] Z. Sun, Z. Deng, J. Nie, J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, in: Proc. 7th ICLR, 2019.

[13] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, X. Zhu, Knowledge Graph Embedding by Flexible Translation, in: Proc. 15th Int. Conf. Principles Knowl. Represent. Reasoning, 2016, pp. 557–560.

[14] H. Xiao, M. Huang, Y. Hao, X. Zhu, From One Point to A Manifold: Orbit Models for Knowledge Graph Embedding, in: Proc. IJCAI, 2016, pp. 1315–1321.

[15] W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec, Embedding Logical Queries on Knowledge Graphs, in: Proc. Adv. Neural Inf. Process. Syst, 2018, pp. 2026–2037.

[16] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, H. Chen, Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks, in: Proc. NAACL, 2019, pp. 3016–3025.

[17] M. Grbovic, H. Cheng, Real-time Personalization using Embeddings for Search Ranking at Airbnb, in: Proc. ACM SIGKDD 24th Int. Conf. Knowl. Discovery Data Mining, 2018, pp. 311–320.

[18] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, IEEE Transactions on Knowledge and Data Engineering. 29 (2017) 2724–2743.

[19] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proc. 29th AAAI Conf. Artif. Intell, 2015, pp. 2181–2187.

[20] L. Cai, B. Yan, G. Mai, K. Janowicz, R. Zhu, TransGCN: Coupling Transformation Assumptions with Graph Convolutional Networks for Link Prediction, in: Proc. 10th International Conference on Knowledge Capture, 2019, pp. 131–138.

[21] M. Nickel, V. Tresp, A Three-Way Model for Collective Learning on Multi-Relational Data, in: Proc. 28th Int. Conf. Mach. Learn, 2011, pp. 809–816.

[22] M. Nickel, L. Rosasco, T. Poggio, Holographic Embeddings of Knowledge Graphs, in: Proc. 30th AAAI Conf. Artif. Intell, 2016, pp. 1955–1961.

[23] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proc. 33rd Int. Conf. Mach. Learn, 2016, pp. 2071–2080.

[24] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: Proc. 32th AAAI Conf. Artif. Intell, 2018, pp. 1811–1818.

[25] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, D. Phung, A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technol, 2018, pp. 327–333.

[26] Z. Wang, Z. Ren, C. He, P. Zhang, Y. Hu, Robust Embedding with Multi-Level Structures for Link Prediction, in: Proc. IJCAI, 2019, pp. 5240–5246.

[27] Y. Jia, Y. Wang, X. Jin, X. Cheng, Path-specific knowledge graph embedding, Knowledge-Based Systems. 151 (2018) 37–44.

[28] K. Toutanova, D. Chen, Observed Versus Latent Features for Knowledge Base and Text Inference, in: Proc. 3rd Int. Workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 57–66.

[29] F. Mahdisoltani, J. Biega, F. Suchanek, YAGO3: A Knowledge Base from Multilingual Wikipedias, in: CIDR, 2013.

[30] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proc. 28th AAAI Conf. Artif. Intell, 2014, pp. 1112–1119.

[31] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proc. Int. Conf. Learn. Representations, 2015.

[32] N. Guan, D. Song, L. Liao, Knowledge graph embedding with concepts, Knowledge-Based Systems 164 (2019) 38–44.

[33] H. Liu, Y. Wu, Y. Yang, Analogical inference for multi-relational embeddings, in: Proc. 34th Int. Conf. Mach. Learn, 2017, pp. 2168–2178.

[34] R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with Neural Tensor Networks for Knowledge Base Completion, in: Proc. Adv. Neural Inf. Process. Syst, 2013, pp. 926–934.

[35] Y. Dai, S. Wang, X. Chen, C. Xu, W. Guo, Generative adversarial networks based on Wasserstein distance for knowledge graph embeddings, Knowledge-Based Systems 190 (2020) 105–165.

[36] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge Graph Embedding via Dynamic Mapping Matrix, in: Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Process., 2015, pp. 687–696.

[37] K. Diederik, B. Jimmy, Adam: A Method for Stochastic Optimization, in: Proc. 3th ICLR, 2015.

[38] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: Proc. European Semantic Web Conference, 2018, pp. 593–607.

[39] M. K. Seyed, P. David, SimplE Embedding for Link Prediction in Knowledge Graphs, in: Proc. NIPS, 2018, pp. 4284–4295.

[40] Z. Zhang, J. Cai, Y. Zhang, J. Wang, Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction, in: Proc. 34th AAAI Conf. Artif. Intell, 2020, pp. 3065–3072.

- Propose a novel knowledge graph embedding model called TimE, which defines each relation as a composition relation to capture diversity distribution of entity embeddings.
- Propose a cross-opteration to model inverse and symmetric relations.
- TimE only requires triples in the knowledge graphs, and no other data or processes.

Qianjin Zhang: Conceptualization, Methodology, Software, Writing - Original Draft.

Ronggui Wang: Validation, Formal analysis.

Juan Yang: Writing - Reviewing & Editing.

Lixia Xue: Reviewing and Editing.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: